Ensemble learning methods on the space of covariance matrices : application to remote sensing scene and multivariate time series classification

Sara Akodad

université de BORDEAUX

THÈSE

présentée pour obtenir le grade de

DOCTEUR

DE L'UNIVERSITÉ DE BORDEAUX

École doctorale Sciences Physique et de l'Ingénieur (SPI)

SPÉCIALITÉ

AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE, INGÉNIERIE COGNITIQUE

Par

Sara Akodad

Ensemble learning methods on the space of covariance matrices: application to remote sensing scene and multivariate time series classification

Méthodes d'ensemble sur l'espace des matrices de covariance : application à la classification de scènes de télédétection et de séries temporelles multivariées.

Sous la direction de : Christian Germain et Lionel Bombrun

Soutenue le 8 décembre 2021

Membres du jury :

M. Abdourrahmane ATTO	Maître de conférences, LISTIC Polytech Annecy-Chambéry	Rapporteur
M. Mohammed EL HASSOUNI	Professeur, Université Mohammed V de Rabat	Rapporteur
Mme. Florence TUPIN	Professeur, Telecom Paris - Institut Polytechnique de Paris	Examinatrice
M. Jean-Baptiste FÉRET	Chargé de recherche, TETIS - INRAE Montpellier	Examinateur
M. Christian GERMAIN	Professeur, Bordeaux Sciences Agro	Directeur de thèse
M. Lionel BOMBRUN	Maître de conférences, Bordeaux Sciences Agro	Co-Directeur
M. Minh-Tan PHAM	Maître de conférences, Université Bretagne Sud - IRISA	Invité
M. Thierry BELOUARD	Ingénieur, BIOGECO - INRAE Cestas	Invité

Contents

Remerciements

Ce travail a été réalisé au Laboratoire d'Intégration du Matériau au Système (IMS) à Bordeaux.

La thèse a été co-encadrée par Christian GERMAIN et Lionel BOMBRUN. Je les remercie chaleureusement pour ce qu'ils m'ont respectivement apporté au cours de ces années d'activités communes, humainement et professionnellement, et pour mavoir épaulé dans la bonne humeur.

Je tiens également à exprimer mes remerciements à Yannick BERTHOUMIEU, chef de l'équipe Signal et Image, pour l'accueil qu'il ma réservé et pour l'intérêt qu'il a bien voulu porter à l'égard de mon travail.

Le projet TEMPOSS a été mené en collaboration avec Biogeco de l'INRAE. J'adresse mes remerciements à Thierry BELOUARD, référent national "Données, télédétection et épidémiologie" du département santé des forêts (DSF). Nos échanges se sont toujours avérés très enrichissants et les formations qu'il m'a proposé de suivre ont été très constructives et m'ont permis de comprendre les exigences liées à la santé des forêts.

Je remercie vivement les stagiaires qui ont participé à ce travail, Solène VILFROY, Quentin TALLON, Kayhan ALVANPOUR, Maria SAPANTAN et Maria-Camelia PUSCASU. Merci pour leur motivation, leur sympathie et surtout les résultats qu'ils ont apportés à ce travail.

Madame Florence TUPIN et Messieurs Abdourrahmane ATTO, Mohammed EL HASSOUNI, Jean-Baptiste FÉRET, Thierry BELOUARD et Minh-Tan PHAM mont fait lhonneur de juger cette thèse. Je leur adresse mes sincères remerciements.

Je remercie chaleureusement tous les autres collaborateurs avec qui j'ai eu l'occasion de travailler, notamment Junshi XIA, chercheur au centre de recherche avancé (AIP) de RIKEN (Japan), pour ses compétences techniques et scientifiques, son investissement et sa sympathie.

L'IMS a été un lieu où il a fait bon travailler et où a régné une ambiance chaleureuse tout au long de mon séjour. Jadresse mes remerciements à lensemble des permanents, aux doctorants, et aux stagiaires pour la bonne atmosphère qu'ils ont su créer.

De façon plus personnelle, je termine ces remerciements en adressant ma profonde gratitude envers ma famille : mes parents, Fatima et Mustapha, mes frères, Jaouad et Reda, et ma petite soeur de coeur, Myriam. Vos encouragements et votre soutien ont été décisifs pour cette thèse. Ce titre de docteur vous appartient aussi ! Je n'oublie pas mes amis, qui ont toujours cru en moi, mes sincères remerciements en particulier à Jihane, Laura, Yousra, Soumia et Youssef. Enfin, je remercie du fond du coeur Amine qui ma encouragé et supporté sans relâche tout le long de cette aventure.

List of Acronyms

AI	Affine-Invariant
AID	Aerial Image Dataset
BCE	Binary Cross Entropy
BI	Brightness Index
BiMap	Bilinear Mapping
BoRW	Bag Of Riemannian Words
BoW	Bag Of Words
CO	Correspondents-Observers
COTE	Collective Of Transformation-based Ensembles
CNN	Convolutional Neural Network
CNPF	Centre National de la Propriété Forestière (National forest ownership center)
CR-SWIR	Continuum Removal-Short Waved Infra-Red
DDTW	Derivative Dynamic Time Warping
DL	Deep Learning
DoG	Difference of Gaussian
DSF	Département Santé des Forêts (Department of forest health)
DTW	Dynamic Time Warping
DT-MRI	Diffusion Tensor Magnetic Resonnance Imaging
EEG	Electroencephalogram
ELCP	Ensemble Learning Covariance Pooling
EL-SCP	Ensemble Learning Covariance Pooling guided by Saliency
EM	Expectation Maximization Algorithm
ESA	European Space Agency
FIM	Fisher Information Matrix
Flat-COVE	Flat Collective of Transformation-based Ensembles
FP	Fixed Point Estimator
FV	Fisher Vectors
GAK	Global Alignment Kernel
GAN	Generative Adversarial Network
GIS	Geographical Information System
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Units

GSoP	Global Second-order Pooling
HH	Horizontal Transmitting, Horizontal Receiving Polarization Image
HIVE-COTE	Hierarchical Vote Collective Of Transformation-based Ensembles
HV	Horizontal Transmitting, Vertical Receiving Polarization Image
KDA	Kernel Discriminant Analysis
k-NN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
LIBRAS	Lingua BRAsileira de Sinais
LE	Log-Euclidean
LE-BoRW	Log-Euclidean Bag of Riemannian Words
LE-FV	Log-Euclidean Fisher Vectors
LE-VLAD	Log-Euclidean Vectors of Locally Aggregated Descriptors
LSTM	Long-Short Term Memory
MAP-EM	Maximum A Posteriori Expectation-Maximization
MCDCNN	Multi-Channel Deep Convolutional Neural Network
MDE	Most Diverse Ensembles
MIR	Middle Infra-Red
MLE	Maximum Likelihood Estimation
MSCP	Multilayer Stacked Covariance Pooling
MTS	Multivariate Time Series
MV	Majority Vote
NATOPS	The Naval Air Training and Operating Procedures Standardization
NBR	Noise Burn Ratio
NDII	Normalized Difference Infrared Index
NDVI	Normalized Difference Vegetation Index
NIR	Near-Infrared
NN	Neural Network
OA	Overall Accuracy
ONF	National Forest Office
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
ReEig	Eigenvalue Rectification
RF	Random Forest
RFV	Riemannian Fisher Vectors
RGD	Riemannian Gaussian Distribution

RNN	Recurrent Neural Network
RVI	Radar Vegetation Index
RVLAD	Riemannian Vectors of Locally Aggregated Descriptors
SAR	Synthetic Aperture Radar
SCM	Sample Covariance Matrix
SIFT	Scale-Invariant Feature Transform
SO-CNN	Second-Order Convolutional Neural Network
SO-TCK	Second-Order Time series Cluster Kernel
SPD	Symmetric Positive Definite
SPD-MTS	Multivariate Time Series of Symmetric Positive Definite matrix
SPDNet	Riemannian Symmetric Positive Definite Matrix Network
SRVF	Square Root Velocity Function
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCK	Time series Cluster Kernel
Time-CNN	Time Convolutional Neural Network
TISELAC	Time Series Land Cover Classification Challenge
TSC	Time Series Classification
t-SNE	t-Distributed Stochastic Neighbor Embedding
TSRVF	Transported Square Root Velocity Function
UTS	Univariate Time Series
VH	Vertical Transmitting, Horizontal Receiving Polarization Image
VLAD	Vectors of Locally Aggregated Descriptors
VV	Vertical Transmitting, Vertical Receiving Polarization Image
WDTW	Weighted Dynamic Time Warping

Résumé étendu en français

Introduction

La nouvelle génération de capteurs d'imagerie de télédétection permet d'obtenir des images à haute résolution spatiale, spectrale et temporelle avec des fréquences de revisite élevées. Ces capteurs permettent l'acquisition de séries temporelles multivariées telles que la réflectance spectrale de la surface dans plusieurs longueurs d'onde. La disponibilité de ces séries temporelles multivariées a suscité l'intérêt de la communauté de la télédétection pour développer de nouvelles stratégies d'apprentissage automatique pour la classification supervisée. Il existe de nombreux algorithmes dédiés à la classification de séries temporelles, où, étant donné un ensemble de séries temporelles avec des étiquettes de classe, un modèle est formé pour prédire avec précision la classe de nouvelles séries. Classiquement, les méthodes liées à la classification des séries temporelles reposent soit sur la mesure de la similarité entre les séries temporelles, soit sur l'extraction de caractéristiques statistiques sur chaque sous-séquence des séries temporelles [Atto *et al.* 2016, D'Urso & Maharaj 2012, Sakoe & Chiba 1978, Berndt & Clifford 1994], afin de former une nouvelle séquence comme données d'entrée ou même sur un ensemble de classifieurs basés sur l'apprentissage pour améliorer la robustesse de la classification face aux valeurs aberrantes.

La plupart de ces approches de pointe s'appuient sur des caractéristiques statistiques de premier ordre pour modéliser l'information contenue dans chaque série temporelle. Les caractéristiques générées à partir des statistiques de premier ordre fournissent des informations liées à chaque distribution de valeurs ponctuelles dans la séquence. Cependant, elles ne donnent aucune information sur les dépendances relatives des caractéristiques au sein d'un même point, comme les dépendances entre les attributs spectraux. Dans le but d'améliorer la précision et l'efficacité de la classification des séries temporelles, de récentes avancées dans les approches méthodologiques ont montré la pertinence de l'utilisation des descripteurs de second ordre pour la classification, l'indexation et la segmentation dans les applications de télédétection [Li *et al.* 2017, Akodad *et al.* 2018b], que ce soit pour des données 2-D, telles que des images, ou des signaux 1-D incluant les séries temporelles. Pour cela, la matrice de covariance de ces attributs spectraux/temporels est calculée et des outils de géométrie de l'information sont utilisés afin de manipuler ce type de données.

Dans ce contexte, l'objectif principal de cette thèse est de proposer de **nouvelles méthodes d'apprentissage d'ensemble sur l'espace des matrices de covariance.** En se basant sur la représentation log-Euclidienne des matrices de covariance calculées sur les sorties de couches convolutives d'un réseau de neurones ou encore sur des attributs multispectraux, nous visons à évaluer le potentiel de ces caractéristiques pour diverses applications dont la classification de scènes de télédétection et la classification de séries temporelles. Un intérêt particulier est consacré à l'évaluation du potentiel des images radar (Sentinel-1) et optiques (Sentinel-2) pour le problème forestier de la maladie de l'encre du châtaignier dans la forêt de Montmorency.

Géométrie Riemannienne et modélisation statistique dans l'espace des matrices symétriques définies positives (SPD)

Comme mentionné précédemment, nous nous intéressons à un problème de classification dans l'espace des matrices de covariance. Pour comparer deux matrices de covariance, l'approche la plus simple est d'utiliser la norme de Frobenius. Elle consiste à utiliser la norme euclidienne une fois que les matrices de covariance sont vectorisées. Tout en étant extrêmement simple, cette norme ne tient pas compte de la structure géométrique de ces matrices symétriques définies positives. En fait, les matrices de covariance *vivent* sur une variété Riemannienne. Dans le cas des matrices de covariance de taille 2×2, elles *vivent* dans un espace contraint de $\mathbb{R}^3$, représenté par un cône comme illustré sur la Figure 1.

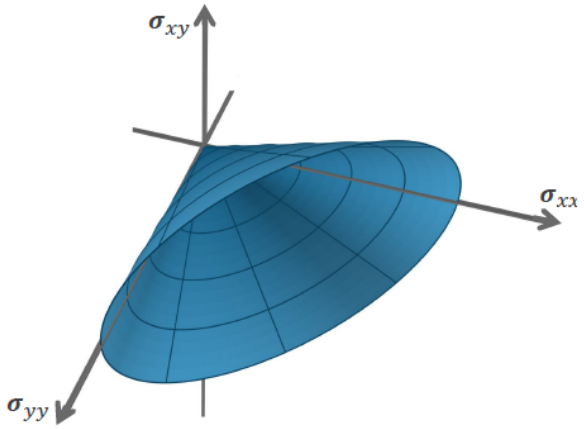


Figure 1: Espace des matrices de covariance de taille 2×2.

Pour traiter ce type de données, certains concepts de la géométrie différentielle sont nécessaires. Deux canevas statistiques Riemanniens peuvent être utilisés. Ils sont basés respectivement sur les métriques Riemannienne affine-invariante (AI) et log-euclidienne (LE). Pour la métrique AI, les calculs sont effectués directement sur la variété alors que pour la métrique LE, les calculs sont effectués sur un espace tangent. Pour cette dernière métrique (LE), les matrices de covariance sont projetées dans l'espace log-euclidien par l'opérateur du logarithme mapping à un point de référence, classiquement choisi égal à la matrice identité. Cette opération est suivie d'une étape de vectorisation pour obtenir la représentation log-Euclidienne. Pour une matrice de covariance $\mathbf{M}_n \in \mathcal{P}_d$ où $\mathcal{P}_d$ est l'espace des matrices symétriques définies positives de taille $d \times d$, le vecteur LE, noté $\mathbf{m}_n$, est obtenu tel que:

$$\mathbf{m}_n = \mathrm{Vec}\Big(\mathrm{Log}_{\mathbf{I}_d}(\mathbf{M}_n)\Big), \tag{1}$$

l'opérateur $\mathrm{Log}_{\mathbf{I}_d}$ représente le logarithme mapping qui permet de projeter la matrice de covariance $\mathbf{M}_n$ sur l'espace tangent à l'identité $\mathbf{I}_d$, suivi de l'opérateur de vectorisation Vec(). Et dans cet espace log-euclidien, les outils classiques de la géométrie Euclidienne peuvent être appliqués. D'un point de vue pratique, Arsigny *et al.* ont montré que les métriques affine-invariantes et log-Euclidiennes fonctionnent mieux que la norme de Frobenius pour l'interpolation et la régularisation destinée à l'imagerie synthétique et clinique de tenseur de diffusion 3D (DT-MRI) par résonance magnétique [Arsigny *et al.* 2006]. D'autre part, en comparant les métriques Riemanniennes log-Euclidiennes et affine-invariantes, on peut

observer qu'elles offrent plusieurs propriétés d'invariance (rotation, échelle, inversion). En outre, en pratique, elles permettent d'obtenir des résultats comparables pour une grande variété d'applications [Ilea *et al.* 2018b, Arsigny *et al.* 2006]. Au cours de cette thèse de doctorat, la métrique log-euclidienne est prioritairement retenue pour sa facilité d'utilisation et sa faible complexité calculatoire.

Dans le premier chapitre, nous introduisons quelques notions fondamentales de la géométrie de l'information, utiles pour traiter ce type de données. Deux cadres statistiques Riemanniens complets, basés sur les métriques log-euclidienne et affine-invariante, sont présentés. Les modèles gaussiens sont considérés sur ces deux espaces métriques, ainsi que leurs extensions aux modèles de mélange de gaussiennes (GMM). Pour cela, un modèle gaussien sur une variété Riemannienne est présenté, ainsi qu'une distribution gaussienne multivariée sur l'espace log-Euclidien. Dans le cas de la métrique log-Euclidienne, les matrices de covariance sont projetées sur l'espace tangent à un point de référence donné, classiquement fixé égal à la matrice identité. Cela peut entraîner une certaine distorsion lors de la projection si les matrices de covariance sont situées loin de cette référence. Pour éviter ce problème en limitant la distorsion pendant la projection, nous proposons un modèle de mélange de gaussiennes avec plusieurs points de référence, un pour chaque composant du modèle. Nous proposons également un algorithme d'expectation maximisation (EM) pour estimer les paramètres de ce GMM.

Méthodes d'ensemble basées sur le calcul de matrices de covariance en sortie de couches convolutives d'un CNN

Dès lors que des descripteurs discriminants ont été extraits. Les méthodes de classification appartenant aux familles d'apprentissages automatiques peuvent être employées. Les approches traditionnelles qui ont été très populaires au début des années 2000 sont basées sur l'encodage de descripteurs, tel que les vecteurs de Fisher [Perronnin & Dance 2007], les vecteurs VLAD [Jégou *et al.* 2010] ou encore les sacs de mots (BoW) [Csurka *et al.* 2004].

Récemment, au vu de la réussite remportée par les réseaux de neurones, en particulier les réseaux de neurones convolutifs (CNN) qui permettent l'extraction et l'apprentissage automatique des descripteurs sur les images, plusieurs auteurs ont dédié leurs travaux à proposition de méthodes hybrides, qui combinent à la fois des méthodes d'encodage avec les réseaux de neurones convolutifs. A titre d'exemple, le réseau de Fisher [Perronnin & Larlus 2015], le NetVLAD [Arandjelovic *et al.* 2015] ou encore l'encodage par vecteur de Fisher des sorties de couches convolutives d'un CNN, nommé "Hybrid FV" [Li *et al.* 2017]. D'autre part, considérant l'attention que les statistiques du second ordre ont pu attirer ces dernières années, l'information modélisée par les matrices de covariance a fait l'objet de travaux de recherche où ces objets ont été intégrés dans des architectures de classification. Dans ce contexte, plusieurs approches ont été proposées, tel que les sacs de mots log-Euclidiens (LE BoW) [Faraki *et al.* 2015b], les sacs de mots Riemmanniens (BoRW) [Faraki *et al.* 2014], les vecteurs VLAD log-Euclidiens (LE VLAD) [Faraki *et al.* 2015a], les VLAD Riemanniens (RVLAD), etc. Enfin, afin de tirer profit de l'information modélisée par les statistiques du second ordre dans des approches d'apprentissage automatique par le biais des réseaux de neurones, plusieurs travaux se sont focalisés sur la proposition d'un large panel de réseaux de neurones qui intègrent des couches destinées au calcul de matrices de covariance, tel que le SPDNet [Huang & Gool 2017], le MPNCov [Li *et al.* 2018], ou encore le SO-CNN [Yu & Salzmann 2017].

Toutefois, toutes ces méthodes requièrent l'apprentissage du réseau sur un large ensemble de données, tel que la base de données ImageNet qui est constituée de plus d'un million d'images [Russakovsky *et al.* 2014]. Dans le contexte de cette thèse, les bases de données utilisées, de taille plus modeste, ne sont pas adaptées au cas d'un apprentissage de bout en bout d'un réseau de neurones. Pour pallier à cela, l'apprentissage par transfert est exploité. Il s'agit d'une approche permettant le transfert de connaissances d'un réseau pré-entraîné sur une base de données de départ, à une autre base de données d'intérêt.

Dans ce contexte, nous avons proposé deux approches hybrides d'apprentissage, combinant à la fois les méthodes d'encodage et les réseaux CNN par transfert, basées sur le calcul de matrices de covariance de façon locale et globale sur les sorties des couches convolutives d'un CNN. En effet, les CNN standards sont généralement composés par des couches de pooling, qui permettent de calculer la valeur moyenne (*average pooling*, en anglais) ou valeur maximale (*max pooling*, en anglais) pour chaque patch des cartes de caractéristiques. Pour améliorer la capacité de ces réseaux, la représentation de second ordre des cartes de caractéristiques de CNN a récemment montré son intérêt [Akodad *et al.* 2018b]. Cela consiste en une opération où la matrice de covariance des cartes de caractéristiques est extraite localement ou globalement (*covariance pooling*, en anglais). Au final, chaque image peut être représentée par un ensemble de matrices de covariance. D'une part, l'approche locale, "Hybrid LE FV" [Akodad *et al.* 2018b], inspirée initialement de l'approche "Hybrid FV" [Li *et al.* 2017], s'appuie sur les matrices de covariance extraites localement sur les premières couches d'un CNN, qui sont ensuite encodées par les vecteurs de Fisher calculés sur leur représentation log-Euclidienne.

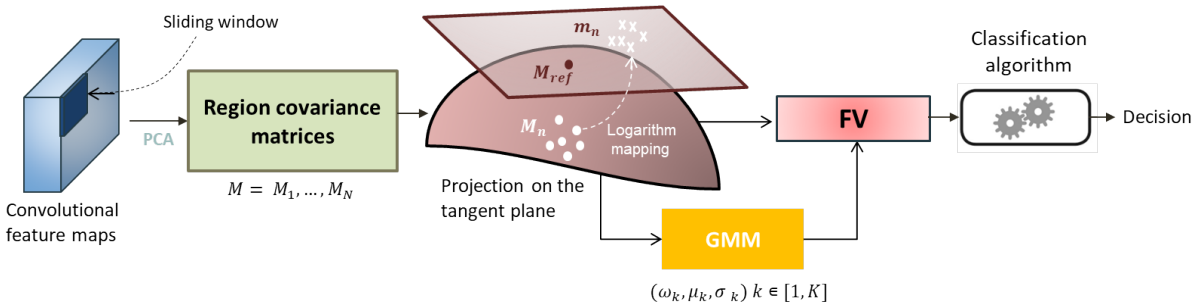


Figure 2: Principe de l'encodage des matrices de covariance par les vecteurs de Fisher log-Euclidiens (Hybrid LE FV).

D'autre part, afin de renforcer la robustesse de la classification, la méthode proposée est intégrée dans un algorithme d'apprentissage d'ensemble, pour donner lieu à l'approche "Ens. Hybrid LE FV". Afin d'illustrer le potentiel des statistiques du second ordre dans un problème de classification, le tableau 2 met en évidence les performances de classification sur la base UC Merced. Les deux premières couches convolutives (Conv1) et (Conv2) d'un réseau VGG-16 pré-entrainé sur la base ImagNet sont considérées et un pourcentage de 10% d'images est retenu pour la phase d'apprentissage. Ce paramétrage est identique pour l'ensemble des résultats ci-dessous. Comme observé, les résultats permettent de mettre en évidence, d'une part, le potentiel des statistiques du second ordre, par le biais de l'utilisation des matrices de covariance où un gain significatif d'environ 20% est accordé à la méthode "Hybrid LE FV" comparée à son homologue "Hybrid FV". D'autre part, l'utilisation d'un ensemble de classifieurs, dans une

Méthode	Conv 1	Conv 2
Hybrid FV [Li *et al.* 2017]	41.4 ± 0.2 %	43.7 ± 1.1 %
Hybrid LE FV [Akodad *et al.* 2018b]	61.2 ± 0.8 %	65.1 ± 1.6 %
Ens. Hybrid LE FV	**62.4 ± 0.9 %**	**68.1 ± 1.7 %**

Table 2: Résultats de classification sur la base UC Merced en considérant la première et la seconde couche convolutive du réseau VGG-16 ($p = 10\%$).

approche d'ensemble, permet d'accroître les résultats de classification.
Une seconde méthode de reconnaissance de scène est proposée. Il s'agit de l'approche globale ELCP. Elle s'inspire de la méthode MSCP de l'état de l'art proposée dans [He *et al.* 2018], où une seule matrice de covariance est calculée sur les cartes de caractéristiques des couches profondes d'un CNN comme illustré sur la Figure 3.

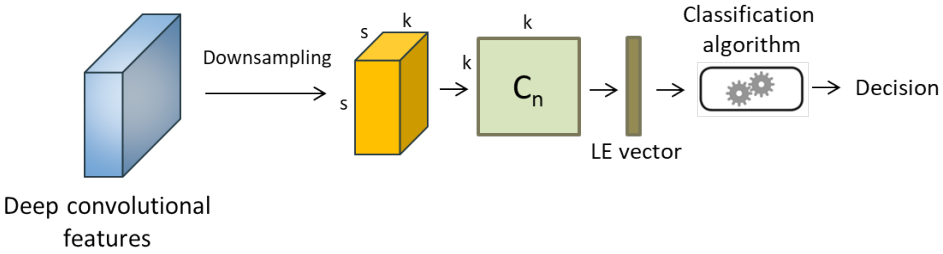


Figure 3: Principe de l'approche de covariance "pooling" globale.

Comparée aux méthodes exploitant les statistiques du second ordre, l'approche ELCP a démontré d'excellents résultats en termes de taux de bonne classification. A titre d'exemple, le tableau 3 illustre les performances de classification sur la base UC Merced. D'une part les méthodes exploitant les réseaux de neurones convolutifs (CNN) surpassent les méthodes traditionnelles, telles que celles basées sur l'encodage par vecteurs de Fisher de descripteurs SIFT. D'autre part l'utilisation de réseaux de neurones pré-entrainés est privilégiée du fait de la faible dimension des jeux de données en termes de nombre d'images d'apprentissage. Par ailleurs, les méthodes exploitant les statistiques du second ordre, à savoir MSCP et ELCP, donnent de meilleurs résultats. En plus, l'emploi d'une méthode d'ensemble résulte en un gain significatif, où la méthode ELCP a su surpasser son homologue de l'état de l'art (MSCP).

Méthode	OA (Mean ± sd)
FV (SIFT) [Perronnin *et al.* 2010b]	62.3 ± 1.1 %
CNN (VGG-16 fine-tuned)	62.7 ± 1.8 %
CNN (VGG-16 feat. extraction + SVM) [Chatfield *et al.* 2014]	82.7 ± 0.6 %
MSCP (VGG-16) [He *et al.* 2018]	86.3 ± 1.0 %
ELCP [Akodad *et al.* 2019c] (VGG-16)	**88.4 ± 1.4 %**

Table 3: Performances de classification de l'approche multi-couches proposée comparée aux méthodes de l'état de l'art sur la base UC Merced ($p = 10\%$).

De plus, afin de donner une plus grande importance aux objets d'intérêt présents dans les images, nous avons proposé d'utiliser une matrice de covariance pondérée par l'information de saillance comme illsutré sur la Figure 4.

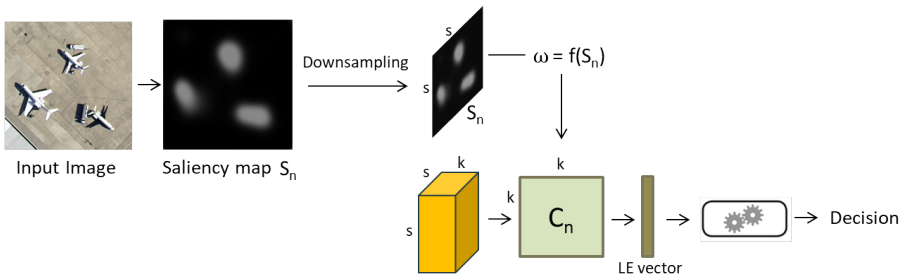


Figure 4: Covariance pooling des descripteurs CNN guidés par les cartes de saillance.

Ensuite, en collaboration avec le centre de recherche RIKEN au Japon, nous proposons d'unifier ces travaux en présentant une approche d'apprentissage par transfert qui bénéficie à la fois des aspects locaux et globaux. Cette approche d'apprentissage d'ensemble, basée sur les ensembles les plus diversifiés notée "Fusion Ens. Hybrid LE FV-ELCP" [Akodad *et al.* 2020c], combine efficacement les décisions fournies et permet d'améliorer la performance de classification. Pour valider les méthodes, nous considérons différents types d'ensembles de données de télédétection, y compris des images aériennes et satellites. Tels que les bases UC Merced Land Use Land Cover, AID et SIRI-WHU. Des expérimentations sont également effectuées sur des images texturées acquises par le satellite Pléiades pour la différenciation des classes d'âge des peuplements de pins maritimes et la classification de parcs ostréicoles dans le bassin d'Arcachon.

Pour illustrer quantitativement les résultats obtenus, le tableau 4 résume les performances de classification sur la base UC Merced en utilisant les différentes approches proposées. Pour

Base de données	**Méthode**	**OA (Mean ± sd)**
UC Merced $p = 10$ %	Ens. Hybrid LE FV (conv1)	62.4 ± 0.9 %
	Ens. Hybrid LE FV (conv2)	68.1 ± 1.7 %
	ELCP	**88.4 ± 1.4 %**
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**88.7 ± 1.1 %**

Table 4: Performances de classification sur la base UC Merced employant les méthodes Ens. Hybrid LE FV, ELCP ainsi que leur fusion Ens. LE FV - ELCP (p = 10 %).

conclure, la combinaison des deux approches, globale et locale, dans une unique architecture révèle un léger gain en termes de performances de classification.

Classification de séries temporelles multivariées

La disponibilité des séries temporelles multivariées a suscité l'intérêt de la communauté de la télédétection et plus généralement du *machine learning* pour l'élaboration de nouvelles stratégies d'apprentissage pour la classification supervisée, notamment les méthodes basées sur le calcul de distance point à point entre les séries. La façon la plus simple de comparer deux séquences de même longueur consiste à additionner la distance ordonnée point à point entre elles. Pour ce faire, la fonction de distance couramment utilisée est la distance euclidienne [Bagnall *et al.* 2016b], qui correspond à la norme $\mathcal{L}_2$. Cependant, la distance euclidienne et ses variantes présentent plusieurs inconvénients. D'une part, la distance euclidienne est sensible aux transformations comme le décalage temporel qui induit des résultats incorrects dans

certaines applications. À titre d'exemple, dans le domaine de la télédétection, en raison de la variabilité intrinsèque entre les champs, comme la température de l'air, le drainage du sol et d'autres caractéristiques environnementales, l'évolution temporelle d'une certaine culture dans deux champs différents peut avoir un comportement différent tout en appartenant à la même classe d'occupation de sol. D'autre part, la distance euclidienne souffre de l'invariance à la reparametrisation. Cela signifie que la distance entre deux séries $\mathbf{x}_1$ et $\mathbf{x}_2$, notée $\mathrm{d}(\mathbf{x}_1, \mathbf{x}_2)$ n'est pas conservée sous une quelconque transformation, telle qu'une composition par la fonction γ, à savoir:

$$d(\mathbf{x}_1, \mathbf{x}_2) \neq d(\mathbf{x}_1(\gamma), \mathbf{x}_2(\gamma)) \tag{2}$$

Compte tenu de ces deux précédentes limitations, le chapitre 3 s'intéresse dans un premier temps aux solutions suivantes. Tout d'abord, la distorsion dans l'axe du temps peut être traitée par la déformation temporelle dynamique (DTW) et l'invariance de reparametrisation est résolue par l'utilisation de la fonction (SRVF) [Srivastava *et al.* 2011], qui fournit une nouvelle représentation de la série temporelle considérée tout en assurant les propriétés d'invariance.

Par ailleurs, la plupart des approches destinées à la classification de séries temporelles reposent sur des statistiques de premier ordre pour modéliser l'information sous-jacente de chaque série temporelle. Les descripteurs générés à partir de statistiques de premier ordre fournissent des informations relatives à chaque distribution des points dans la séquence. Cependant, ils ne donnent aucune information sur les dépendances relatives des attributs d'un même point, telles que les dépendances entre les attributs spectraux. Afin d'améliorer l'exactitude et l'efficacité de la classification des séries temporelles, les progrès récents dans les approches méthodologiques ont montré la pertinence de l'utilisation des descripteurs du second ordre pour la classification, l'indexation et la segmentation pour des applications de télédétection [Pham *et al.* 2016, He *et al.* 2018, Akodad *et al.* 2019c, Akodad *et al.* 2020a], que ce soit pour les données 2D, telles que les images, ou les signaux 1-D, y compris les séries temporelles. Celles-ci comprennent des matrices de covariance symétriques définies positives qui décrivent les statistiques du second ordre de la série temporelle multivariée. Pour illustrer l'idée, la Figure 5 montre un exemple de deux séries temporelles pour deux applications différentes. Le premier représente une action de course où les capteurs enregistrent les coordonnées x, y et z des mouvements de la main et du genou pour une application de reconnaissance d'action. La seconde montre l'évolution temporelle de la réflectance spectrale et des indices de végétation (R, G, B, NDVI, etc.) pour reconnaître une plantation de riz pour une application en télédétection.

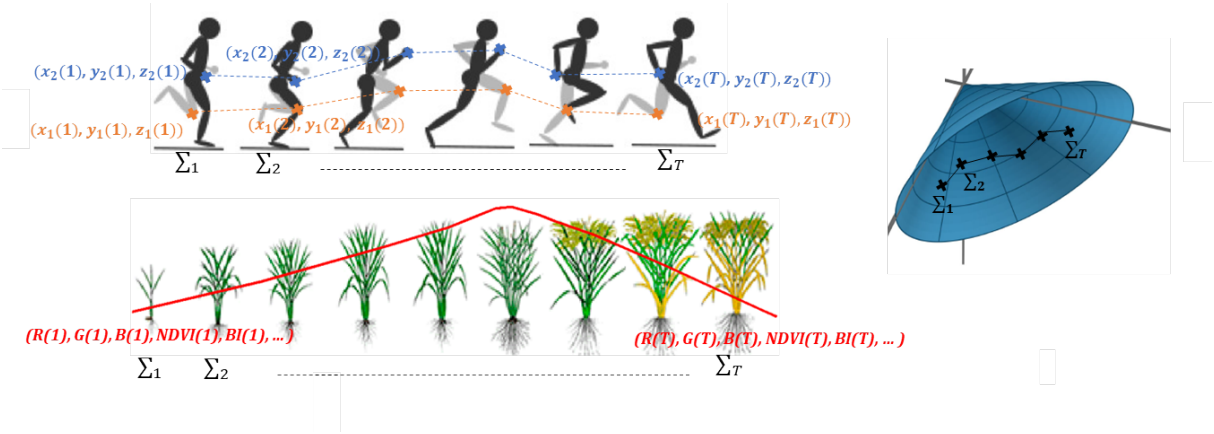


Figure 5: Exemples de séries temporelles multivariées et de trajectoires de matrices de covariance.

À mesure qu'une personne exécute une action, le bras et le genou peuvent se déplacer de façon corrélée au fil du temps. De même, dans la partie inférieure de la figure 5, la variation temporelle de la réflectance spectrale et des indices de végétation de la culture de riz peut avoir un comportement corrélé, et cette relation peut évoluer avec le temps. Pour capturer ces corrélations entre les attributs, une matrice de covariance Σ_t est calculée à chaque instant. À droite de la figure 5, les matrices de covariance calculées forment une trajectoire dans le cône bleu. Ainsi, le problème de classification de séries temporelles multivariées peut être reformulé en un problème de classification de trajectoires de matrices de covariance.

De plus, comme la principale contribution du chapitre est la proposition de modèles de classification adaptés à des trajectoires de matrice de covariance, le cadre SRVF est étendu à la représentation TSRVF (Transport Square-Root Velocity Function) [Su *et al.* 2014a]. Elle fournit un moyen de représenter les trajectoires sur une variété Riemannienne de sorte que l'invariance de re-paramétrage reste valide. Pour aller plus loin, afin de bénéficier des avantages des méthodes à noyau, des représentations basées sur l'apprentissage de dictionnaires et des stratégies d'apprentissage d'ensemble, Mikalsen *et al.* ont introduit la méthode TCK (Time series Cluster Kernel) dans [Mikalsen *et al.* 2018]. Cette dernière a montré des résultats compétitifs pour la classification des séries temporelles. Dans ce travail, nous proposons une extension de la méthode TCK (nommée SO-TCK) au cas de données qui *vivant* sur une variété. Toutes ces approches sont validées sur des bases de données de séries temporelles multivariées incluant des applications pour la reconnaissance d'actions et la classification de séries temporelles de télédétection pour la reconnaissance de cultures. Le tableau 5 illustre les résultats de classification obtenus sur différentes base de données de séries temporelles en reconnaissance de mouvement [Dua & Graff 2017] et télédétection (Tiselac) [Ienco 2017].

Méthode	Libras	Natops	Character trajectory	Racket sports	Tiselac[1]
Euclidean distance	79.2 ± 1.1	77.8 ± 3.4	**95.5 ± 0.4**	68.8 ± 2.6	60.4 ± 1.0
Warping + Euclidean distance	85.6 ± 3.8	71.7 ± 4.3	**95.6 ± 0.2**	81.3 ± 2.9	62.8 ± 2.9
SRVF + Euclidean distance	82.8 ± 2.3	**80.5 ± 2.4**	91.9 ± 2.9	63.7 ± 2.7	70.4 ± 1.3
SRVF + Warping + Euclidean distance	82.8 ± 2.3	**80.5 ± 2.4**	91.9 ± 0.2	63.8 ± 2.8	70.2 ± 1.2
TCK	72.6 ± 2.9	61.4 ± 3.5	91.7 ± 0.5	81.5 ± 3.2	63.7 ± 1.3
TSRVF + Euclidean distance	80.1 ± 3.0	76.2 ± 2.3	94.7 ± 0.4	91.1 ± 1.7	89.3 ± 0.5
TSRVF + Warping + Euclidean distance	**88.5 ± 2.7**	75.0 ± 2.7	93.2 ± 0.4	**94.7 ± 1.2**	**92.7 ± 0.4**
SO-TCK	87.1 ± 2.2	71.3 ± 3.7	93.9 ± 0.8	87.8 ± 2.8	74.5 ± 0.9

Table 5: Performances de classification sur différentes bases de données comparant les différentes stratégies, incluant la distance Euclidienne, l'alignement par DTW et les représentations du premier et second ordre par SRVF et TSRVF respectivement.

Comme observé, les résultats de classification démontrent un intérêt à exploiter les dépendances entre attributs par le biais des statistiques du second ordre. D'une part, la méthode proposée SO-TCK permet d'obtenir un gain significatif comparé à la version originale TCK. D'autre part, l'alignement par DTW et les représentations TSRVF améliorent les résultats de classification dans la majorité des cas.

Suivi de la santé des forêts par le biais des séries temporelles Sentinel-1 et Sentinel-2

En tant que ressources naturelles vitales, les forêts fournissent divers services écosystémiques, où elles constituent une source clé de nourriture et de fibres pour les humains. Aujourd'hui, de nombreux pays ont conclu des accords internationaux ou régionaux pour protéger leurs ressources forestières. Le dernier axe de cette thèse concerne la modélisation de signaux temporels mesurés par les capteurs radar (Sentinel-1) et optique (Sentinel-2). En particulier, pour le suivi du problème sylvo-sanitaire de la maladie de l'encre du châtaignier de la forêt de Montmorency. Pour cela, nous cherchons à évaluer le potentiel des images radars et optiques acquises respectivement par les capteurs Sentinel-1 et Sentinel-2. Nous étudions également l'intérêt d'une approche multimodale en combinant ces deux types de données.

Aujourd'hui, les méthodes de télédétection par satellite permettent de détecter relativement bien les coupes rases. Cependant, le cycle foliaire saisonnier n'étant pas forcément bien saisi, il est généralement difficile de distinguer les différents niveaux de maladie. Pour illustrer cela, la Figure 6 représente l'évolution temporelle de l'indice de végétation NDVI pour chaque classe d'intérêt, à savoir : (1) Peuplement sain ou peu atteint, (2) en déclin sévère, (3) ruiné et (4) coupes rases.

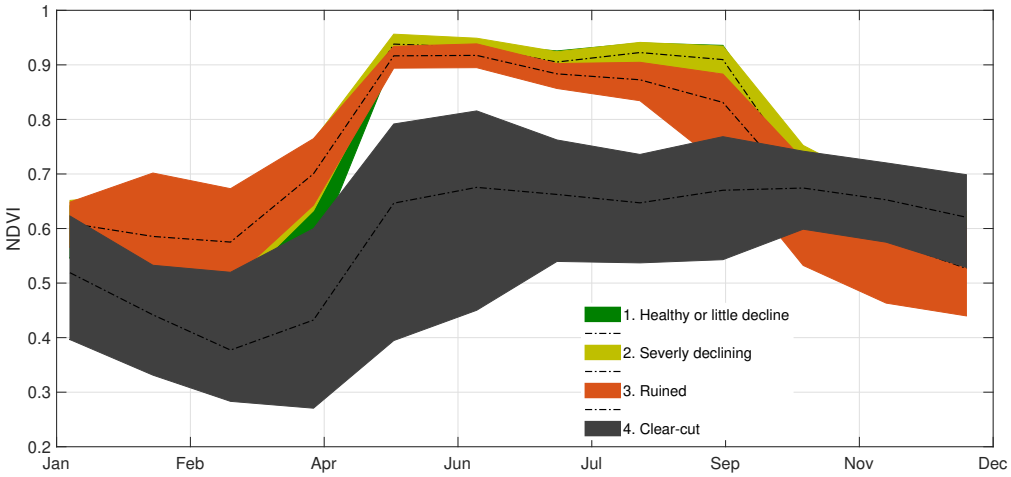


Figure 6: Évolution temporelle sur l'année 2020 de l'indice de végétation NDVI pour chaque classe d'intérêt.

D'un point de vue pratique, il existe diverses méthodes permettant d'étudier les changements saisonniers de végétation, en particulier le suivi des couverts forestiers, à travers des images satellite, l'une d'entre elles consiste à extraire des indices de végétation. Plusieurs indices de végétation liés à la dégradation forestière peuvent être calculés à partir des images optiques ou radars afin de surveiller la santé des forêts. A cet effet, les indices utilisés impliquent les bandes spectrales du visible, du proche infrarouge et du moyen infrarouge pour les données optiques, et des ratios de coefficients de rétrodiffusion polarimétriques pour les données radar. De plus, en considérant la dépendance entre ces indices par le biais des statistiques du second ordre, nous introduisons une approche d'apprentissage d'ensemble pour la classification de l'état de santé de la forêt (saine ou peu atteinte, en déclin sévère, ruinée et coupe rase). Ensuite, étant donné que la maladie évolue progressivement, d'un peuplement sain à un peuplement complètement détruit, nous proposons de reformuler le problème en prédisant une variable quantitative qui correspond à un indice de dégradation de la forêt (ou de l'état de santé). Sur cette base, le potentiel des données Sentinel-1 et Sentinel-2 est évalué pour cette application.

En termes de taux de bonne classification, le tableau 6 illustre un léger gain de l'approche d'ensemble proposée basée sur la covariance globale. Les résultats relatifs aux données optiques et radars démontrent une bonne détection de la classe coupes rases (Classe 4), tandis que les autres classes ne sont pas suffisamment bien classifiées.

Comme la maladie évolue continuellement de peuplements sains à des arbres complètement détruits, nous proposons de reformuler le problème comme un problème de prédiction d'une variable quantitative correspondant à un indice de dégradation des forêts. Pour cela, un modèle de régression est proposé. Comme les coupes rases sont bien détectées, elle ne seront pas prises en compte dans ce qui suit. Afin de suivre la maladie de la forêt, un indicateur de la santé des forêts est calculé sur la base du protocole d'observation DEPERIS. Les arbres de type A sont les plus sains et les arbres de type F sont les plus touchés par la maladie de l'encre. Selon ce dernier score, nous avons dérivé l'indicateur I pour estimer l'état sanitaire de la forêt en fonction du

		Classe 1	Classe 2	Classe 3	Classe 4	**OA** (%)	**Kappa** (%)
Optique	**1st-order**	48.5 ± 12.1	**42.1 ± 11.3**	**61.6 ± 7.6**	76.2 ± 7.6	58.8 ± 4.7	44.6 ± 6.0
	2nd-order	**73.8 ± 10.2**	27.3 ± 10.1	**68.8 ± 5.9**	**89.5 ± 4.6**	**70.9 ± 2.8**	**59.6 ± 3.5**
Radar	**1st-order**	44.5 ± 12.9	**46.3 ± 9.2**	**65.0 ± 4.5**	69.2 ± 7.8	56.2 ± 6.2	41.9 ± 7.6
	2nd-order	**74.2 ± 10.9**	11.5 ± 10.4	**65.5 ± 7.3**	**83.7 ± 7.5**	**66.1 ± 3.1**	**52.5 ± 3.6**

Table 6: Comparaison des performances de classification entre les modèles du premier et second ordre: Utilisation de la moyenne des observations pour le modèle du premier ordre et de la covariance augmentée globale pour le second ordre. Résultats obtenus pour l'imagerie optique et radar.

pourcentage d'arbres sains/atteint. Il est donné par :

$$I = \frac{5 \times (\%A) + 4 \times (\%B) + 3 \times (\%C) + 2 \times (\%D) + (\%E)}{5}. \tag{3}$$

L'indicateur de la santé des forêts représente une moyenne pondérée des scores DEPERIS. Il varie entre 0 et 1. Le score le plus bas correspond à un peuplement composé de 100% d'arbres de type F et la valeur maximale de 1 est atteinte pour les arbres sains (100% de type A). Comme cet indicateur est une variable continue, un modèle de régression est utilisé pour le prédire à partir des observations Sentinel-1 et Sentinel-2. Pour ce faire, le formalisme de l'approche de covariance globale proposé est réadapté au problème de régression dans un modèle d'ensemble. Les résultats obtenus sont reportés sur le tableau 7.

		Optique	**Radar**	**Fusion**
MSE	Moyenne	2.87×10^{-2}	2.96×10^{-2}	2.78×10^{-2}
	Covariance	3.17×10^{-2}	3.20×10^{-2}	2.83×10^{-2}
	Covariance + moyenne	2.67×10^{-2}	2.79×10^{-2}	$\mathbf{2.59\times10^{-2}}$

Table 7: Méthode d'ensemble pour la covariance globale dans un modèle de régression. Comparaison entre l'utilisation des données optiques, radar et la fusion des deux.

Quantitativement, l'erreur quadratique moyenne mesurée (MSE) montre une amélioration mineure lors de la combinaison des caractéristiques statistiques du premier et du deuxième ordre, par exemple la combinaison de la matrice de covariance et le vecteur de moyenne (Covariance + mean). En outre, l'utilisation de données optiques et radars dans un schéma de fusion apporte une petite amélioration par rapport à leur utilisation séparée.

Conclusion

Cette thèse de doctorat a pour objectif principal de proposer de nouvelles méthodes d'apprentissage d'ensemble sur l'espace des matrices de covariance. Dans ce contexte, nous menons une classification supervisée sur la base de la métrique log-Euclidienne, où les matrices de covariance des caractéristiques CNN, ou bien des attributs multi-spectraux sont représentées par leurs vecteurs log-euclidiens. Nous avons évalué le potentiel de ces caractéristiques

de deuxième ordre, en comparaison avec les modèles basés sur le premier ordre, pour diverses applications, y compris la classification des scènes de télédétection et la classification des séries temporelles. Enfin, un intérêt particulier est accordé à l'évaluation du potentiel des images radar (Sentinel-1) et optique (Sentinel-2) sur une application sylvosanitaire, qui concerne la maladie de l'encre de châtaignier dans la forêt de Montmorency.

Introduction

The new generation of remote sensing imaging sensors enables high spatial, spectral and temporal resolution images with high revisit frequencies. These sensors allow the acquisition of multivariate time series such as spectral surface reflectance in several wavelengths. The availability of these multivariate time series has raised the interest of the remote sensing community to develop novel machine learning strategies for supervised classification. There are many algorithms dedicated to time series classification, where, given a set of time series with class labels, a model is trained to accurately predict the class of new time series. Classically, methods related to time series classification rely on either similarity measurement between time series, on the extraction of statistical features on each sub-sequences of time series, in order to form a new sequence as the input data [Atto *et al.* 2016, D'Urso & Maharaj 2012] or even on an ensemble learning based classifiers to improve classification robustness toward outliers. Most of these state-of-the-art approaches rely on first-order statistical features to model the information behind each time series. Features generated from first-order statistics provide information related to each point value distribution in the sequence. However, they do not give any information about the relative dependencies of the features within the same point, such as the dependencies between spectral attributes. Aiming at improving the accuracy and efficiency of time series classification, recent advances in methodological approaches have shown the relevance of using second-order descriptors for classification, indexing and segmentation on remote sensing applications [Pham *et al.* 2016, He *et al.* 2018], whether for 2-D data, such as images, or 1-D signals including time series. These include symmetric positive define matrices that describe the local second-order statistics of the time series. To illustrate the idea, Figure 7 shows an example of two time series for two different applications. The first represents an action of running where sensors record the x, y and z coordinates of the hand and knee movements for action recognition application. The second shows the temporal evolution of spectral reflectance and vegetation indices (R, G, B, NDVI, etc.) for recognizing a rice plantation for a remote sensing application.

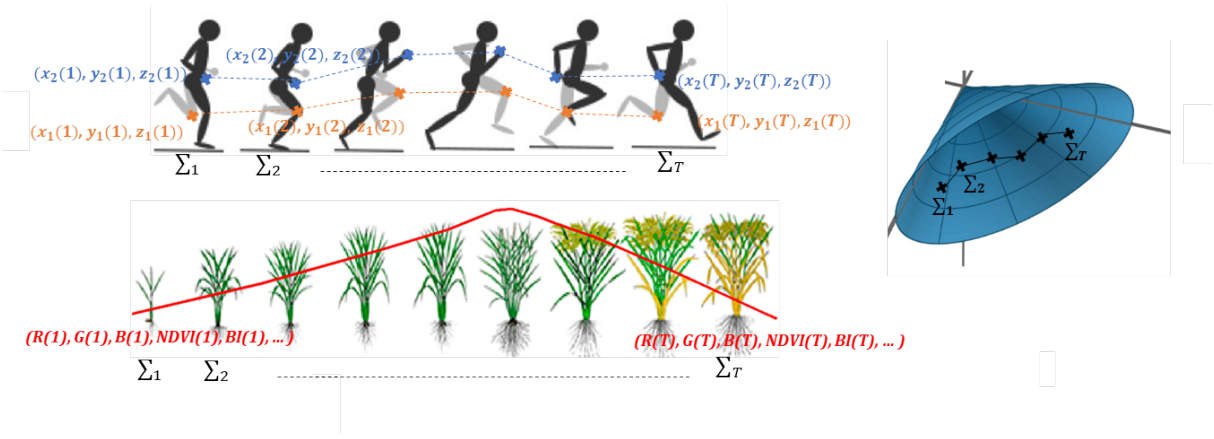


Figure 7: Examples of multivariate time series and covariance matrix trajectory.

As a person make an action, the arm and the knee may move in a correlated manner over time. Similarly in the bottom part of Figure 7, the temporal variation of spectral reflectance and vegetation indices of the rice crop may have a correlated behaviour, and this relationship may evolve with time. To capture those correlations between attributes, a covariance matrix Σ_t

is computed at each timestep. In the right side of Figure 7, the computed covariance matrices form a trajectory on the space of a particular curved shape, which is the interior of the convex blue cone. As such, the multivariate time series classification problem can be reformulated as a covariance matrix trajectory classification problem.

At the same time, the availability of large volumes of satellite image data and the emergence of new deep learning methods pose major challenges for the automatic interpretation of remote sensing dedicated to Earth observation. In this context, Long Short Term Memory networks (LSTMs) and Convolutional Neural Networks (CNNs) are capable of mining dynamical characteristics of time series, hence their success. However, training such models involves large labeled datasets. For datasets of relatively small dimension, the training from scratch of such deep neural networks is hardly conceivable. In image classification problems, an effective solution for limited training set consists of transfer learning. In that case, CNN models are considered as feature extractors. Classically, deep CNN models pre-trained on the ImageNet dataset are used [Russakovsky *et al.* 2014]. Then, features are extracted from a single or multiple layers, and are processed with some traditional machine learning algorithm. Those combinations between deep learning and traditional machine learning algorithms give rise to hybrid architectures, such as combining deep neural network architectures with Fisher vectors encoding strategies [Perronnin & Larlus 2015, Simonyan *et al.* 2013, Arandjelović & Zisserman 2013]. They enable to benefit from both families in favour of enhancing classification performance. Nevertheless, all these strategies generally exploit only a first-order representation of the CNN feature maps. Indeed, standard CNNs are generally composed by a pooling operator which calculates the mean (average pooling) or maximum (max pooling) value for each patch of the feature maps. To improve the capability of these networks, the second-order representation of the CNN feature maps has recently shown its interest [He *et al.* 2018]. It consists in a covariance pooling operator where the covariance matrix of the feature maps are extracted whether locally or globally. In the end, each image can be represented by a collection of covariance matrices which lie inside the blue cone of symmetric positive definite matrices depicted in Figure 7.

As discussed previously, for both applications on time series classification and image classification, we are interested in a problem of classification on the space of covariance matrices. The simplest approach is certainly to use the Frobenius norm. It consists in employing the Euclidean norm once covariance matrices are vectorized. While being extremely simple, it fails to take into account the geometric structure of these symmetric positive definite matrices. Actually covariance matrices lie in a constrained Euclidean space which is a Riemannian manifold. To handle these kind of data, some concepts of differential geometry are needed. Two Riemannian statistical frameworks can basically be employed. They are based respectively on the affine-invariant and log-Euclidean Riemannian metrics. For the former, computations are performed directly on the manifold whereas for the latter they are done on a tangent space. For that, covariance matrices are represented in the log-Euclidean space by the log map operator. In this log-Euclidean space, standard operations performed in an Euclidean space can be employed. From a practical point of view, Arsigny *et al.* have shown that affine-invariant and log-Euclidean frameworks perform better than the Frobenius one for the interpolation and regularization of their synthetic and clinical 3D diffusion tensor magnetic resonance imaging (DT-MRI) [Arsigny *et al.* 2006]. This has the advantage of more accurately capturing the underlying scatter of the data points, that are covariance matrices in our case, than is possible

with methods that treat data points as elements in a vector space. When comparing the log-Euclidean and affine-invariant Riemannian metrics, it can be observed that they offer several invariance properties (rotation, scaling, inversion). Moreover, in practice, they allow to obtain comparable results for a large variety of applications [Ilea *et al.* 2018b, Arsigny *et al.* 2006]. During this PhD thesis, the log-Euclidean metric is prioritized for its ease to use and its low computational complexity.

In this context, the main objective of this PhD thesis is to propose new ensemble learning methods on the space of covariance matrices. Based on the log-Euclidean representation of the covariance matrices of CNN features or multispectral attributes, we aim to evaluate the potential of these features for various applications including remote sensing scene classification and time series classification. A special interest is devoted to evaluate the potential of radar (Sentinel-1) and optical (Sentinel 2) images to the forestry problem of the chestnut ink disease in the Montmorency forest.

The work presented in this thesis is part of the CONFETTI[2] project funded by the Nouvelle Aquitaine region and the CNES TEMPOSS[3] project. These projects focus on the characterization of the forest structure and the monitoring of its changes, with a view to identify, map and follow sylvosanitary problems using Sentinel-1 and 2 time series. This work is also the subject of an international collaboration with the RIKEN[4] research centre in Japan as part of the PHC Sakura project, titled "Deep ensemble learning in big data era: from models to applications". The aim of this project is to combine ensemble learning strategies, which fuses multiple learners to improve prediction ability, and deep learning models in order to take advantage of their powerful ability. In addition, the PhD thesis is co-funded by the Nouvelle Aquitaine region and Bordeaux Sciences Agro.

Covariance matrices are fundamental tools in statistical signal processing and has been the subject of many studies. Since they lie in a Riemannian manifold, concepts of differential geometry are needed. In chapter 1, we introduce the basic notion of information geometry that are necessary to handle this kind of data. Two complete Riemannian statistical frameworks, based on the log-Euclidean and affine-invariant Riemannian metrics, are presented. Gaussian models are considered on both metric spaces, as well as their Gaussian mixture model extensions. For that, a Riemannian Gaussian model is presented, as well as a multivariate Gaussian distribution on the log-Euclidean space. When considering the log-Euclidean metric, covariance matrices are projected on the tangent space at a given reference point, classically fixed equal to the identity matrix. This may lead to some distortion if the covariance matrices are far away from this reference. To avoid this problem by limiting the distortion during the projection, we propose to consider a Gaussian mixture model (GMM) with multiple reference points, one for each component of the model. We also derive an expectation maximization algorithm to estimate the GMM parameters.

[2]CONFETTI: Characterization and monitoring of the forest ecosystem by multi-source and multi-temporal remote sensing images.
[3]TEMPOSS: Modeling of the temporal trajectory of Sentinel-1 & 2 observations for forest health monitoring.
[4]RIKEN Center for Advanced Intelligence Project (AIP), https://www.riken.jp/en/research/labs/aip/.

Since the general mathematical background for the log-Euclidean representation of a covariance matrix is assessed, and motivated by the success of deep neural networks and hybrid architectures, chapter 2 introduces two hybrid transfer learning approaches based on covariance pooling of CNN features [Akodad *et al.* 2018b, Akodad *et al.* 2019c]. These two methods use either local or global second-order representation of CNN features. The local approach, called the hybrid log-Euclidean Fisher vectors (Hybrid LE FV), relies on the covariance matrices extracted locally on the first layers of a CNN, which are then encoded by the Fisher vectors computed on their log-Euclidean representation. While for the global approach, namely the ensemble learning covariance pooling (ELCP), a single covariance matrix is computed on the feature maps of the deepest CNN layers. Moreover, in order to give more importance to the objects of interest present in the images, we proposed to use a covariance matrix weighted by the saliency information. It engenders the EL-SCP architecture. Next, in collaboration with the RIKEN Center for Advanced Intelligence Project, we propose to unify these works by presenting a transfer learning approach which benefit from both local and global aspects [Akodad *et al.* 2020c]. This ensemble learning approach, based on the most diverse ensembles, efficiently combines the provided decisions and allows to enhance the classification performance. To validate the methods, we consider different kinds of remote sensing datasets, including both aerial and satellite images. Some state-of-the-art databases of remote sensing scene classification are studied such as the UC Merced land use land cover, AID and SIRI-WHU dataset. Some applications are also done on textured images acquired from the Pleiades satellite for the differentiation of age classes of maritime pine stands, and the classification of oyster cultures in the Arcachon bay.

Regarding time series classification, the most simple way to compare two sequences of same length is by summing the ordered point-to-point distance between them. To do that, the commonly used distance function is the Euclidean distance [Bagnall *et al.* 2016b], which corresponds to the $\mathcal{L}_2$-norm. However, the Euclidean distance and its variants present several drawbacks. First, Euclidean distance is sensitive to signal transformations as time shifting which induces inaccurate results in certain applications. As an example, due to the intrinsic variability between fields, such as air temperature, soil drainage and other environmental characteristics, the temporal evolution of a certain crop in two different fields may have different temporal behaviour while providing the same information, and thus belonging to the same class. Second, the Euclidean distance suffers from invariance to re-parametrisation. It means that the distance between two series $\mathbf{x}_1$ and $\mathbf{x}_2$ is not preserved under any transformations. Given the above points, chapter 3 focuses on providing the following solutions. First, the distortion in the time axis can be addressed by the Dynamic Time Warping (DTW)[Sakoe & Chiba 1978, Berndt & Clifford 1994] and the re-parameterization invariance is solved by the use of square-root velocity function (SRVF) [Srivastava *et al.* 2011] representations of the considered time series. Furthermore, as the main contribution of the chapter is the proposition of classification models suitable with second-order matrix trajectories, the SRVF framework is extended to the transport square-root velocity function (TSRVF) representation [Su *et al.* 2014a] as a representation that provides a way to represent trajectories on Riemannian manifolds such that the re-parameterization invariance remains valid. To go further, in order to get benefit of the advantages of kernel methods, codebook based representations and ensemble learning strategies, Mikalsen *et al.* have introduced the time series cluster kernel (TCK) method in [Mikalsen *et al.* 2018] which has demonstrated competitive results for times series classification. In this work, we investigate the potential

of extending the TCK method to second-order matrix trajectories, namely the second-order TCK (SO-TCK) algorithm. All these approaches are validated on multivariate time series including applications on action recognition and crop classification based on temporal signatures.

Finally, forest health monitoring is a necessary step to ensure sustainable development and requires the gathering of information for forest conditions concerning the status of insect inhibiting plants, leaf defoliation and trunk damages. In that context, the focus of this study, in collaboration the UMR BIOGECO[5], is on the monitoring of diseases that destroy forests in France. Today, satellite remote sensing methods make it possible to detect clear cuts relatively well. However, since the seasonal foliar cycle is not necessarily well captured, it is generally difficult to detect thinning cuts and distinguish them from other silvicultural operations, as well as distinguishing different disease levels. In chapter 4, we focus on the specific application of the chestnut ink disease on the Montmorency forest. We aim to evaluate the potential of both radar and optical images acquired respectively by Sentinel-1 and Sentinel-2 sensors. We also investigate the interest of a multimodal approach by combining these two kinds of data. From a practical aspect, we first review some state-of-the-art vegetation and degradation forest indices that can be extracted from these images to monitor forest health. Next, based on the covariance pooling of these indices, we introduce an ensemble learning approach for the classification of the forest health status (healthy, declining, severely declining, and clear cut). Next, as the disease evolves continuously from healthy stands to completely destroyed trees, we propose to reformulate the problem as predicting a quantitative variable corresponding to a forest degradation (or health status) index. We show how the proposed classification model can be adapted to this regression problem. Based on it, we evaluate the potential of Sentinel-1 and Sentinel-2 data for this application.

Finally, the last chapter synthesizes the main conclusions of this work and presents some perspectives.

[5]BIOGECO: Biodiversity Genes and Communities, https://www6.bordeaux-aquitaine.inrae.fr/biogeco.

Chapter 1

Riemannian geometry and statistical modeling on the space of Symmetric Positive Definite (SPD) matrices

Contents

1.1 Introduction

The goal of a supervised classification algorithm is to assign an image to the appropriate class depending on its content. The basic technique involves extracting discriminative information within image data, called features. Then, a suitable classification method is applied to categorize the image into defined groups or classes. During the first stage, various kinds of features can be extracted such as color, gradient, shape, edge or textural information. Therefore, the major challenge is to consider image features which are highly distinctive and robust to different nuisances such as photometric or geometrical transformations. To this end, characterizing local image properties attracted a great research interest. Standard approaches are based on computing first-order statistics to model the information behind each image. If one takes a simple example, features generated from first-order statistics provide information related to the pixel value distribution on the image. However, they do not give any information about the relative dependencies of the features within the pixel, such as the dependencies between spectral attributes. To this aim, some authors have dedicated their works in exploiting the information behind second-order statistics using covariance matrix features. These statistics have proved to be highly effective in diverse classification tasks, including person re-identification, texture recognition, material categorization or electroencephalogram (EEG) signals classification in braincomputer interfaces to cite a few of them [Faraki *et al.* 2015a, Barachant *et al.* 2013, Said *et al.* 2015a].

Due to their specific properties, covariance matrices lie on a Riemannian manifold. In fact, conventional Euclidean tools are not adapted for covariance matrix manipulation since they are symmetric positive definite (SPD) matrices. To deal with covariance matrices geometry, other Riemannian metrics are usually considered. Two Riemannian metrics are generally employed: the log-Euclidean and the affine-invariant Riemannian metrics. When analyzing those two metrics, log-Euclidean and affine-invariant Riemannian metrics offer several invariance properties and permit to obtain comparable results for a large variety of applications [Ilea *et al.* 2018b, Arsigny *et al.* 2006] compared to the Euclidean metric. When considering the log-Euclidean metric, the tangent plane to the manifold is usually defined at the identity matrix. This may lead to distortions when covariance matrices are located far from this reference point. To avoid this problem, we will propose, latter in the chapter, to consider multiple reference points [Simo-Serra *et al.* 2017, Calinon & Jaquier 2019]. By limiting the distortion during the projection, this approach will permit a better modeling of the observed covariance matrices.

The second section of this chapter focuses on second-order statistics estimation. Starting from the usual sample covariance matrix, two main limitations are highlighted: sensitivity to outliers and run time of computation which is roughly proportional to the data set size. In order to faster covariance matrices computation, an integral image based method is presented [Tuzel *et al.* 2006] as well as the fixed point estimation algorithm [Tyler 1987] which permits to enhance estimation robustness regarding outliers.

Considering the specific geometry of SPD matrix space, the most common Riemmannian metrics are introduced in section 1.3. They allow to deal with the geometrical properties of the SPD matrix space, in particular the affine-invariant and log-Euclidean metrics. Thus, the second part provides some definitions that have been introduced in the literature.

Once the general framework is established for handling the specific geometry of covariance matrices, the focus in section 1.4 is on the statistical modeling on SPD matrix space. In fact, the measure induced by the Riemannian metric allows to define probability density functions. Two complete Riemannian statistical frameworks for characterising covariance matrix sample sets are thus detailed. They are defined on the affine-invariant (AI) and log-Euclidean (LE) metric spaces. Gaussian models are considered on both metric spaces, with their mixture extensions. The descriptive comparison of the two families of models are assessed.

Finally, the last part of section 1.4 constitutes the main contribution of the chapter, it focuses on a proposition of a Gaussian mixture model with multiple reference points, one for each cluster. This allows a better modeling by limiting the distortion when projecting the set of covariance matrices in the tangent space. The induced distortion is analyzed and quantified to address the effect of data projection on tangent plane against the distance between the identity matrix and the considered reference point.

1.2 Covariance matrix estimation

With the tremendous technological advancement and increase in computational power over the past decade, it becomes usual to extract and analyse high dimensional data in many fields ranging from economics and finance to biology, social networks, and health sciences [Donoho 2000]. Furthermore, estimating well-conditioned and large covariance matrices is an elementary problem in modern multivariate analysis. A large covariance matrix dimension causes an estimation problem which is generally challenging. In addition, the aggregation of massive amount of estimation errors can make considerable adverse impacts on the estimation accuracy. Therefore, estimating large covariance matrices attracts rapidly growing research attentions.

1.2.1 Sample covariance matrix

When there are complete observations, estimation of the covariance matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ based on a sample $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^d$ of N independent and identically distributed (iid) observations according to a multivariate Gaussian distribution is conventionally performed using the sample covariance matrix (SCM) estimator. The SCM estimator $\mathbf{C}$, under the condition that the number of observations is large enough compared to the number of features $N >> d$ to obtain a well-conditioned estimator, is defined by the classical maximum likelihood estimate as:

$$\mathbf{C} = \frac{1}{N} \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T, \tag{1.1}$$

where $\mathbf{x}_k$ is a $d-$dimensional observation, $\mu = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_k$ is the sample mean and $(.)^T$ is the transpose operator.

Therefore, when it comes to real data sets, they are often subject to measurement or recording errors and uncommon observations may also appear for a variety of reasons. Those uncommon observations are called outliers. In an other side, dealing with a large number of data may lead to high resource and time consumption. To circumvent this drawback, a fast covariance

computation method based on integral images is detailed in the following.

1.2.2 Integral image based method for fast covariance computation

At first, the covariance matrix as a region descriptor was proposed by Tuzel *et al.* [Tuzel *et al.* 2006], following the modeling methodology introduced in [Viola & Jones 2001]. The main idea is to perform a global modeling for the overall image region of interest. In the following, the principle of integral image method is explained and then applied to the computation of the sample covariance matrix (SCM) estimator.

1.2.2.1 Integral image principle

Integral images are an intermediate representation for the image allowing fast calculation of region sums where few operations per pixel are performed. The integral image at location (x', y') contains the sum of the pixels inside the rectangle bounded by the upper left corner pixel of the image and the pixel of the interest. For an image I, its integral image I' is given by:

$$I'(x', y') = \sum_{x<x', y<y'} I(x, y), \tag{1.2}$$

where $I'(x', y')$ is the integral image and $I(x, y)$ is the original image. Each point of the integral image I' corresponds to the summation of all point values inside the feature image rectangle of I bounded by the upper left corner and the point of interest. Following this latter representation, computation of any rectangular region can be performed in constant time. Actually, by the use of integral image, any rectangular sum can be computed in four array references as illustrated in Figure 1.1. The sum of pixels within rectangle R can be computed using the four reference

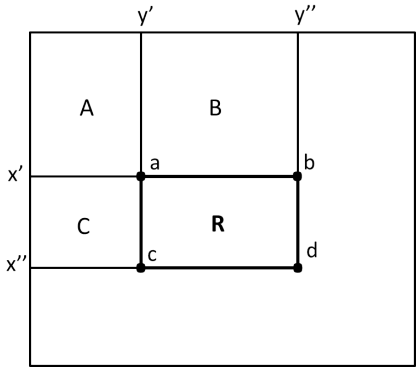


Figure 1.1: Integral Image. The rectangle R is defined by its upper left a and lower right corners d in the image.

arrays. The value of the integral image I' at location a is the sum of pixels inside the rectangle A and the value at location b is the sum of the pixels $A+B$, at location c is $A+C$ and at location d is $A+B+C+R$. Then the sum within R can be computed as $I'(d) + I'(a) - (I'(b) + I'(c))$.

1.2.2.2 Application to SCM computation

In order to speed up the SCM computation, the principle of integral images is exploited. Let's consider F a d dimensional feature image $W \times H \times d$ extracted from an image I. For a given rectangular region R belonging to F, let $\mathbf{x}_{k_{\{k=1\ldots N\}}}$ be the d-dimensional feature pixels inside. The $d \times d$ covariance matrix of the region R is computed using integral image representation [Tuzel *et al.* 2006]. The (i, j)-th element of the covariance matrix defined in (1.1) can be

rewritten as:

$$\mathbf{C}_R(i,j) = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x}_k(i) - \mu(i))(\mathbf{x}_k(j) - \mu(j)). \tag{1.3}$$

When rearranging the terms and expanding the mean, it becomes:

$$\mathbf{C}_R(i,j) = \frac{1}{N}\left[\sum_{k=1}^{N}\mathbf{x}_k(i)\mathbf{x}_k(j) - \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k(i)\sum_{k=1}^{N}\mathbf{x}_k(j)\right] \tag{1.4}$$

According to this latter expression, to find the covariance, the integral images $\mathbf{P}$ is constructed for the sum of all separate features $\mathbf{x}_k(i)_{i=1\ldots d}$ as well as integral images $\mathbf{Q}$ of the sum of the multiplication of any two feature combinations, $\mathbf{x}_k(i)\mathbf{x}_k(j)_{i,j=1\ldots d}$. This involves the construction of $d + d^2$ integral images: one for each feature $\mathbf{x}_k(i)$ and one for any two features $\mathbf{x}_k(i)\mathbf{x}_k(j)$. Considering $\mathbf{P}$ be the $W \times H \times d$ tensor of integral images and $\mathbf{Q}$ be the $W \times H \times d \times d$ the tensor of second order integral images, both defined as:

$$\mathbf{P}(x', y', i) = \sum_{x<x', y<y'} F(x, y, i) \text{ with } i = 1 \ldots d \tag{1.5}$$

$$\mathbf{Q}(x', y', i, j) = \sum_{x<x', y<y'} F(x, y, i)F(x, y, j) \text{ with } i, j = 1 \ldots d \tag{1.6}$$

The main powerful idea about integral images is that the computation is made in one pass over the image by taking advantage of the spatial arrangement of image pixels, which allows hastening the calculation. As such:

$$\mathbf{P}_{x,y} = [\mathbf{P}(x, y, 1) \ldots \mathbf{P}(x, y, d)]^T \tag{1.7}$$

$$\mathbf{Q}_{x,y} = \begin{pmatrix} \mathbf{Q}(x, y, 1, 1) & \ldots & \mathbf{Q}(x, y, 1, d) \\ . & & . \\ . & & . \\ . & & . \\ \mathbf{Q}(x, y, d, 1) & \ldots & \mathbf{Q}(x, y, d, d) \end{pmatrix}. \tag{1.8}$$

$\mathbf{P}_{x,y}$ is a d dimensional vector and $\mathbf{Q}_{x,y}$ is a $d \times d$ dimensional symmetric matrix. Then, due to this symmetry, only $d + (d^2 + d)/2$ passes are sufficient to compute both $\mathbf{P}$ and $\mathbf{Q}$. Then the complexity of constructing integral images is $O(d^2WH)$. If one consider the rectangular region R illustrated in Figure 1.1, where the upper left location a has coordinates (x', y') and the lower right location d is at coordinates (x'', y'') and according to (1.4), the covariance matrix of the region $R(x', y'; x", y")$ can be computed as:

$$\mathbf{C}_R(x', y'; x'', y'') = \frac{1}{n}[\mathbf{Q}_{x'',y''} + \mathbf{Q}_{x',y'} - \mathbf{Q}_{x'',y'} - \mathbf{Q}_{x',y''}$$
$$-\frac{1}{n}(\mathbf{P}_{x'',y''} + \mathbf{P}_{x',y'} - \mathbf{P}_{x'',y'} - \mathbf{P}_{x',y''})(\mathbf{P}_{x'',y''} + \mathbf{P}_{x',y'} - \mathbf{P}_{x'',y'} - \mathbf{P}_{x',y''})^T],$$

where $n = (x'' - x')(y'' - y')$ is the number of pixels inside the region $R(x', y'; x'', y'')$. Therefore, using integral images, the covariance of any rectangular region decreases the complexity to $O(d^2)$.

1.2.3 Fixed-point estimator algorithm

Due to its sensitivity to outliers, the use of the sample covariance matrix estimator may have a poor performance in many real-world applications. Real-world data are often large and contain a significant amount of aberrant data. As a consequence, a single erroneous observation can lead to a completely unreliable estimate. A way to circumvent the aforementioned problem is to use a robust covariance matrix estimator that preserves high performance even if the underlying distribution deviates from the Gaussian assumption. The fixed point estimator (FP), also known as the Tyler's estimator, has been introduced in [Tyler 1987] as one possible choice to solve the non robustness problem. The estimated covariance matrix $\hat{\mathbf{M}}$ is obtained through a recursive algorithm where it is the solution of the following fixed-point equation:

$$\hat{\mathbf{M}}_{\text{it}+1} = \frac{1}{N}\sum_{i=1}^{N} \frac{(\mathbf{x}_i - \hat{\mu}_{it})(\mathbf{x}_i - \hat{\mu}_{it})^T}{(\mathbf{x}_i - \hat{\mu}_{it})^T \hat{\mathbf{M}}_{\text{it}}^{-1}(\mathbf{x}_i - \hat{\mu}_{it})}, \tag{1.9}$$

where the mean μ_{it} is also estimated recursively by:

$$\hat{\mu}_{it+1} = \frac{\sum_{i=1}^{N} \frac{\mathbf{x}_i}{\left((\mathbf{x}_i - \hat{\mu}_{it})^T \hat{\mathbf{M}}_{\text{it}}^{-1}(\mathbf{x}_i - \hat{\mu}_{it})\right)^{1/2}}}{\sum_{i=1}^{N} \frac{1}{\left((\mathbf{x}_i - \hat{\mu}_{it})^T \hat{\mathbf{M}}_{\text{it}}^{-1}(\mathbf{x}_i - \hat{\mu}_{it})\right)^{1/2}}}, \tag{1.10}$$

where it is the iteration number. In practice, the existence and the uniqueness, up to a scalar factor, of the FP estimator of the normalized covariance matrix are established in [Pascal *et al.* 2008, Gini & Greco 2002], as well as the convergence of the recursive algorithm for any initialization. Therefore, the algorithm can be initialized with the identity matrix and converges in practice within 10 iterations. Regarding the scale factor, for any positive scalar $c \geq 0$, if $\hat{\mathbf{M}}$ is a solution of (1.9), $c\hat{\mathbf{M}}$ is also a solution. In practice, a normalization is performed such that $tr(\hat{\mathbf{M}}) = d$ where $tr(.)$ is the trace operator and d is the matrix dimension.

Contrary to the sample covariance estimator (SCM) which gives the same weight to all observations (1.1), this robust covariance matrix estimation using the FP algorithm, expressed in (1.9) allows to give a different weight to each observation $\mathbf{x}_i$. It thus permits to control the influence of aberrant observation in the estimation process.

1.3 SPD matrix space geometry

A covariance matrix is a square matrix, symmetric and positive semi-definite (SPD) of particular properties. In the following, the focus is on the two-dimensional case, but it can be easily generalized in higher dimension. According to the previous equations, the 2×2 covariance matrix is given by:

$$\mathbf{C} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix}. \tag{1.11}$$

The diagonal terms of the covariance matrix $\mathbf{C}$ are the variances of the features and the other entries are the covariances. For this reason, the covariance matrix is sometimes called the variance-covariance matrix. In fact, variance measures the variation of a single random variable, whereas covariance is a measure of how much two random variables $\mathbf{x}$ and $\mathbf{y}$ vary together. Since $\sigma_{xy} = \sigma_{yx}$, the covariance matrix is symmetric. Also, the covariance matrix $\mathbf{C}$ is positive

semi-definite (SPD), i.e., for $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbf{x}^T \mathbf{C} \mathbf{x} \geq 0, \tag{1.12}$$

and

$$\mathbf{C} - \mathbf{C}^T = 0, \tag{1.13}$$

where $(.)^T$ stands for the matrix transpose operator. As a consequence, equation (1.12) implies that:

$$\det(\mathbf{C}) \geq 0. \tag{1.14}$$

Due to those latter properties, namely the positivity of the eigenvalues and the constraint of the non-diagonal components, the symmetric positive semi-definite matrix $\mathbf{C}$ can be viewed as a point lying on a constrained Euclidean space. Also called a Riemannian manifold, the space has a particular curved shape which is the interior of a convex cone of $\mathbb{R}^3$ as illustrated in Figure 1.2. The symmetric strictly positive-definite matrices are located inside the cone whereas singular positive-semi-definite matrices with at least one null eigenvalue reside on the cone's surface.

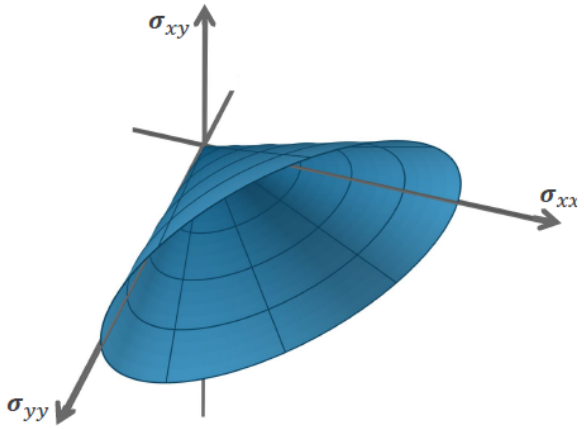


Figure 1.2: Convex cone of $\mathbb{R}^3$: space of 2×2 covariance matrices.

While Euclidean tools are well-suited when it comes to the characterization of objects lying on flat spaces, they do not take into consideration the curvature of the geometrical SPD space defined by covariance matrices. Actually, in the Euclidean space, the mean of SPD matrices is just the empirical average of SPD matrices which is not a complete characterization of these matrices. In fact, the cone of $\mathbb{R}^3$ is a differentiable manifold endowed with a Riemannian structure, where the base point-dependent inner product is defined by $< \mathbf{A}, \mathbf{B} >_{\mathbf{P}} = tr(\mathbf{P}^{-1}\mathbf{A}\mathbf{P}^{-1}\mathbf{B})$ for any elements $\mathbf{A}$ and $\mathbf{B} \in \mathcal{P}_d$ where $\mathcal{P}_d$ is the space of $d \times d$ SPD matrices.

As a consequence, applying standard Euclidean operations on covariance matrix data, for instance computing the Euclidean distance between two covariance matrices, are not adapted and may lead to undesirable results such as the swelling effect [Arsigny *et al.* 2006], which means that the determinant (and thus the dispersion) of the Euclidean mean can be larger than the original determinants of the two tensors being averaged. Tensor swelling occurs in tasks such as diffusion tensor interpolation, restoration or filtering of tensor-valued images [Castaño-Moraga *et al.* 2007]. In contrast, if one regards SPD matrices as points in Riemannian manifold and calculates their corresponding mean [Fréchet 1948, Karcher 1977], then the swelling effect disappears completely as illustrated in Figure 1.3 taken from the work of

[Pennec *et al.* 2006]. It shows a geodesic interpolation of two tensors according to the three frameworks : Euclidean, affine-invariant and Log-Euclidean metrics.

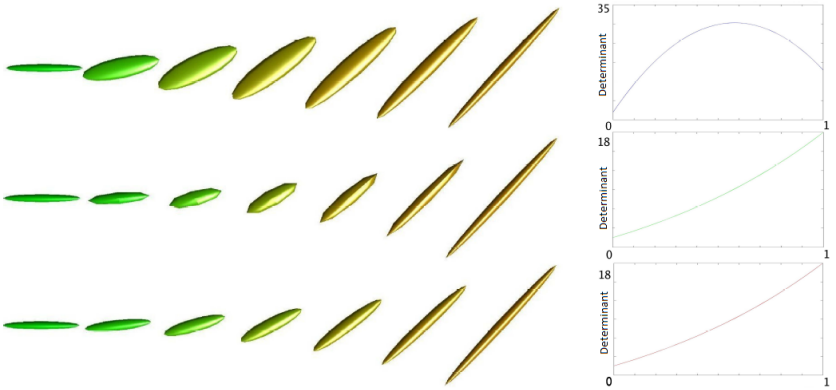


Figure 1.3: Geodesic interpolation of two tensors. **Left**: interpolated tensors. **Right**: graphs of the determinants of the interpolated tensors. **Top**: linear interpolation on coefficients. **Middle**: affine-invariant interpolation. **Bottom**: Log-Euclidean interpolation [Pennec *et al.* 2006].

As observed, the log-Euclidean and affine-invariant frameworks completely overcome the swelling effect which can be observed in the Euclidean case. This effect causes tensors to grow after a processing and it is shown through the growing volume of the ellipsoids, which is related to the determinant. However, in both Riemannian frameworks, determinants are monotonically interpolated.

1.3.1 Riemanian manifold

The geometry of non-Euclidean spaces gives rise to the notion of manifolds. Briefly, Euclidean geometry is the study of flat space whereas a differential manifold is a generalization of our basic understanding of a curved surface in an Euclidean space. An informal definition of this mathematical object, could be the following: a manifold is a space that is locally similar to the Euclidean space. This local similarity with the Euclidean space will appear to be very convenient, as it will allow us to extend all the tools of the Euclidean space to any differential manifold through mapping operations. Therefore, we will focus on this local resemblance to define and describe manifolds and functions allowing the transition from manifold to the local Euclidean space and vice versa.

Riemannian manifolds, also called elliptic geometry, are one of the non-Euclidean geometries which are smooth [1] and equipped with the Riemannian metric which allow one to measure geometric quantities such as distances and angles [Lee 1997]. In fact, it is a continuous collection of scalar products $(<,>_x)$ on each tangent space $\mathcal{T}_x\mathcal{M}$ at points x of the manifold. To actually perform calculation like distance on a manifold, few concepts are introduced in the following. The first one is the notion of geodesics and its relation with distance, which refers to the curves that are the shortest paths between two points. For example, straight lines in Euclidean space and great circles on a sphere. Then, the tangent plane definition is assessed, followed by operations of logarithm and exponential mapping allowing the transition from manifold to tangent space and vice versa.

[1] A smooth manifold is a differentiable manifold, also called a C_∞ or infinitely differentiable manifold.

1.3.1.1 Distance and geodesics

If we consider a curve $\gamma(t) : [0,1] \rightarrow \mathcal{P}_d$ on the manifold, the velocity vector $\dot{\gamma}(t)$ and its norm $||\dot{\gamma}(t)||$ the instantaneous speed, can be computed at each point. To compute the length of the curve, the norm is integrated along the curve.

$$\mathbb{L}(\gamma) = \int_0^1 ||\dot{\gamma}(t)|| \mathrm{d}t = \int_0^1 (< \dot{\gamma}(t), \dot{\gamma}(t) >_{\gamma(t)})^{\frac{1}{2}} \mathrm{d}t \tag{1.15}$$

In the case of SPD matrices, the unique geodesic parameterized by the length, $t \rightarrow \gamma(t)$, joining two covariance matrices $\mathbf{X}$ and $\mathbf{Y}$, is defined as [James 1973]:

$$\gamma(t) = \mathbf{X}^{\frac{1}{2}} (\mathbf{X}^{-\frac{1}{2}} \mathbf{Y} \mathbf{X}^{-\frac{1}{2}})^t \mathbf{X}^{\frac{1}{2}}, \tag{1.16}$$

The geodesic distance $d(\mathbf{X}, \mathbf{Y}) : \mathcal{P}_d \times \mathcal{P}_d \rightarrow \mathbb{R}^+$, or the Rao's distance [Terras 1988], is equal to the minimum length connecting the two points $\mathbf{X}$ and $\mathbf{Y}$ on the manifold among the smooth curves.

$$d(\mathbf{X}, \mathbf{Y}) = min_\gamma \mathbb{L}(\gamma), \quad \text{with} \quad \gamma(0) = \mathbf{X} \ \text{ and } \ \gamma(1) = \mathbf{Y}. \tag{1.17}$$

Then, for SPD matrices, the geodesic distance $d(\mathbf{X}, \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ is given by:

$$d^2(\mathbf{X}, \mathbf{Y}) = tr \left[\mathrm{logm}^2(\mathbf{X}^{-1/2} \mathbf{Y} \mathbf{X}^{-1/2}) \right] = \sum_{i=1}^{d} \log^2(\lambda_i), \tag{1.18}$$

where logm stands for the matrix logarithm function and λ_i is the i^{th} eigenvalue of the matrix $\mathbf{X}^{-1/2} \mathbf{Y} \mathbf{X}^{-1/2} \in \mathcal{P}_d$.

1.3.1.2 Tangent vectors and tangent space $\mathcal{T}_x\mathcal{M}$

In differential geometry, one can attach to every point x of a differentiable manifold a tangent space which is a real vector space that contains all possible directions in which one can tangentially passes through x. Intuitively, when walking along a curve on a smooth manifold, as one pass through the point x, it implicitly has velocity (magnitude and direction) that is tangent to the manifold, in other words: a tangent vector. Then, all tangent vectors at x are elements of the tangent space at the same point x. This is a generalization of the notion of a bound vector in a Euclidean space where the tangent space $\mathcal{T}_x\mathcal{M}$ at a point x on an n-dimensional manifold $\mathcal{M}$ is an n-dimensional hyperplane that best approximates $\mathcal{M}$ around x as illustrated in Figure 1.4. In addition, the dimension of the tangent space at every point of a connected manifold is the same as that of the manifold itself. The tangent space of a manifold $\mathcal{M}$ at a point x is noted $\mathcal{T}_x\mathcal{M}$. An element $y \in \mathcal{T}_x\mathcal{M}$ is called a tangent vector. Generally, the point x is also called a reference point and noted $\mathbf{M}_{ref}$.

1.3.1.3 Logarithmic and exponential mapping

The notions of matrix logarithm and exponential are central in the theoretical framework presented here. Let x be a point of the manifold that is a reference point and $\overrightarrow{xy}$ a vector of the tangent space $\mathcal{T}_x\mathcal{M}$ at that point.

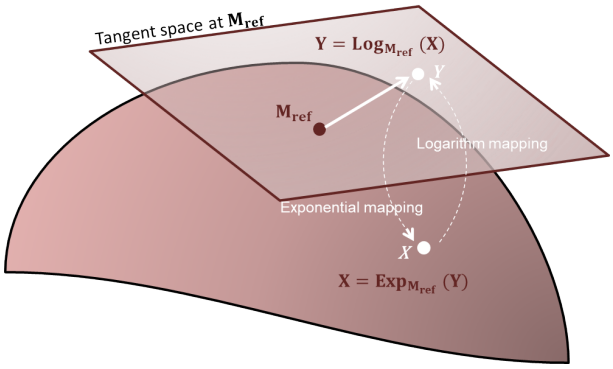


Figure 1.4: Illustration of a tangent space, tangent vectors and Logarithm/Exponential mapping.

As in the scalar case, the logarithm mapping is defined as the inverse of the exponential. The important point here is that the logarithm of an SPD matrix is well defined and is also a symmetric matrix. Conversely, the exponential of any symmetric matrix yields an SPD matrix. This means that under the matrix exponentiation operation, there is a one to one correspondence between symmetric matrices and covariance matrices as illustrated in Figure 1.4 where $\mathbf{M}_{ref} = x$. The inverse mapping from $\mathcal{M}$ to $\mathcal{T}_x\mathcal{M}$ is called the logarithmic map which allows unfolding the manifold in the tangent space along geodesics.

Regarding the space $\mathcal{P}_d$ of symmetric positive matrices (SPD), the exponential map and its inverse map onto the tangent vector space at a given reference matrix $\mathbf{M}_{ref}$ are respectively defined in a closed form as [Pennec *et al.* 2006]:

$$\mathrm{Exp}_{\mathbf{M}_{ref}} = \begin{cases} \mathcal{T}_{\mathbf{M}_{ref}}\mathcal{M} \to \mathcal{P}_d \\ \mathbf{Y} \to \mathrm{Exp}_{\mathbf{M}_{ref}}(\mathbf{Y}) = \mathbf{M}_{ref}^{\frac{1}{2}}\ \mathrm{expm}(\mathbf{M}_{ref}^{-\frac{1}{2}}\mathbf{Y}\mathbf{M}_{ref}^{-\frac{1}{2}})\mathbf{M}_{ref}^{\frac{1}{2}} \\ \qquad\qquad\qquad = \mathbf{M}_{ref}\ \mathrm{expm}(\mathbf{M}_{ref}^{-1}\mathbf{Y}) \end{cases} \tag{1.19}$$

$$\mathrm{Log}_{\mathbf{M}_{ref}} = \begin{cases} \mathcal{P}_d \to \mathcal{T}_{\mathbf{M}_{ref}}\mathcal{M} \\ \mathbf{X} \to \mathrm{Log}_{\mathbf{M}_{ref}}(\mathbf{X}) = \mathbf{M}_{ref}^{\frac{1}{2}}\ \mathrm{logm}(\mathbf{M}_{ref}^{-\frac{1}{2}}\mathbf{X}\mathbf{M}_{ref}^{-\frac{1}{2}})\mathbf{M}_{ref}^{\frac{1}{2}} \\ \qquad\qquad\qquad = \mathbf{M}_{ref}\ \mathrm{logm}(\mathbf{M}_{ref}^{-1}\mathbf{X}). \end{cases} \tag{1.20}$$

Note that logm and expm are matrix logarithm and exponential, respectively. Both are explained further. As a direct consequence, the re-interpretation of addition and subtraction using logarithmic and exponential maps is very powerful to generalize algorithms working on vector spaces to algorithms on Riemannian manifolds such as distance, mean and gradient descent computation, as summarized in the following Table 1.1.

	Euclidean space	Riemannian manifold
Substraction	$\vec{xy} = y - x$	$\vec{xy} = \mathrm{Log}_x(y)$
Addition	$y = x + \vec{xy}$	$y = \mathrm{Exp}_x(\vec{xy})$
Distance	$\mathrm{d}(x,y) = \|\|y - x\|\|$	$\mathrm{d}(x,y) = \|\|\vec{xy}\|\|_x$
Mean value	$\sum_i \vec{\overline{x}x_i} = 0$	$\sum_i \mathrm{Log}_{\overline{x}}(x_i) = 0$
Gradient descent	$x_{t+\varepsilon} = x_t - \varepsilon\nabla C(x_t)$	$x_{t+\varepsilon} = \mathrm{Exp}_{x_t}(-\varepsilon\nabla C(x_t))$

Table 1.1: Re-interpretation of basic standard operations in a Riemannian manifold.

Indeed, in the Riemannian manifold, a vector $\vec{xy}$ (attached at the point x) can be seen as a vector of the tangent space at the same point x. Such a vector can be identified to a point in a manifold using the exponential map: $y = \text{Exp}_x(\vec{xy})$ and conversely, the logarithmic mapping can be used to map two points (x, y) into a vector: $\vec{xy} = \text{Log}_x(y)$. Those tools are very powerful in terms of implementation since all geometric operations can be expressed, for instance the mean value and the gradient descent algorithm.

1.3.1.4 Parallel transport

In Riemanian geometry, parallel transport, also called affine transformation, is a technique which permits transporting geometrical data along smooth curves and thereby linking tangent spaces in a manifold by preserving the relationship between data. Let $\mathbf{X}$ and $\mathbf{Y}$ two matrices of the manifold $\mathcal{M}$, the explicit expression of the parallel transport from $\mathbf{X}$ to $\mathbf{Y}$ of any $\mathbf{S} \in \mathcal{T}_{\mathbf{X}}\mathcal{M}$ is defined in [Pennec *et al.* 2006] by:

$$\begin{aligned} \mathcal{T}_{\mathbf{X}}\mathcal{M} &\rightarrow \mathcal{T}_{\mathbf{Y}}\mathcal{M} \\ \mathbf{S} &\rightarrow \Gamma_{\mathbf{X}\rightarrow\mathbf{Y}}(\mathbf{S}) = \mathbf{E}^T\ \mathbf{S}\ \mathbf{E}, \end{aligned} \tag{1.21}$$

with $\mathbf{E} = (\mathbf{Y}\mathbf{X}^{-1})^{\frac{1}{2}}$. As illustrated in Figure 1.5, the parallel transportation allows to compare vectors locally on a Riemannian manifold such as the comparison of probability density functions, coordinates or vectors that are defined in tangent spaces at different points on the manifold.

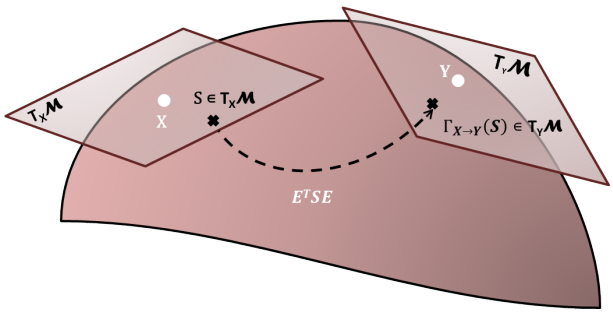


Figure 1.5: Illustration of parallel transport on manifold from $\mathcal{T}_{\mathbf{X}}\mathcal{M}$ to $\mathcal{T}_{\mathbf{Y}}\mathcal{M}$ with $\mathbf{E} = (\mathbf{Y}\mathbf{X}^{-1})^{\frac{1}{2}}$.

Furthermore, parallel transport of SPD matrices on manifold $\mathcal{M}$ from $\mathbf{X}$ to $\mathbf{Y}$ is given the same transformation (1.21) applied to $\mathbf{P} = \text{Exp}_{\mathbf{X}}(\mathbf{S})$ such that:

$$\text{Exp}_{\mathbf{Y}}\left(\Gamma_{\mathbf{X}\rightarrow\mathbf{Y}}(\mathbf{S})\right) = \mathbf{E}^T\ \mathbf{P}\ \mathbf{E}, \tag{1.22}$$

with $\mathbf{E} = (\mathbf{Y}\mathbf{X}^{-1})^{\frac{1}{2}}$.

Since the fundamental tools of the Riemannian manifold have been defined, the focus in the following is on the affine-invariant and log-Euclidean statistical frameworks which allow to consider the Riemannian geometry characteristics of the space $\mathcal{P}_d$ of $d \times d$ symmetric and positive definite (SPD) matrices. The choice for these two metrics is justified by their strong invariance properties compared to the Euclidean metric. Thus, they provide the most faithful representations of data lying on Riemannian manifold.

For the following sections, in order to simplify notations, the tangent space to the manifold at a reference point $\mathcal{T}_{\mathbf{M}_{ref}}\mathcal{M}$ will be noted $\mathcal{T}_{\mathbf{M}_{ref}}$.

1.3.2 Affine-invariant (AI) metric

Many authors have raised the need of intrinsic tools to analyze covariance matrices [Smith 2005, Pennec 2006, Arsigny *et al.* 2006]. As pointed out by Pennec *et al.* in [Pennec *et al.* 2006], one of the most popular Riemannian metrics is the affine-invariant one, called also Fisher-Rao. The curvature of the manifold is taken into consideration and thus the affine-invariant Riemannian metric enjoys desirable invariance properties compared to the Euclidean metric. The affine-invariant Riemannian distance has the property of being invariant by affine transformations [Pennec 2006]. This means that for any matrices $\mathbf{X}$ and $\mathbf{Y} \in \mathcal{P}_d$ and any invertible real matrix $\mathbf{A}$ of size $d \times d$, the following property holds:

$$d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{A}^T\mathbf{X}\mathbf{A}, \mathbf{A}^T\mathbf{Y}\mathbf{A}), \tag{1.23}$$

where $d(\mathbf{X}, \mathbf{Y})$ is the geodesic distance between $\mathbf{X}$ and $\mathbf{Y}$ given in (1.18). For more details on the geometric properties induced by the Rao-Fisher metric on the space $\mathcal{P}_d$, see [Said *et al.* 2015b, Said *et al.* 2018].

Although this metric have excellent theoretical properties and allows to develop precise and robust processing tools, it also leads in practice to complex and slow algorithms due to the high computational costs. Indeed, it generally involves recursive algorithms. To illustrate that, we consider the example of a sample's center of mass computation. It usually relies on a squared Euclidean distance [MacQueen 1967]. Since this distance is not adapted to the Riemmannian geometry, the Euclidean distance is replaced by the Riemannian distance. In fact, considering a random sample $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_N\}$ of N SPD matrices of size $d \times d$, characterized by its central value $\bar{\mathbf{M}}$, the estimated centroid, $\hat{\bar{\mathbf{M}}}$, is obtained by minimizing the following cost function $f(\bar{\mathbf{M}})$:

$$\hat{\bar{\mathbf{M}}} = \underset{\bar{\mathbf{M}} \in \mathcal{P}_d}{\arg\min}\, f(\bar{\mathbf{M}}). \tag{1.24}$$

In the case of the center of mass, also known as the Fréchet [Fréchet 1948] or Karcher mean [Karcher 1977], it is obtained by minimizing the sum of squared distances between $\bar{\mathbf{M}}$ and the observations $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_N\}$. The cost function is therefore defined by:

$$f(\bar{\mathbf{M}}) = \frac{1}{N}\sum_{n=1}^{N} d^2(\mathbf{M}_n, \bar{\mathbf{M}}), \tag{1.25}$$

where $d(.)$ is the Riemannian geodesic distance introduced in (1.18).

To solve this optimization problem, a gradient-based algorithm is often proposed for the estimation of the center of mass $\bar{\mathbf{M}}$ [Absil *et al.* 2008, Lenglet *et al.* 2006]. Starting from (1.25), the center of mass is recursively estimated using the following expression:

$$\bar{\mathbf{M}}_{it+1} = \mathrm{Exp}_{\bar{\mathbf{M}}_{it}}\left(-\alpha_{it}\nabla f(\bar{\mathbf{M}}_{it})\right), \tag{1.26}$$

where α_{it} is the descent step and $\mathrm{Exp}_{\mathbf{M}}$ the exponential map given in (1.20). Moreover, the

Armijo's backtracking procedure [Armijo 1966] is classically used to fix the step α_{it} at each iteration it. The procedure is repeated as long as the norm of $\nabla f(\bar{\mathbf{M}}_{it})$, noted D_{it} remains greater than a predefined precision parameter ε, or until a maximum number of iterations N_{iter} is reached. Moreover, the gradient of the cost function (1.25) with respect to $\bar{\mathbf{M}}$ is defined as:

$$\nabla f(\bar{\mathbf{M}}) = -\frac{2}{N}\sum_{i=1}^{N} \mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{M}_i), \tag{1.27}$$

with $\mathrm{Log}_{\bar{\mathbf{M}}}(.)$ the Riemannian logarithm mapping defined in (1.20). Algorithm 1 presents a pseudo-code describing the entire procedure for recursively estimating the centroid.

Algorithm 1 Center of mass estimator - Fréchet/Karcher mean

Input: $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ a set of covariance matrices where $\mathbf{M}_n \in \mathcal{P}_d$, ε, N_{iter}
Initialize: $\bar{\mathbf{M}}$ using the sample mean

1: $it \leftarrow 1$
2: **while** $(it \leq N_{iter})$ and $(D_{it} > \varepsilon)$ **do**
3: Estimate $\bar{\mathbf{M}}$ by using (1.26).
4: Compute the gradient norm D_{it}.
5: $it \leftarrow it + 1$
6: **end while**

Output: $\bar{\mathbf{M}} \in \mathcal{P}_d$.

As observed, affine-invariant computations involve an intensive use of matrix inverses, square roots, logarithms, and exponentials. It thus results on slow algorithms which may be critical in practice. To remedy this limitation, a new family of Riemannian metrics called Log-Euclidean is introduced in [Arsigny *et al.* 2006]. In fact, they also have excellent theoretical properties and yield similar results in practice, but with much simpler and faster computations.

1.3.3 Log-Euclidean (LE) metric

The log-Euclidean metric was proposed by Arsigny et al. in [Arsigny *et al.* 2006], as an interesting alternative to the affine-invariant metric. Although the log-Euclidean metric does not yield full affine invariance, it is invariant by similarity (orthogonal transformation and scaling). This means that computations using this metric will be invariant with respect to a change of coordinates obtained by a similarity. From a practical point of view, Arsigny *et al.* have shown in [Arsigny *et al.* 2006] that affine-invariant and log-Euclidean frameworks perform better than the Euclidean one for the interpolation and regularization of their synthetic and clinical 3D diffusion tensor magnetic resonance imaging (DT-MRI) data. This has the advantage of more accurately capturing the underlying scatter of the data points (that are covariance matrices) than is possible with methods that treat data points as elements in a vector space. For many applications, the log-Euclidean framework has shown competitive results compared to the affine-invariant Riemannian one [Ilea *et al.* 2018b, Arsigny *et al.* 2006]. This log-Euclidean framework is considered in this work for its efficiency and ease of use. In fact, it principally involves the notions of matrix logarithm and exponential which are central in the theoretical framework presented here. Actually, the use of logarithm matrix permits to locally flatten the manifold via the tangent space approximation. This consists of projecting each covariance matrix $\mathbf{X}$ onto a common tangent space of this manifold at the reference point $\mathbf{M}_{ref}$ via the log map

operator [Faraki *et al.* 2015b, Arsigny *et al.* 2006, Rosu *et al.* 2017] defined as:

$$\mathbf{X}^{\mathcal{T}_{\mathbf{M}_{ref}}} = \mathrm{Log}_{\mathbf{M}_{ref}}\mathbf{X} \tag{1.28}$$

$$= \mathbf{M}_{ref}\mathrm{logm}\left(\mathbf{M}_{ref}^{-1}\mathbf{X}\right). \tag{1.29}$$

$\mathbf{X}^{\mathcal{T}_{\mathbf{M}_{ref}}}$ means that covariance matrix $\mathbf{X}$ is projected on the tangent space at the reference point $\mathbf{M}_{ref}$. Then, if one is interested in using a covariance matrix as a feature descriptor in a classifier, a natural choice consists in vectorizing it in order to process this quantity as a vector and use any vector-based classification algorithms. Due to symmetry, the following modified half-vectorization operator Vec(.) stacks, with appropriate weighting, the upper triangular part of $\mathbf{P}$ into a column vector such that:

$$\mathrm{Vec}(\mathbf{P}) = \left[P_{11}, \sqrt{2}P_{12}, \ldots, \sqrt{2}P_{1d}, P_{22}, \sqrt{2}P_{23}, \ldots, P_{dd}\right], \tag{1.30}$$

A coefficient of $\sqrt{2}$ is applied on the off-diagonal elements of $\mathbf{P}$ in order to conserve equality of norms $||\mathbf{P}||_F = ||\mathrm{Vec}(\mathbf{P})||_2$, where $||.||_F$ is the Frobenius norm. The reverse operation is defined in a straightforward manner by an operator denoted unVec(x). P_{ij} are the elements of $\mathbf{P}$ at row i and column j. Those two operations yield to the definition of the log-Euclidean vector representation of $\mathbf{X}$ computed at the reference point $\mathbf{M}_{ref}$, denoted $\mathbf{x}^{\mathcal{T}_{\mathbf{M}_{ref}}} \in \mathbb{R}^{\frac{d(d+1)}{2}}$ where :

$$\mathbf{x}^{\mathcal{T}_{\mathbf{M}_{ref}}} = \mathrm{Vec}\left(\mathbf{X}^{\mathcal{T}_{\mathbf{M}_{ref}}}\right) = \mathrm{Vec}\left(\mathrm{Log}_{\mathbf{M}_{ref}}(\mathbf{X})\right). \tag{1.31}$$

These covariance matrices are projected on the tangent space at $\mathbf{M}_{ref}$; they lie in a vector space where conventional image processing and machine learning methods can be used. Within this framework, the tangent space is computed at a reference point $\mathbf{M}_{ref}$ as shown in (1.28). Different choices can be made for this reference point, such as the identity matrix, the center of mass or the median. The use of the identity matrix $\mathbf{I}_d$ for this latter is undoubtedly the simplest and the most usual and adopted way to map covariance matrices on the tangent space. This choice will be made for the following as long as no additional clarification is added. In that case, the log map operator in Equation (1.28) vanishes to:

$$\mathrm{Log}_{\mathbf{I}_d}(\mathbf{X}) = \mathrm{logm}(\mathbf{X}). \tag{1.32}$$

This consists of computing the ordinary matrix logarithm. Let $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ be the eigenvalue decomposition of an SPD matrix, the logarithm is defined as: $\mathrm{logm}(\mathbf{A}) = \mathbf{V}\mathrm{logm}(\mathbf{D})\mathbf{V}^T$. Since $\mathbf{D}$ is the diagonal matrix of eigenvalues, $\mathrm{logm}(\mathbf{D})$ is also a diagonal matrix whose diagonal elements are the logarithm of the eigenvalues.

1.3.4 Comparison between the affine-invariant and log-Euclidean frameworks

The invariance properties of the LE and AI metrics are synthesized in Table 1.2. It demonstrates the strength of the Rao's distance invariance properties compared to the log-Euclidean one. Let $\mathbf{X}$ and $\mathbf{Y}$ be two covariance matrices of size $d \times d$ and $d(.)$ a distance measure between the two matrices $\mathbf{X}$ and $\mathbf{Y}$ according to each metric.

	Euclidean	Log-Euclidean	Affine-Invariant
Rotation and reflection	✓	✓	✓
Scaling	×	✓	✓
Inversion	×	✓	✓
Affine invariance	×	×	✓

Table 1.2: Invariance properties comparison between Euclidean, LE and AI metrics

The different invariance properties are explained in the following.

1. Rotation and reflection invariance

$$d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{R}\mathbf{X}\mathbf{R}^T, \mathbf{R}\mathbf{Y}\mathbf{R}^T), \tag{1.33}$$

where $\mathbf{R} \in \mathrm{O}_p$ is a rotation and reflection matrix belonging to the group O_p of real orthogonal matrices of size $d \times d$.

2. Scaling invariance

$$d(\mathbf{X}, \mathbf{Y}) = d(\alpha\mathbf{X}, \alpha\mathbf{Y}), \tag{1.34}$$

with $\alpha > 0$.

3. Invariance under inversion

$$d(\mathbf{X}, \mathbf{I}_d) = d(\mathbf{X}^{-1}, \mathbf{I}_d), \tag{1.35}$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix.

4. Invariance under affine transformation

$$d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{A}\mathbf{X}\mathbf{A}^T, \mathbf{A}\mathbf{Y}\mathbf{A}^T), \tag{1.36}$$

with $\mathbf{A}$ a full rank $d \times d$ matrix.

As observed, although the AI metric space is endowed with stronger invariance properties compared to the LE metric, estimating the parameters of the statistical models relies on recursive estimation algorithms, inducing thus high computational expenses. In contrast, as the LE mapping allows the transformation to a vector-form representation of covariance matrices, the complexity and computational expenses of the algorithms on the LE metric space are significantly reduced. Moreover, the performance evaluation of different metrics have been assessed on both real and simulated SPD data samples in [Arsigny *et al.* 2006]. The non-Euclidean metrics showed, in a global point of view, similar performance and outperform the Euclidean framework in several image processing applications involving SPD matrices. For example, in the context of visual objects categorisation, Jayasumana et al. in [Jayasumana *et al.* 2013] demonstrated the benefits of non-Euclidean based approaches over their Euclidean counterparts in terms of classification performances.

1.4 Statistical modelling on the SPD space

As we have seen above, in response to the need for effective methods of processing data which lie in the space $\mathcal{P}_d$, extensive use of SPD matrices and attention has been given to metrics

and distance measurement on the Riemannian manifold. However, only few studies are dedicated to assess a formal and adapted statistical modeling for using a set of covariance matrices on the manifold of SPD matrices which is able to represent the statistical variability of data in $\mathcal{P}_d$.

Pennec [Pennec *et al.* 2006] and Lenglet et al. [Lenglet *et al.* 2006] have proposed statistical models by considering the geometry of covariance matrices. Namely, the Gaussian distributions on the Riemannian manifold, relying on the affine-invariant metric, has been introduced. Despite of being well adapted to the data geometry, these propositions rely on a particular case of compact distributions. These limitations are overcomed by Said et al. in [Said *et al.* 2015b] where, opposed to the previous propositions, the computation of the normalization constant of the probability density function is achieved, leading thus to an exact expression of the Riemannian Gaussian distribution. The distribution is characterized by its central value given by the Riemannian center of mass and its dispersion around this central value.

The goal of this section is to generalize statistical models and perform statistical inferences on the Riemannian manifold of the space of SPD matrices. It has three major parts. In the first part, the definition of a Riemannian and log-Euclidean Gaussian model are assessed on the space of covariance matrices as well as the methods for parameter estimation. Then the extension to their corresponding Gaussian mixture model (GMM) of K components is detailed. Finally, after assessing comparison between the two frameworks, the last part constitutes the main contribution of the chapter, it focuses on a proposition of an alternative Gaussian mixture model with multiple reference points, one for each cluster in order to better fit the data samples while preserving the theoretical advantages of AI and LE metrics.

For this section, a sample $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_N\}$ of N independent and identically distributed (i.i.d) observations is considered.

1.4.1 Riemannian affine-invariant Gaussian distribution

1.4.1.1 Model definition

A Riemannian Gaussian distribution (RGD) depends on two parameters, $\bar{\mathbf{M}} \in \mathcal{P}_d$ and $\sigma > 0$. It is defined by its probability density function:

$$p(\mathbf{M}_n|\bar{\mathbf{M}}, \sigma) = \frac{1}{Z(\sigma)} \exp\left[-\frac{d^2(\mathbf{M}_n, \bar{\mathbf{M}})}{2\sigma^2}\right], \tag{1.37}$$

where $d(.)$ is the Riemannian geodesic distance defined in (1.18), $\bar{\mathbf{M}}$ and σ represents respectively the central value (centroid) and the dispersion. A first important issue for a complete description of (1.37) is the explicit definition of $Z(\sigma)$. $Z(\sigma)$ is a normalization factor independent of $\bar{\mathbf{M}}$ [Said *et al.* 2015b], given by:

$$Z(\sigma) = \int_{\mathcal{P}_d} \exp\left[-\frac{d^2(\mathbf{M}, \bar{\mathbf{M}})}{2\sigma^2}\right] dv(\mathbf{M}), \tag{1.38}$$

where $dv(\mathbf{M}_n)$ is the Riemannian volume element. As explained in [Said *et al.* 2015b], an analytical expression can be derived only for $d = 2$,

$$Z(\sigma) = (2\pi)^{3/2}\sigma^2 \exp\left(\frac{\sigma^2}{4}\right) erf\left(\frac{\sigma}{2}\right), \tag{1.39}$$

where $erf(.)$ stands for the error function.
While for $d > 2$, the normalization factor $Z(\sigma)$ is given by;

$$Z(\sigma) = q_d \times \int_{\mathbb{R}^d} \exp\left(-\frac{|r|^2}{2\sigma^2}\right) \prod_{i<j} \sinh \frac{|r_i - r_j|}{2} dr_1 \dots dr_d, \tag{1.40}$$

where $|r| = (r_1^2 + \dots + r_d^2)^{\frac{1}{2}}$ and q_d is given by:

$$q_d = \frac{1}{d!} \frac{\pi^{\frac{d^2}{2}}}{\Gamma_d(\frac{d}{2})} 8^{\frac{d(d-1)}{4}}. \tag{1.41}$$

$\Gamma_d(.)$ is the multivariate Gamma function [Muirhead 1982], defined as:

$$\Gamma_d(y) = 8^{\frac{d(d-1)}{4}} \prod_{j=1}^{d} \Gamma\left(y + \frac{1-j}{2}\right), \tag{1.42}$$

and $\Gamma(.)$ is the usual Gamma function. The evaluation of $Z(\sigma)$ is necessary for distribution parameter estimation, in particular for the estimate of σ. In practice, $Z(\sigma)$ can be tabulated using a Monte Carlo integration. For more information, the interested reader is referred to [Said *et al.* 2015b]. Moreover, in that paper, an algorithm is proposed to generate samples from this distribution.

1.4.1.2 Parameter estimation

The RGD's parameters, the dispersion and the central value of the probability density function, $\bar{\mathbf{M}}$, can both be estimated through the maximum likelihood estimation (MLE). The log-likelihood function is given by:

$$\mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}, \sigma) = \log \prod_{n=1}^{N} p(\mathbf{M}_n|\bar{\mathbf{M}}, \sigma) \tag{1.43}$$

$$= -N \log Z(\sigma) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} d^2(\mathbf{M}_n, \bar{\mathbf{M}}) \tag{1.44}$$

This leads to the maximum-likelihood estimate $\hat{\bar{\mathbf{M}}}$ of the Riemannian center of mass $\bar{\mathbf{M}}$, also known as the Fréchet [Fréchet 1948] or Karcher mean [Karcher 1977] and is obtained by minimizing the sum of squared distances between $\bar{\mathbf{M}}$ and the observations $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_N\}$:

$$\hat{\bar{\mathbf{M}}} = \arg\max_{\bar{\mathbf{M}} \in \mathcal{P}_d} \mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}, \sigma) = \arg\min_{\bar{\mathbf{M}} \in \mathcal{P}_d} \sum_{n=1}^{N} d^2(\mathbf{M}_n, \bar{\mathbf{M}}), \tag{1.45}$$

where $d(.)$ is the geodesic distance defined in (1.18). The solution of this minimization problem is provided by a Riemannian gradient descent algorithm detailed above in algorithm 1 [Lenglet *et al.* 2006].

Moreover, the maximum likelihood estimate $\hat{\sigma}$ of the dispersion parameter σ is the solution of the following equation:

$$\hat{\sigma} = \arg\max_{\sigma \in \mathbb{R}^+} \mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}, \sigma). \tag{1.46}$$

This leads to find the solution of the non-linear equation:

$$\frac{1}{N}\sum_{n=1}^{N} d^2(\mathbf{M}_n, \bar{\mathbf{M}}) = \sigma^3 \frac{d}{d\sigma}(\log(Z(\sigma)). \tag{1.47}$$

Indeed, the right-hand side of equation (1.47) depends only on the unknown σ, while its left-hand side has a fixed value. As detailed in the work provided by Said *et. al* in [Said *et al.* 2015b], there exists a function $\Phi : \mathbb{R}^+ \to \mathbb{R}^+$ such that:

$$\hat{\sigma} = \Phi\left(N^{-1}\sum_{n=1}^{N} d^2(\mathbf{M}_n, \bar{\mathbf{M}})\right). \tag{1.48}$$

According to (1.48), it means that Φ is the inverse function of $\sigma^3 \frac{d}{d\sigma}(\log(Z(\sigma))$. For more details, the interested reader is referred to [Said *et al.* 2015b]. In practice, it is solved by tabulating the values of the normalization factor $Z(\sigma)$ introduced in (1.38).

1.4.2 Riemannian affine-invariant Gaussian mixture model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of multiple Gaussian component densities. GMMs parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm [Dempster *et al.* 1977]. Here, the definition of mixture models is extended to Riemannian Gaussian distributions.

1.4.2.1 Model definition

Starting from (1.37), the probability density function for a mixture of K Riemannian Gaussian distributions is given by:

$$p(\mathbf{M}|\omega, \bar{\mathbf{M}}, \sigma) = \sum_{k=1}^{K} \omega_k \; p(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k), \tag{1.49}$$

where $p(\mathbf{M}|\bar{\mathbf{M}}_k, \sigma_k)$ is the probability density function of a Riemannian Gaussian distribution given by (1.37). In (1.49), $\omega_k \in [0, 1]$, $\bar{\mathbf{M}}_k \in \mathcal{P}_d$ and $\sigma_k \in \mathbb{R}^+$ are respectively the weight, mean and dispersion for the k^{th} component of the GMM model.

1.4.2.2 Parameter estimation

For each component $k = 1, \ldots, K$, the parameters $\hat{\theta} = \{(\omega_k, \bar{\mathbf{M}}_k, \sigma_k)_{1 \leq k \leq K}\}$ can be estimated using the expectation maximization algorithm (EM) extended to the Riemannian geometry of the space $\mathcal{P}_d$ as proposed by Said et al. in [Said *et al.* 2015a] where they proposed an extension of the EM algorithm to the Riemannian case.

In fact, the EM algorithm consists in two main steps.

The expectation step permits assigning a responsibility score γ_k to each data $\mathbf{M}_n$ for each component k. This quantity indicates how much the data $\mathbf{M}_n$ belongs to the k^{th} Riemannian Gaussian distribution. It is defined by:

$$\gamma_k(\mathbf{M}_n) = \frac{\omega_k p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \sigma_k)}{\sum_{j=1}^{K} \omega_j p(\mathbf{M}_n|\bar{\mathbf{M}}_j, \sigma_j)}. \tag{1.50}$$

The maximization step permits the update of the estimate values considering the scores obtained in the previous step until some convergence threshold is reached.

For each iteration, the parameters are given by:

- **The mixture weight:**

$$\hat{\omega}_k = \frac{\gamma_k(\hat{\theta})}{\sum_{j=1}^{K} \gamma_j(\hat{\theta})}. \tag{1.51}$$

- **The center of mass $\hat{\bar{\mathbf{M}}}_k$:**

$$\hat{\bar{\mathbf{M}}}_k = \arg\min_{\bar{\mathbf{M}}} \sum_{n=1}^{N} \frac{\gamma_k(\mathbf{M}_n)}{N_k} d^2(\mathbf{M}_n, \bar{\mathbf{M}}), \tag{1.52}$$

  where $d(.)$ stands for the geodesic distance defined in equation (1.18).

- **The dispersion $\hat{\sigma}_k$:**

$$\hat{\sigma}_k = \Phi\left(N_k^{-1} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) d^2(\mathbf{M}_n, \bar{\mathbf{M}}_k) \right), \tag{1.53}$$

  where N_k can be seen as the effective number of points assigned to component k and is defined by:

$$N_k = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n). \tag{1.54}$$

Φ is the inverse function of $\sigma \to \sigma^3 \times \frac{d}{d\sigma} Z(\sigma)$ introduced previously. The steps of the EM method for the estimation of the parameters of a Riemannian Gaussian mixture model are summarized in Algorithm 2.

Algorithm 2 EM algorithm for a Riemannian GMM model

Input: $\mathbf{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ a set of covariance matrices where $\mathbf{M}_n \in \mathcal{P}_d$, K number of components of the GMM, N_{iter} maximum number of iterations.

Initialize: $\gamma_k(\mathbf{M})$ via the EM algorithm for the GMM defined in (1.57) at the tangent plane of the identity matrix.

1: $it \leftarrow 1$
2: **while** $(it \leq N_{iter})$ **do**
3: Update $\bar{\mathbf{M}}_k$ by solving (1.52) with Karcher mean Algorithm 1.
4: Update σ_k with (1.53).
5: Update ω_k with (1.51).
6: Update the posterior probability $\gamma_k(\mathbf{M}_n)$ with (1.50).
7: $it \leftarrow it + 1$
8: **end while**

Output: $\omega_k \in [0, 1]$, $\bar{\mathbf{M}}_k \in \mathcal{P}_d$ and $\sigma_k \in \mathbb{R}^+$.

The statistical models defined using the affine-invariant metric space are well suited to the Riemannian geometry and provide precise characterisation of SPD matrices. Nevertheless, they involve complex recursive algorithms of high computational time in particular when estimating the parameters either for the Riemannian Gaussian distribution or its corresponding mixture

model. For example, the calculation of the center of mass requires an iterative procedure (1.25) as well as the computation of the normalization factor $Z(\sigma)$ which needs to be tabulated. Moreover, an isotropic model is used for the Riemannian Gaussian model, which is characterized by a scalar dispersion parameter σ. Considering an isotropic model may be too restrictive when it comes to practical applications. As a consequence, many authors oriented their researches to a less complex framework while preserving accurate practical results. This is the case for the log-Euclidean framework detailed in the next subsection.

1.4.3 Log-Euclidean Gaussian distribution

Once SPD matrices are mapped on the Log-Euclidean space via the operations explained in Section 1.3.3, a multivariate Gaussian distribution can be defined on the LE vector space.

1.4.3.1 Model definition

The probability density function of the Log-Euclidean Gaussian distribution depends on two parameters, the mean $\text{Vec}(\log_{\mathbf{M}_{ref}}(\bar{\mathbf{M}})) \in \mathbb{R}^{d(d+1)/2}$, computed on the tangent space $\mathcal{T}_{\mathbf{M}_{ref}}$ and the sample covariance matrix $\Sigma \in \mathcal{P}_{d(d+1)/2}$ which ensures model anisotropy. It is given by:

$$p(\mathbf{M}_n|\bar{\mathbf{M}},\Sigma) = \frac{\exp\left\{-\frac{1}{2}\left(\text{Vec}\left(\text{Log}_{\mathbf{M}_{ref}}(\mathbf{M}_n)\right) - \text{Vec}\left(\text{Log}_{\mathbf{M}_{ref}}(\bar{\mathbf{M}})\right)\right)^T \Sigma^{-1}\left(\text{Vec}\left(\text{Log}_{\mathbf{M}_{ref}}(\mathbf{M}_n)\right) - \text{Vec}\left(\text{Log}_{\mathbf{M}_{ref}}(\bar{\mathbf{M}})\right)\right)\right\}}{(2\pi)^{\frac{d(d+1)}{4}}|\Sigma|^{1/2}}. \tag{1.55}$$

As discussed previously, different choices can be made for the reference point $\mathbf{M}_{ref}$ such as the identity matrix $\mathbf{I}_d$, the center of mass or the median. The use of the identity matrix is certainly the simplest and the most usual way to map covariance matrices on the tangent space. In that case, the log map operator vanishes to:

$$\text{Log}_{\mathbf{I}_d}(\mathbf{M}) = \text{logm}(\mathbf{M}) \tag{1.56}$$

By considering that the reference point is the identity matrix $\mathbf{I}_d$, (1.55) vanishes to:

$$\begin{aligned} p(\mathbf{M}|\bar{\mathbf{M}},\Sigma) &= p(\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}}|\mu,\Sigma) \\ &= \frac{\exp\{-\frac{1}{2}(\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} - \mu)^T\Sigma^{-1}(\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} - \mu)\}}{(2\pi)^{\frac{d(d+1)}{4}}|\Sigma|^{1/2}}, \end{aligned} \tag{1.57}$$

where

$$\mu = \text{Vec}\left(\log_{\mathbf{I}_d}(\bar{\mathbf{M}})\right) = \bar{\mathbf{m}}^{\mathcal{T}_{Id}} \in \mathbb{R}^{\frac{d(d+1)}{2}}, \tag{1.58}$$

is the log-Euclidean mean vector.

1.4.3.2 Parameter estimation

Identically to the Riemannian Gaussian distribution defined in the previous paragraphs, the parameters of the multivariate LE Gaussian distribution are estimated by the maximum likelihood estimation method. Moreover, since covariance matrices are projected into the tangent plane and represented by their corresponding vectors, all the algorithms developed in an

Euclidean space can be employed, parameters μ and Σ are thus respectively estimated by the sample mean and sample covariance matrix by:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} \tag{1.59}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} - \hat{\mu})(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} - \hat{\mu})^T \tag{1.60}$$

Since the parameters have close-form expressions, it enable fast and effective computations.

1.4.4 Log-Euclidean Gaussian mixture model

1.4.4.1 Model definition

Equivalently to the case of the Riemannian Gaussian mixture model previously presented, a mixture model of multivariate Gaussian distributions is proposed on the LE metric space as well, allowing the characterizing of variability in data sets mapped on the LE space. The probability density function is given by:

$$p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \omega, \mu, \Sigma) = \sum_{k=1}^{K} \omega_k p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \mu_k, \Sigma_k). \tag{1.61}$$

1.4.4.2 Parameter estimation

The mixture model's parameters are: the set of mixture weights ω_k, the mean vectors μ_k and the covariance matrices Σ_k. Given the same data set $\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} = \{\mathbf{m}_1^{\mathcal{T}_{\mathbf{I}_d}}, ..., \mathbf{m}_N^{\mathcal{T}_{\mathbf{I}_d}}\}$ of N independent and identically distributed (i.i.d) drawn from a multivariate Gaussian mixture model, the estimated parameters through the EM algorithm are:

- **The mixture weight $\hat{\omega}_k$:**

$$\hat{\omega}_k = \frac{N_k}{N} \tag{1.62}$$

- **The mean vectors $\hat{\mu}_k$:**

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}})\ \mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} \tag{1.63}$$

- **The dispersion matrices $\hat{\Sigma}_k$:**

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) \left((\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} - \mu_k)(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} - \mu_k)^T \right), \tag{1.64}$$

where

$$\gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) = \frac{\omega_k p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \omega_j p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \mu_j, \Sigma_j)}, \tag{1.65}$$

and

$$N_k = \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}).$$

The steps of the EM algorithm for the estimation of the parameters of a multivariate Gaussian mixture model are summarized in Algorithm 3.

Algorithm 3 EM algorithm for a Log-Euclidean GMM model

Input: $\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} = \{\mathbf{m}_1^{\mathcal{T}_{\mathbf{I}_d}}, ..., \mathbf{m}_N^{\mathcal{T}_{\mathbf{I}_d}}\}$ the vector-form presentation of the set of covariance matrices $\mathbf{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$, K number of components of the GMM, N_{iter} maximum number of iterations.
Initialize: $\omega_k = \frac{1}{K}$, μ_k with a random data sample from $\mathbf{m}_1^{\mathcal{T}_{\mathbf{I}_d}}, ..., \mathbf{m}_N^{\mathcal{T}_{\mathbf{I}_d}}$ and $\gamma_k(\mathbf{M})$ according to (1.65) with $N_k = N$ and $\gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) = \frac{1}{K}$.

1: $it \leftarrow 1$
2: **while** $(it \leq N_{iter})$ **do**
3: Update μ_k with (1.63).
4: Update Σ_k with (1.64).
5: Update ω_k with (1.62).
6: Update the posterior probability $\gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}})$ with (1.65).
7: $it \leftarrow it + 1$
8: **end while**

Output: $\omega_k \in [0, 1]$, $\mu_k \in \mathcal{P}_d$ and $\Sigma_k \in \mathcal{P}_{d(d+1)/2}$.

Note that in this log-Euclidean case, the parameter estimates are given in closed form, while for the statistical models defined on the AI metric space, recursive estimation algorithms are needed to estimate these parameters. In the following, a more detailed comparison between the two metrics is provided.

1.4.5 Comparison between the AI and LE Gaussian models

1.4.5.1 Overview of AI and LE Gaussian models

Table 1.3 draws an overview of both Riemannian and log-Euclidean statistical models on the space of covariance matrices.

	Log-Euclidean metric[2]	**Affine-invariant metric**
Gaussian model	$p\left(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\middle\vert \mu, \Sigma\right) = \frac{\exp\{-\frac{1}{2}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu)^T\Sigma^{-1}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu)\}}{(2\pi)^{\frac{d(d+1)}{4}}\vert\Sigma\vert^{1/2}}$	$p(\mathbf{M}_n\vert\bar{\mathbf{M}}, \sigma) = \frac{1}{Z(\sigma)}\exp\left[-\frac{d^2(\mathbf{M}_n,\bar{\mathbf{M}})}{2\sigma^2}\right]$
Gaussian mixture model	$p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\vert\theta) = \sum_{k=1}^K \omega_k p\left(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\middle\vert\mu_k, \Sigma_k\right)$ where $p\left(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\middle\vert\mu_k, \sigma_k\right) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu_k)^T\Sigma_k^{-1}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu_k)\right\}}{(2\pi)^{\frac{d(d+1)}{4}}\vert\Sigma_k\vert^{1/2}}$ with $\mu_k \in \mathbb{R}^{\frac{d(d+1)}{2}}$, $\sigma_k^2 = \text{diag}(\Sigma_k) \in \mathbb{R}^{\frac{d(d+1)}{2}}$ and $\omega_k \in [0, 1]$.	$p(\mathbf{M}_n\vert\omega, \bar{\mathbf{M}}, \sigma) = \sum_{k=1}^K \omega_k\, p(\mathbf{M}_n\vert\bar{\mathbf{M}}_k, \sigma_k)$ where $p(\mathbf{M}_n\vert\bar{\mathbf{M}}_k, \sigma_k) = \frac{1}{Z(\sigma_k)}\exp\left[-\frac{d^2(\mathbf{M}_n,\bar{\mathbf{M}}_k)}{2\sigma_k^2}\right]$ $\bar{\mathbf{M}}_k \in \mathcal{P}_d$, $\sigma_k > 0$ and $\omega_k \in [0, 1]$.
Characteristics	Computations on the tangent space Anisotropic model ($\Sigma \in \mathcal{P}_{d(d+1)/2}$) Closed-form parameter expression Simple normalization factor	Computations on manifold Isotropic model ($\sigma \in \mathbb{R}$) Recursive parameter estimation algorithm Complex normalization factor $Z(\sigma)$

Table 1.3: Comparison between the two considered GMM models: one defined using the LE metric and second using the AI Riemannian metric.

[2]Note that the image classification algorithms introduced in the next chapter imposes Gaussian model covariance matrices, Σ_k, to be diagonal. This is suitable for most computer vision applications, for example Fisher vector encoding assumes that descriptors are generated by a GMM model with diagonal covariance matrices.

Both models share some similarities (Gaussian distribution) but differs in some points. An anisotropic model is considered for the LE metric where the dispersion for the Gaussian mixture model is a diagonal matrix Σ_k while an isotropic model is used for the affine-invariant Riemannian metric by considering a scalar dispersion σ_k. Moreover, for the LE metric, the computations are made on the tangent space at a defined reference point. As such, calculations are simplified while for the AI metric, the computations are performed on the manifold which involves recursive algorithms for parameter estimation and a computation of a complex normalization factor.

1.4.5.2 Application to texture image classification

The LE metric has been tested against the affine-invariant metric in different applications, in particular Ilea *et al.* provides in [Ilea *et al.* 2018b] a fair comparison between these two approaches in terms of overall accuracy on an image classification application. Table 1.4 shows the classification results obtained on four texture databases (VisTex [Picard *et al.* 2010], Brodatz [Brodatz 1966], Outex [Ojala *et al.* 2002], USPtex [Backes *et al.* 2012]). The performances are displayed for the Fisher vector encoding computed by using the derivative with respect to the centroid $\bar{\mathbf{M}}_k$. This FV encoding strategy will be explained in section 2.2.1.3 of chapter 2. For the LE metric, an isotropic model can be built by considering that $\Sigma_k = \sigma_k^2 \mathbf{I}_{d(d+1)/2}$. For the affine-invariant Riemannian metric, we recall that the Riemannian Gaussian distribution is isotropic.

	Anisotropic Model	**Isotropic Model**	
Database	**Log-Euclidean Metric**	**Log-Euclidean Metric**	**Affine-Invariant Riemannian Metric**
VisTex	95.5 ± 0.01	88.7 ± 0.01	91.3 ± 0.01
Brodatz	93.5 ± 0.01	87.1 ± 0.01	92.9 ± 0.01
Outex	87.3 ± 0.01	83.2 ± 0.01	85.4 ± 0.01
USPtex	88.3 ± 0.01	81.5 ± 0.01	87.0 ± 0.01

Table 1.4: Comparison between anisotropic and isotropic models, classification results.

One can notice that for the LE metric, an anisotropic model yields to a significant gain of about 4% to 7% compared to an isotropic model. More interestingly, for an isotropic model, descriptors based on the affine-invariant Riemannian metric yield to better performances than that obtained with the LE metric. A gain of about 2% to 6% is observed. These experiments clearly illustrate that the gain observed for the LE metric comes better from the anisotropicity of the Gaussian mixture model than from the metric definition. Furthermore, considering an anisotropic Gaussian model based on the AI metric leads to very complex calculations. In addition, when considering the Log-Euclidean metric, the GMM modeling is limited to a single tangent plane defined at the identity matrix. This makes a hidden assumption that the covariance matrices are located on a local region of the manifold and may lead to projection distortions when covariance matrices are localized far from it. Since then, Pennec has introduced in [Pennec 2004] an anisotropic Gaussian model on the Riemannian manifold to preserve the geometrical data properties. Nevertheless, it involves complex calculations and requires an approximation of the normalization factor for a covariance matrix of small variance. As a consequence, this proposition remains difficult to apply in practice where FV computation induce complex calculations.

Since then, an intermediate alternative is proposed in the following which allows preserving as far as possible the specific geometry of SPD matrices as well as reducing the computational cost. It permits modeling the Gaussian distributions by considering multiple tangent planes at different reference points that are suited properly to the distribution of covariance matrices.

1.4.6 Extension to multiple tangent planes

As shown before, the projection on a unique tangent plane allows to ease the development of processing methods which are based on covariance matrices features, since standard Euclidean tools can be considered. But the choice of the reference point (and in particular the identity matrix) might be problematic. The observed covariance matrices can be far away from this reference point. In order to capture more accurately the structure of the observed covariance matrices, we introduce a GMM model defined at different reference points, one per component of the GMM [Simo-Serra *et al.* 2017, Calinon & Jaquier 2019].

1.4.6.1 Model definition

In order to have a reference point close to the covariance matrices which belong to the cluster, we propose to define it equal to the centroid $\bar{\mathbf{M}}_k$. It yields that (1.55) vanishes to:

$$\begin{aligned} p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k) &= \frac{\exp\{-\frac{1}{2}(\mathbf{m}^{\mathcal{T}_{\bar{\mathbf{M}}_k}})^T \Sigma_k^{-1} (\mathbf{m}^{\mathcal{T}_{\bar{\mathbf{M}}_k}})\}}{(2\pi)^{\frac{d(d+1)}{4}} |\Sigma_k|^{1/2}} \\ &= \frac{\exp\left\{-\frac{1}{2}\left(\mathrm{Vec}(\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}))\right)^T \Sigma_k^{-1} \left(\mathrm{Vec}(\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}))\right)\right\}}{(2\pi)^{\frac{d(d+1)}{4}} |\Sigma_k|^{1/2}}. \end{aligned} \tag{1.66}$$

As observed, the k^{th} component of this GMM model corresponds to a zero-mean multivariate Gaussian distribution for the vectors computed at the reference point $\bar{\mathbf{M}}_k$. Interestingly, the mean is zero since it has been transferred to the reference point $\bar{\mathbf{M}}_k$.

1.4.6.2 Parameter estimation

Let $\mathbf{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ be a set of N i.i.d covariance matrices issued from the GMM model where its component is defined in (1.66). We propose to define an EM algorithm to estimate the GMM parameters. First the Log-likelihood function is defined as follows:

$$\mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}, \Sigma) = \log \prod_{n=1}^{N} \sum_{k=1}^{K} \omega_k p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \Sigma_k) \tag{1.67}$$

$$= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \omega_k p(\mathbf{M}_n|\bar{\mathbf{M}}_k, \Sigma_k) \tag{1.68}$$

- **The center of mass $\bar{\mathbf{M}}_k$:**

The estimation of $\bar{\mathbf{M}}_k$ is found by deriving the log-likelihood (1.68) with respect to $\bar{\mathbf{M}}_k$ as follows:

$$\frac{\partial}{\partial \bar{\mathbf{M}}_k} \mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k) = \sum_{n=1}^{N} \frac{\omega_k \frac{\partial}{\partial \bar{\mathbf{M}}_k} p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k)}{\sum_{j=1}^{K} \omega_j p(\mathbf{M}|\bar{\mathbf{M}}_j, \Sigma_j)}. \tag{1.69}$$

In order to simplify calculations of the maximum likelihood (ML) derivation with respect to $\bar{\mathbf{M}}_k$, we consider two functions f and g to compute the derivative of $p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k)$ with respect to $\bar{\mathbf{M}}_k$. First, the function $f(\mathbf{x})$ is defined by:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{(2\pi)^{p/2}\prod_{j=1}^{p}\sigma_k(j)}\exp\left(-\frac{1}{2}\mathbf{x}^T\Sigma_k^{-1}\mathbf{x}\right), \\ &= \frac{1}{(2\pi)^{p/2}\prod_{j=1}^{p}\sigma_k(j)}\exp\left(-\frac{1}{2}\sum_{j=1}^{p}\frac{\mathbf{x}^2(j)}{\sigma_k^2(j)}\right). \end{aligned} \tag{1.70}$$

where $\mathbf{x} = \mathbf{m}^{\mathcal{T}_{\bar{\mathbf{M}}_k}}$, $p = \frac{d(d+1)}{2}$ and Σ_k is a diagonal covariance matrix. Secondly, the function $g(\mathbf{x})$ is defined by:

$$g(\mathbf{x}) = \mathrm{Vec}(\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{x})). \tag{1.71}$$

i - Computation of the derivative f'

According to the matrix derivative formulated in the matrix cookbook [Petersen & Pedersen 2008], the derivative of $\mathbf{x}^T\Sigma\mathbf{x}$ with respect to $\mathbf{x}$ is given by:

$$\begin{aligned} \frac{\partial \mathbf{x}^T\Sigma^{-1}\mathbf{x}}{\partial \mathbf{x}} &= \left(\Sigma^{-1} + (\Sigma^{-1})^T\right)\mathbf{x} \\ &= 2\ \Sigma^{-1}\ \mathbf{x}. \end{aligned} \tag{1.72}$$

It yields that the derivative of $f(\mathbf{x})$ is:

$$f'(\mathbf{x}) = -\Sigma\ \mathbf{x}\ f(\mathbf{x}), \tag{1.73}$$

ii- Computation of the derivative g'

In order to derive $g(\mathbf{x})$, a first order Taylor series expansion of the log-map operator is employed as follows:

$$\begin{aligned} \mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}) &= \bar{\mathbf{M}}_k \mathrm{logm}(\bar{\mathbf{M}}_k^{-1}\mathbf{M}) \\ &\approx \bar{\mathbf{M}}_k(\bar{\mathbf{M}}_k^{-1}\mathbf{M} - \mathbf{I}_d) \\ &= \mathbf{M} - \bar{\mathbf{M}}_k \end{aligned} \tag{1.74}$$

The approximation with Taylor series expansion holds if $\mathbf{M}$ is close to the reference point $\bar{\mathbf{M}}_k$. This is valid since $\bar{\mathbf{M}}_k$ is the centroid of the elements which belong to cluster k. Since $\bar{\mathbf{M}}_k$ is symmetric and by using (1.74), the derivative of the log-map operator for the off-diagonal terms are given by:

$$\frac{\partial \mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M})}{\partial \bar{\mathbf{M}}_k(i,j)} \approx -(\mathbf{J}^{ij} + \mathbf{J}^{ji}), \tag{1.75}$$

where $\mathbf{J}^{ij}$ is the $d \times d$ single-entry matrix where the (i,j) element is one and the rest of elements are zero. Similarly, the derivative of the log-map operator with respect to its diagonal terms is

expressed as:

$$\frac{\partial \text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M})}{\partial \bar{\mathbf{M}}_k(i,i)} \approx -\mathbf{J}^{ii}. \tag{1.76}$$

Finally, the computation of the derivative (1.69) using the composite function rule is rewritten, where:

$$(f \circ g)' = g'(f' \circ g). \tag{1.77}$$

After some straightforward computations and by combining (1.73), (1.75) and (1.76), the maximum likelihood estimator of $\bar{\mathbf{M}}_k$ is found as the solution of:

$$\frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n) = 0, \tag{1.78}$$

where $\gamma_k(\mathbf{M})$ is the posterior probability that a covariance matrix $\mathbf{M}$ belongs to cluster k:

$$\gamma_k(\mathbf{M}) = \frac{\omega_k \; p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k)}{\sum\limits_{j=1}^{K} \omega_j \; p(\mathbf{M}|\bar{\mathbf{M}}_j, \Sigma_j)}. \tag{1.79}$$

But since $\frac{\partial d^2(\bar{\mathbf{M}}_k, \mathbf{M})}{\partial \bar{\mathbf{M}}_k} = -2\text{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M})$, (1.78) can equivalently be rewritten as:

$$\hat{\bar{\mathbf{M}}}_k = \arg\min_{\bar{\mathbf{M}}} \sum_{n=1}^{N} \frac{\gamma_k(\mathbf{M}_n)}{N_k} d^2(\bar{\mathbf{M}}, \mathbf{M}_n), \tag{1.80}$$

where

$$N_k = \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n), \tag{1.81}$$

and $d(.)$ is the Rao's geodesic distance induced by the affine-invariant Riemannian metric defined in (1.18). In practice, (1.80) can be solved by a Karcher mean algorithm [Karcher 1977] as seen in Algorithm 1.

- **The dispersion σ_k^2 :**

Similarly, the maximum likelihood estimator of the i^{th} component of σ_k^2 is defined as:

$$\frac{\partial}{\partial \sigma_k(i)} \mathbb{L}(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k) = \sum_{n=1}^{N} \frac{\omega_k \frac{\partial}{\partial \sigma_k(j)} p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k)}{\sum_{j=1}^{K} \omega_j p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k}), \tag{1.82}$$

where,

$$\frac{\partial}{\partial \sigma_k(i)} p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k) = -\frac{1}{\sigma_k(i)} p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k) + \frac{\left[\mathbf{m}_n^{\mathcal{T}_{\bar{\mathbf{M}}_k}}(i)\right]^2}{\sigma_k^3(i)} p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k). \tag{1.83}$$

Then (1.82) can be rewritten as:

$$\frac{\partial}{\partial \sigma_k(i)} \mathbb{L}(\mathbf{M}_n|\bar{\mathbf{M}}_k, \Sigma_k) = \sum_{n=1}^{N} \left[\frac{\omega_k p(\mathbf{M}_n|\Sigma_k, \bar{\mathbf{M}}_k)}{\sum_{j=1}^{K} \omega_j p(\mathbf{M}_n|\Sigma_j, \bar{\mathbf{M}}_j)} \left(-\frac{1}{\sigma_k(i)} + \frac{\left[\mathbf{m}_n^{\mathcal{T}_{\bar{\mathbf{M}}_k}}(i)\right]^2}{\sigma_k^3(i)} \right) \right]. \tag{1.84}$$

Using the posterior probability $\gamma_k(\mathbf{M})$ defined in (1.79), the maximum likelihood estimator of the i^{th} component of $\hat{\sigma}_k^2$ is the weighted sample variance defined as:

$$\hat{\sigma}_k^2(i) = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left[\mathbf{m}_n^{\mathcal{T}_{\bar{\mathbf{M}}_k}}(i)\right]^2. \tag{1.85}$$

- **The weight ω_k:**

The maximum likelihood estimator of ω_k is given by:

$$\hat{\omega}_k = \frac{N_k}{N}. \tag{1.86}$$

1.4.6.3 Numerical instability

In practice, for the model with several tangent planes, very large dispersion coefficients σ_k have been estimated for some experiments on real data. In fact, since multiple tangent planes at $\bar{\mathbf{M}}_k$ are considered, dispersion coefficients σ_k are computed for observations projected onto each tangent plane. As a result, large values are obtained and may yield to some numerical instabilities for the proposed EM algorithm. To circumvent this drawback and ensure a fair comparison, prior to estimate σ_k, we propose, according to the definition of parallel transport introduced in section 1.3.1.4, to transport the observed data on a same tangent space around the identity matrix by applying the following operation:

$$\mathbf{Z}_{(n,k)} = \bar{\mathbf{M}}_k^{-\frac{1}{2}} \ \mathbf{M}_n \ (\bar{\mathbf{M}}_k^{-\frac{1}{2}})^T. \tag{1.87}$$

To illustrate the fact that this transport is done for each component k of the GMM model, the centered covariance matrix $\mathbf{M}_n$ is denoted $\mathbf{Z}_{(n,k)}$.

It is now possible to estimate the variance $\eta_k^2(j)$ for the transported set according to:

$$\hat{\eta}_k^2(j) = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left[\mathbf{z}_{(n,k)}^{\mathcal{T}_{\mathbf{I}_d}}(j)\right]^2, \tag{1.88}$$

where $\mathbf{z}_{(n,k)}^{\mathcal{T}_{\mathbf{I}_d}}$ is the LE vector representation of $\mathbf{Z}_{(n,k)}$ computed at the identity matrix $\mathbf{I_d}$. In order to explain why transporting the set of covariance matrices $\mathbf{M}_n$ around the identity matrix allows to reduce the variance values, let's compute:

$$\mathrm{Log}_{\bar{\mathbf{M}}_k}(\mathbf{M}_n) = \bar{\mathbf{M}}_k^{\frac{1}{2}} \ \mathrm{Log}_{\mathbf{I}_d}(\mathbf{Z}_{(n,k)}) \ (\bar{\mathbf{M}}_k^{\frac{1}{2}})^T. \tag{1.89}$$

As an example, if we consider $\bar{\mathbf{M}}_\mathbf{k} = K \ \mathbf{I}_d$ and by combining (1.88) and (1.89), we obtain:

$$\begin{aligned} \mathrm{Log}_{\bar{\mathbf{M}}_\mathbf{k}}(\mathbf{M}_n) &= (K \ \mathbf{I}_d)^{\frac{1}{2}} \ \mathrm{Log}_{\mathbf{I}_d}(\mathbf{Z}_{(n,k)}) \ ((K \ \mathbf{I}_d)^{\frac{1}{2}})^T \\ &= K \ \mathrm{Log}_{\mathbf{I}_d}(\mathbf{Z}_{(n,k)}). \end{aligned} \tag{1.90}$$

Then, (1.88) induced to:

$$\hat{\sigma}_k^2(j) = K\ \hat{\eta}_k^2(j). \tag{1.91}$$

It yields that the dispersion σ_k increases as the set of covariance matrices $\mathbf{M}_n$ are located far from the identity matrix $\mathbf{I}_d$.

The proposed approach of parallel transport of data around the identity matrix allows hence to reduce the dispersion and avoid the numerical instabilities. Now, to derive the EM algorithm, the posterior probability $\gamma_k(\mathbf{M}_n)$ should be computed with the shifted covariance matrix $\mathbf{Z}_{(n,k)}$. It yields:

$$\gamma_k(\mathbf{M}_n) = \frac{\omega_k\ p(\mathbf{Z}_{(n,k)}|\mathbf{I}_d, \mathbf{H}_k)}{\sum_{j=1}^{K} \omega_j\ p(\mathbf{Z}_{(n,j)}|\mathbf{I}_d, \mathbf{H}_j)}, \tag{1.92}$$

where $\mathbf{H}_k$ is a diagonal matrix containing the variance vector elements η_k^2 on its diagonal. To summarize, the EM algorithm for the GMM with K reference points $\bar{\mathbf{M}}_k$ is given in Algorithm 4.

Algorithm 4 EM algorithm for a GMM model with different reference points

Input: $\mathbf{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ a set of covariance matrices where $\mathbf{M}_n \in \mathcal{P}_d$, K number of components of the GMM, N_{iter} maximum number of iterations.

Initialize: $\gamma_k(\mathbf{M}_n)$ via the EM algorithm for the GMM defined in (1.57) at the tangent plane of the identity matrix.

1: $it \leftarrow 1$
2: **while** $(it \leq N_{iter})$ **do**
3: Update $\bar{\mathbf{M}}_k$ by solving (1.80) with Karcher/Fréchet mean algorithm.
4: Compute $\mathbf{Z}_{(n,k)}$ with (1.87) to transport $\mathbf{M}_n$.
5: Update η_k with (1.88).
6: Update ω_k with (1.86).
7: Update the posterior probability $\gamma_k(\mathbf{M}_n)$ with (1.92).
8: $it \leftarrow it + 1$
9: **end while**

Output: $\omega_k \in [0, 1]$, $\bar{\mathbf{M}}_k \in \mathcal{P}_d$ and $\eta_k \in \mathbb{R}^{\frac{d(d+1)}{2}}$.

1.4.6.4 Comparison between the two GMM models

i - Models comparison

Table 1.5 draws an overview of the two considered GMM models. The first one consists in a classical GMM model where covariance matrices are projected on the tangent plane at the identity matrix while the second one considers projections with multiple tangent planes. Each mixture component has its own tangent space. Even if these two models have many similarities (GMM models, projection on a tangent plane), they differ in some aspects:

- For the second model, there is no offset parameter μ_k since the mean has been transferred to the reference point $\bar{\mathbf{M}}_k$.
- The maximum likelihood estimator of the centroid for the GMM model defined at a unique reference point is the weighted log-Euclidean mean vector while for the second model, it is the centroid on the manifold, *i.e.* the Karcher/Fréchet mean [Karcher 1977].

	Unique reference point at identity	K reference points
Gaussian mixture model	$p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\|\lambda) = \sum_{k=1}^{K} \omega_k p\left(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\|\mu_k, \sigma_k\right)$ where $p\left(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}\|\mu_k, \sigma_k\right) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu_k)^T\Sigma_k^{-1}(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}-\mu_k)\right\}}{(2\pi)^{\frac{d(d+1)}{4}}\|\Sigma_k\|^{1/2}}$ with $\mu_k \in \mathbb{R}^{\frac{d(d+1)}{2}}$, $\sigma_k^2 = diag(\Sigma_k) \in \mathbb{R}^{\frac{d(d+1)}{2}}$ and $\omega_k \in [0,1]$.	$p(\mathbf{M}_n\|\lambda) = \sum_{k=1}^{K} \omega_k p\left(\mathbf{Z}_{(n,k)}\|\mathbf{I}_d, \eta_k\right)$ where $p\left(\mathbf{Z}_{(n,k)}\|\mathbf{I}_d, \eta_k\right) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{z}_{(n,k)}^{\mathcal{T}_{\mathbf{I}_d}})^T\mathbf{H}_k^{-1}(\mathbf{z}_{(n,k)}^{\mathcal{T}_{\mathbf{I}_d}})\right\}}{(2\pi)^{\frac{d(d+1)}{4}}\|\mathbf{H}_k\|^{1/2}}$ with $\mathbf{Z}_{(n,k)} = (\bar{\mathbf{M}}_k^{-\frac{1}{2}})^T\,\mathbf{M}_n\,\bar{\mathbf{M}}_k^{-\frac{1}{2}}$, $\bar{\mathbf{M}}_k \in \mathcal{P}_d$, $\eta_k^2 = diag(\mathbf{H}_k) \in \mathbb{R}^{\frac{d(d+1)}{2}}$ and $\omega_k \in [0,1]$.
Characteristics	Projection onto a unique tangent plane at $\mathbf{I}_d$ Centroïd : Log-Euclidean mean vector	Projection on multiple tangent spaces Centroïd : Karcher/Fréchet mean

Table 1.5: Comparison between the two considered GMM models: one defined at a unique reference point and one with K reference points, one per cluster.

From a practical point of view, the difference between the two methods can be discussed in terms of distortions induced by matrix projection onto the tangent plane. Indeed, in order to illustrate the difference between the two approaches, in particular the influence of the chosen reference point on fitting correctly a Gaussian model, a set of synthetic data $\{\mathbf{x}_i\}_{i=1,\dots,N}$ has been generated according to a multivariate Gaussian distribution $\mathcal{N}(0,\Sigma)$. To obtain covariance matrices $\{\mathbf{X}_i\}_{i=1,\dots,N}$, exponential mapping is performed to project the Gaussian vectors into the Riemannian manifold at a reference point $\bar{\mathbf{M}}$. The block diagram of Figure 1.6 illustrates the conducted experiments (1, 2, 3 and 4) to clarify the influence of the chosen reference point.

ii - Experimental procedure

Two main experiments are carried out. In the left side, blocks surrounded by a red dotted line permit to assess the importance of transporting the data around the identity matrix $\mathbf{I}_d$ before projection in the tangent space. The scatter plot $(\mathbf{x}_i^{\mathcal{T}_{\mathbf{I}_d}})_{i=1\dots N}$ of data generated at $\bar{\mathbf{M}}$ and projected directly at the identity matrix $\mathbf{I}_d$ (1) is compared with the scatter plot $(\mathbf{z_i}^{\mathcal{T}_{\mathbf{I}_d}})_{i=1\dots N}$ of the same data that was first transported from $\bar{\mathbf{M}}$ to $\mathbf{I}_d$ according to the parallel transport (1.87) then projected on the tangent space (2). The scheme in the lower left illustrates roughly the experiment to show the quality of data fitting to the Gaussian model after projection. As observed, the first experiment (1) results on some projection distortions where the initial shape is not exactly preserved (in red). In contrast, when data are transported to the identity (2), it remains fitting a Gaussian model (in blue).

Secondly, on the right side, starting from the generated covariance matrices, a novel reference point $\hat{\bar{\mathbf{M}}}$ is estimated, in this instance the center of mass is computed (1.25). The comparison is made between scatter point of data transported to the identity after applying the transport operation from $\bar{\mathbf{M}}$ (2) and $\hat{\bar{\mathbf{M}}}$ (3). In the lower right scheme, one can observe that the scatter point corresponding to both experiments (blue and orange), preserve its initial Gaussian shape after projection to the tangent plane at $\mathbf{I}_d$.

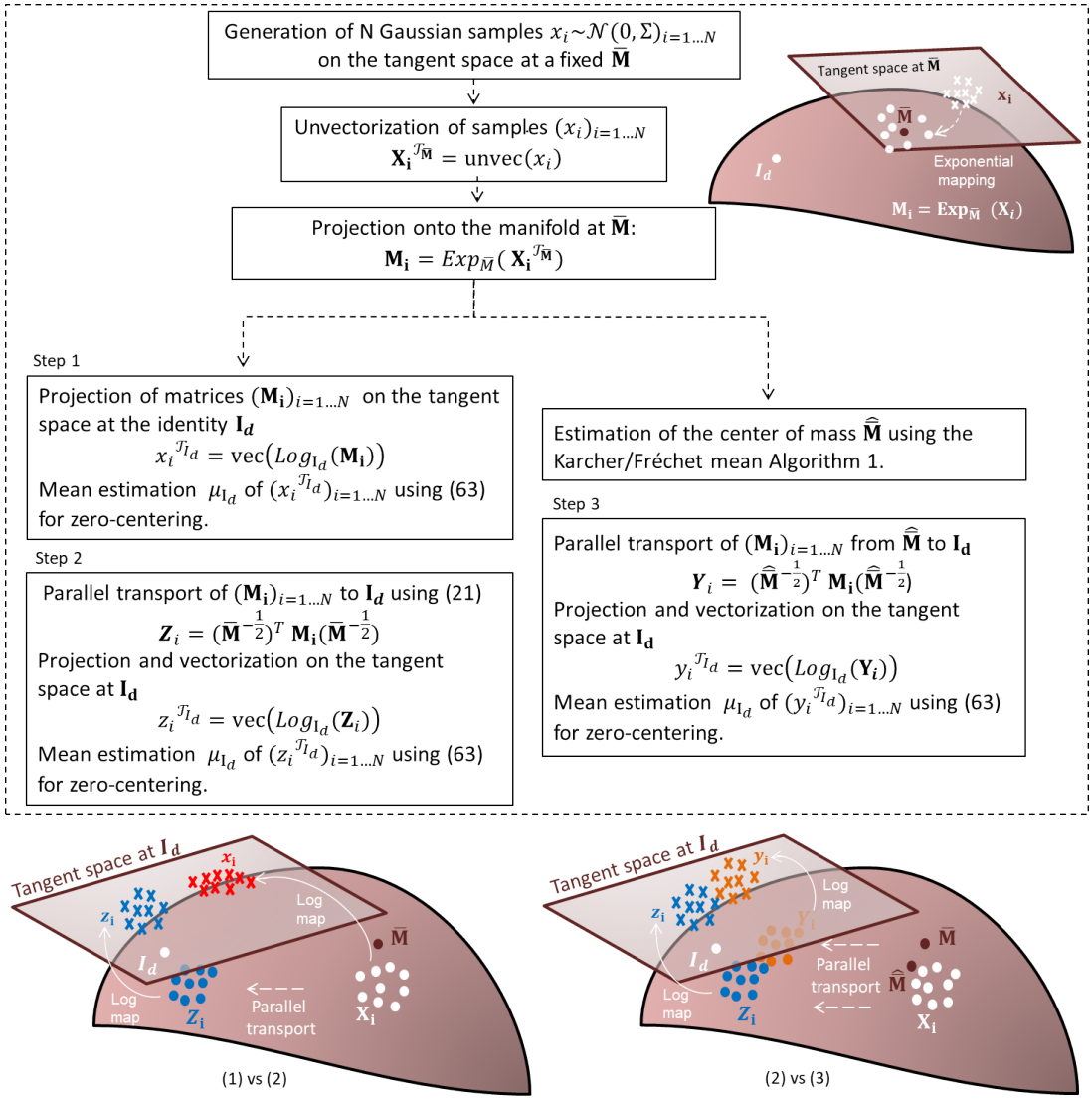


Figure 1.6: Block diagram of the two studied construction to analyze the projection behaviour according to the chosen reference point and highlight the induced distortions.

iii - Distortion measurement

In order to quantify those effects, Figure 1.7 illustrates the distortion between two sets of covariance matrices as a function of the geodesic distance between the identity matrix and the considered reference point $\bar{\mathbf{M}}$. To quantify the distortion, a similarity measure between two set of points is employed. Here, we propose to use the Hausdorff distance. It has been used in [Labsir 2020] for similar purpose and permits measuring the similarity between two sets. Two sets are considered close to each other in the Hausdorff distance if every point of the first set is close to some point of the second set. Let be $\mathbf{X}$ and $\mathbf{Y}$ be two non-empty subsets of a metric space, their Hausdorff distance $d_H(\mathbf{X},\mathbf{Y})$ is defined by:

$$d_H(\mathbf{X},\mathbf{Y}) = \max\left\{\sup_{x\in\mathbf{X}} d(x,\mathbf{Y}), \sup_{y\in\mathbf{Y}} d(y,\mathbf{X})\right\}, \tag{1.93}$$

with $d(x,\mathbf{Y}) = \min_{y\in\mathbf{Y}} d(x,y)$ and $d(.)$ a distance. Since the computation is made on the tangent plane, an Euclidean distance is performed, where $d(x,y) = ||x-y||_2$.

For this experiment, two models are considered for comparison as a function of the geodesic distance between the identity matrix $\mathbf{I}_d$ and $\bar{\mathbf{M}}$:

- **Distor($\mathbf{I_d}$)** : Distance is computed between scatter plot of matrices projected at the identity matrix (step 1), and matrices transported to the identity matrix from $\bar{\mathbf{M}}$ according to the parallel transport in (1.87) (step 2). The mean is removed to center at the same point with the purpose of comparing the scatter plot shapes.

- **Distor($\hat{\bar{\mathbf{M}}}$)** : Distance is computed between scatter plot of matrices projected at the identity matrix after applying the transport operation (1.87) from $\bar{\mathbf{M}}$ (setp 2) and $\hat{\bar{\mathbf{M}}}$ (step 3) to the identity matrix $\mathbf{I}_d$.

The numbers (1), (2) and (3) refers to the blocks of the diagram in Figure 1.6.

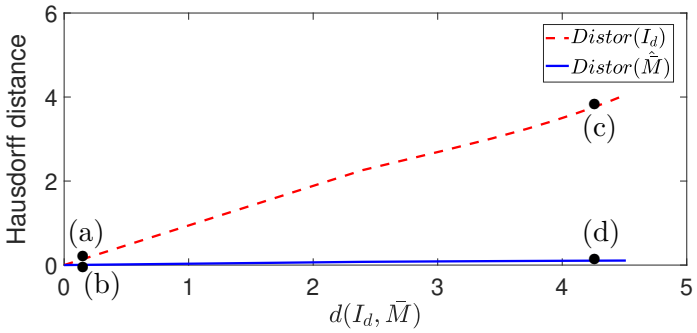


Figure 1.7: Hausdorff distance comparison as a function of geodesic distance between $\mathbf{I_d}$ and $\bar{\mathbf{M}}$.

As demonstrated, the Hausdorff distance in red dash line **Distor($\mathbf{I_d}$)** increases with the increase of the geodesic distance between $\bar{\mathbf{M}}$ and the identity $\mathbf{I}_d$. This dissimilarity reveals the induced distortion between the two sets. In contrast, the distance **Distor($\hat{\bar{\mathbf{M}}}$)** in blue remains unchanged regardless the distance between $\bar{\mathbf{M}}$ and $\mathbf{I}_d$ as a sign of ensured similarity of the two sets.

Moreover, for a complete description, the distinction between two individual cases is assessed. It concerns the manner in which $\bar{\mathbf{M}}$ is distant from the identity matrix $\mathbf{I}_d$.

Case of $\bar{\mathbf{M}} \neq K\mathbf{I}_d$, with K a positive scalar:

To visualize samples behaviour, two instances are experimented according to the two considered models. It corresponds to the selected points (a), (b), (c) and (d) on the graph of Figure 1.7.

- First, the considered $\bar{\mathbf{M}}$ is fixed close to the identity matrix, with $\bar{\mathbf{M}} = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$. Figure 1.8-(a) shows the log-Euclidean vectors at the identity matrix in red and the shifted set from $\bar{\mathbf{M}}$ to $\mathbf{I}_d$ in blue. Where in Figure 1.8-(b), the red color is associated to samples shifted from the estimated centroid $\hat{\bar{\mathbf{M}}}$ to $\mathbf{I}_d$. In addition, the theoretical 3-D ellipse has also been plotted to judge the quality of the fitting at 95% confidence interval. As observed in (a), since the two compared reference points, identity matrix and $\bar{\mathbf{M}}$, are quite close, the distance between the two projections remains small and the Gaussian modeling is ensured.

- Second, the considered $\bar{\mathbf{M}}$ is set far from the identity matrix where $\bar{\mathbf{M}} = \begin{pmatrix} 1 & 0 \\ 0 & 90 \end{pmatrix}$. As observed in Figure 1.8-(c), distortions appear on covariance projected at the identity matrix (red), where in 1.8-(d) the use of an estimated $\hat{\bar{\mathbf{M}}}$ as a reference point (red), allows preventing those distortions and ensuring a well fitting of the Gaussian model. As such, 1.8-(b) and 1.8-(d) have the same behaviour where there is no distortion regardless the distance between $\bar{\mathbf{M}}$ and $\mathbf{I}_d$ as previously illustrated in Figure 1.7.

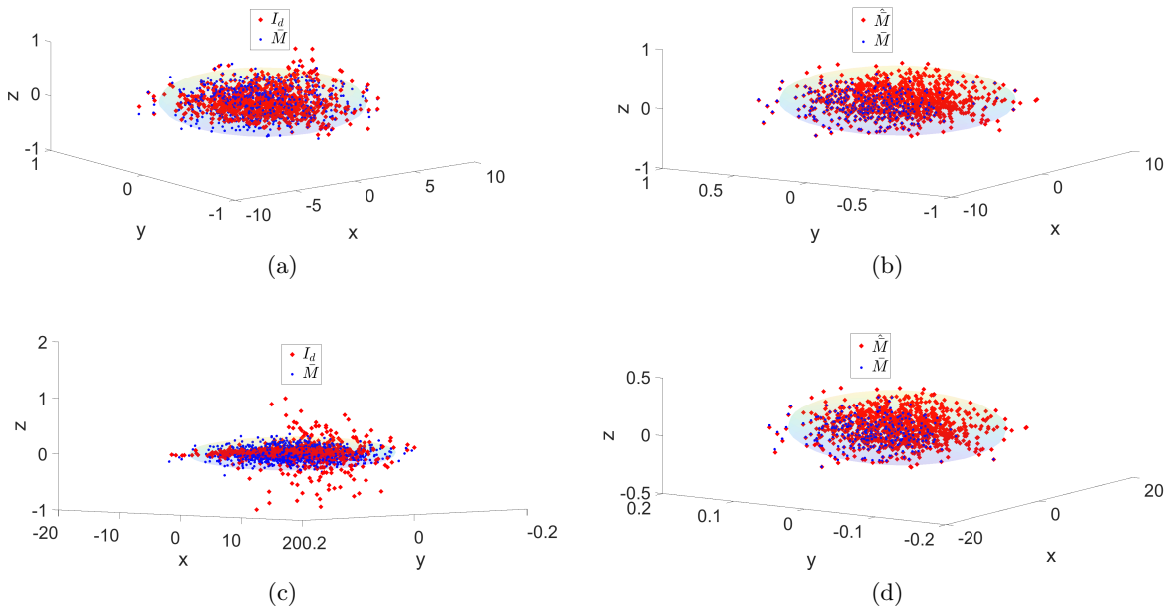


Figure 1.8: Samples behaviour on the tangent space at the identity matrix $\mathbf{I}_d$ and representation of the ellipse at 95 % confidence interval for a set of 3-D normally distributed samples. (a) and (b): Experiments Distor($\mathbf{I}_d$) and Distor($\hat{\bar{\mathbf{M}}}$) with $\bar{\mathbf{M}}$ close to $\mathbf{I}_d$, (c) and (d): Experiments Distor($\mathbf{I}_d$) and Distor($\hat{\bar{\mathbf{M}}}$) with $\bar{\mathbf{M}}$ far from $\mathbf{I}_d$.

Case of $\bar{\mathbf{M}} = K\mathbf{I}_d$, with K a positive scalar:

The geodesic between the two points remains linear. To illustrate that, the geodesic between two points $\mathbf{X} = \mathbf{I}_d$ and $\mathbf{Y} = K\mathbf{I}_d$ defined in (1.16) is rewritten as:

$$\begin{aligned} \gamma(t) &= \mathbf{X}^{\frac{1}{2}} \ (\mathbf{X}^{-\frac{1}{2}} K \mathbf{X} \mathbf{X}^{-\frac{1}{2}})^t \ \mathbf{X}^{\frac{1}{2}} \\ &= K^t \ \mathbf{X} \\ &= K^t \ \mathbf{I}_d. \end{aligned} \tag{1.94}$$

Moreover, the logarithm mapping of a matrix $\mathbf{X}$ on the tangent space at the reference point $\bar{\mathbf{M}} = K\mathbf{I}_d$ is equivalent to the projection onto the tangent plane at the identity matrix $\mathbf{I}_d$ up to a scale factor K such as:

$$\mathrm{Log}_{\bar{\mathbf{M}}}(\mathbf{X}) = K \ \mathrm{Log}_{\mathbf{I}_d}(\mathbf{X}) = K \ \mathrm{logm}(\mathbf{X}). \tag{1.95}$$

As a result, there is no distortion anymore. To illustrate that, the same experiment displayed in Figure 1.7 is conducted in Figure 1.9 where the Hausdorff distance is computed as a function of the geodesic distance between the identity matrix and $\bar{\mathbf{M}} = K\mathbf{I}_d$. As demonstrated, in this case, the distance remains close to zero. To visualize samples behaviour, the model at the point (e)

of Figure 1.9 is illustrated in Figure 1.10 where $\bar{\mathbf{M}} = \begin{pmatrix} 90 & 0 \\ 0 & 90 \end{pmatrix}$.

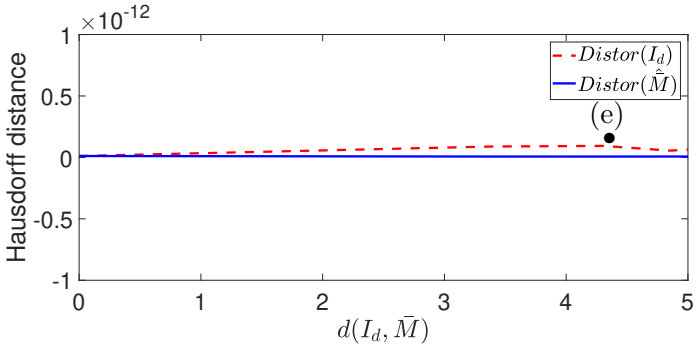


Figure 1.9: Hausdorff distance comparison as a function of geodesic distance between $\mathbf{I_d}$ and $\bar{\mathbf{M}}$.

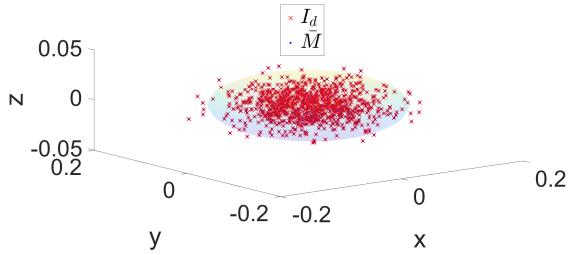


Figure 1.10: Samples behaviour on the tangent space at the identity matrix $\mathbf{I}_d$ and representation of the ellipse at 95 % confidence interval for a set of 3-D normally distributed samples. (e): Experiment Distor($\mathbf{I}_d$) with $\bar{\mathbf{M}} = K\mathbf{I}_d$.

As observed, since $\bar{\mathbf{M}}$ is proportional to the identity matrix, the distance between the two projections remains small and the Gaussian modeling is ensured.

To conclude, the use of an adapted tangent plane relies on data properties. In fact, considering multiple tangent plane model may be useful in the case of covariance matrices which are located far from the identity matrix, without a proportionality link to $\mathbf{I}_d$. As shown, the projection distortion is limited since covariance matrices remain close to the chosen reference point. Furthermore, the distortion power is related to (i) the distance between the considered data generated at $\bar{\mathbf{M}}$ and the targeted tangent plane, here the identity matrix $\mathbf{I}_d$ and (ii) the way in which $\bar{\mathbf{M}}$ is far from $\mathbf{I}_d$. The distortion will increase by increasing the distance between the two points $\bar{\mathbf{M}}$ and $\mathbf{I}_d$ except in the special case of proportionality where $\bar{\mathbf{M}} = K\mathbf{I}_d$ for any scalar K. Moreover, since we are using different reference points $\bar{\mathbf{M}}_k$ for each cluster k and thus different tangent planes on the manifold, performing EM algorithm for parameter estimation requires to transport data to a common tangent plane to ensure numerical stability via the parallel transport operation (1.87).

1.5 Conclusion

In this chapter, the space of symmetric positive definite matrices has been introduced. First, methods insuring fast computation of covariance matrices and robustness to outliers have been discussed. Then, the differential geometry, Riemannian manifolds, and related tools to the space of symmetric positive definite matrices have been introduced. This chapter

provides only a brief overview of the underlying deep theory. Since Euclidean computations have proven to not be well adapted, non-Euclidean alternatives have been considered for their characterization. In particular, the affine-invariant and log-Euclidean metrics were introduced as being the strongest in terms of their invariance properties.

Two complete Riemannian frameworks for SPD matrix sample sets characterization have been developed. They rely on the intrinsic affine-invariant and log-Euclidean metrics. In addition to the LE and AI Gaussian models, their corresponding mixture models have been introduced as well as the parameter estimation process for every case.

Then, a comparison between the two characterizations has been assessed. In fact, while the AI metric space is endowed with stronger invariance properties, estimating the model parameters relies on recursive estimation algorithms, which results on high computational expenses. Furthermore, the LE metric is endowed with relatively similar invariance properties, but, as its mapping on the tangent space allows the vector-form representation of covariance matrices, the complexity and computational expenses associated to the algorithms on the LE metric space are significantly reduced.

Regarding the corresponding GMM models, close conclusions are drawn. In addition, considering the AI metric leads to very complex calculations whereas considering the Log-Euclidean metric, the GMM modeling is limited to a single tangent plane defined at the identity matrix. In fact, projecting covariance matrices on a tangent plane can lead to projection distortions. Thus, the projection behaviour related to covariance matrix set distance to the identity matrix has been analyzed where two cases were depicted. The special case of proportionality between the covariance matrices centroid and the identity matrix permits to get rid of the distortions. In order to better preserve the geometrical data properties as well as reducing the computational cost, the main contribution of this chapter remains on the proposition of an alternative of GMM modeling. It is based on the choice of reference point for data projection onto the tangent plane. A GMM model defined at different reference points is introduced, one per component of the GMM. As such, the structure of the observed covariance matrices is captured more accurately regardless the distance to the identity matrix.

Finally, by using covariance matrices as data features we enjoy a few key advantages. They benefit of useful geometrical properties and a well-developed Riemannian framework, as described in this chapter. Particularly, they are endowed with appropriate Riemannian metrics, facilitating data samples comparisons and distance computation, which are basic ingredients of many analysis and learning techniques. In the following of this work, we build classification algorithms based on covariance matrices aiming at considering dependencies between image features and improving classification performances.

Chapter 2

Ensemble learning approaches based on covariance pooling of CNN Features

Contents

2.1 Introduction

The aim of a supervised classification algorithm consists of labeling an image with the corresponding class according to its content. Conventional approaches are based on encoding handcrafted features with, for example, the bag of words model (BoW) [Sivic *et al.* 2005], the vector of locally aggregated descriptors (VLAD) [Jégou *et al.* 2010, Arandjelović & Zisserman 2013] or the Fisher vectors (FV) [Perronnin & Dance 2007, Perronnin *et al.* 2010b, Perronnin *et al.* 2010a]. These latter strategies allowed to obtain successful results in a wide range of applications such as image classification [Perronnin & Dance 2007, Douze *et al.* 2011, Sánchez *et al.* 2013], text retrieval [Salton & Buckley 1988], action and face recognition [Faraki *et al.* 2015a], etc.

Recently, the emergence of deep learning algorithms has been demonstrated to outperform benchmark machine learning methods in many situations. In fact, neural networks are constructed to model the human brain, where each layer is responsible for automatically extracting and learning specific features from the input images [Kriegeskorte 2015]. One of the most popular neural networks is the convolutional neural network (CNN), which has become a standard for image classification problems [Le Cun *et al.* 1990, Krizhevsky *et al.* 2012]. CNN is built from various hidden layers performing different kinds of transformations, such as convolutions, pooling, and activation functions.

In recent years, in order to benefit from both CNN architectures and encoding methods, many authors have focused on proposing hybrid architectures that consist of combining deep neural network architectures with FV/VLAD descriptors. For example, Perronnin et al. have introduced in [Perronnin & Larlus 2015] a network of fully connected layers trained on the FV descriptors. Inspired by the multi-layer structure of neural networks, Simonyan et al. proposed in [Simonyan *et al.* 2013], the Fisher network, which is composed of several stacked FV layers. In the same spirit, the NetVLAD layer has been proposed in [Arandjelovic *et al.* 2015] to mimick a VLAD layer. To benefit of multi-layer representation, other strategies have been proposed to include the FV or VLAD encoding of CNN features from different layers of the network [Ng *et al.* 2015, Cimpoi *et al.* 2016, Diba *et al.* 2017, Li *et al.* 2017]. Nevertheless, all these strategies do not exploit second-order statistics, i.e., dependencies between features, which have been shown to be important in the human visual recognition process.

To this aim, some authors have dedicated their works to exploit the information behind second-order statistics using covariance matrix features. These have proved to be highly effective in diverse classification tasks, including person re-identification, texture recognition, material categorization or EEG classification in brain–computer interfaces to cite a few of them [Faraki *et al.* 2015a, Barachant *et al.* 2013, Said *et al.* 2015a, Kong & Fowlkes 2016]. Several works have been proposed to extend the encoding formalism to covariance matrix descriptors. Therefore, as explained in chapter 1, covariance matrices are symmetric positive definite (SPD) matrices and conventional Euclidean tools are not adapted. To deal with covariance matrices geometry, two Riemannian metrics are usually considered: the log-Euclidean and the affine-invariant Riemannian metrics. Since then, some authors have proposed to extend the usual coding methods to these two metrics, yielding to the proposition of the following approaches: the log-Euclidean bag of words (LE BoW) [Yuan *et al.* 2010, Faraki *et al.* 2015b],

the bag of Riemannian words (BoRW) [Faraki *et al.* 2014], the log-Euclidean vector of locally aggregated descriptors (LE VLAD) [Faraki *et al.* 2015a] and the intrinsic Riemannian vector of locally aggregated descriptors (RVLAD) [Faraki *et al.* 2015a]. Recently, FV descriptors extended to SPD matrices have been proposed. This has involved the log-Euclidean Fisher vectors (LE FV) [Akodad *et al.* 2018b] and the Riemannian Fisher vectors (RFV) [Ilea *et al.* 2016, Ilea *et al.* 2018a, Ilea *et al.* 2018b]. When analyzing those two metrics, log-Euclidean and affine-invariant Riemannian metrics offer several invariance properties and obtain comparable results for a large variety of applications [Ilea *et al.* 2018b, Arsigny *et al.* 2006] compared to the Euclidean metric. However, the log-Euclidean approach is much more straightforward. To model covariance matrices that lie in a Riemannian manifold, it merely consists in projecting them in a tangent space of a reference point classically chosen equal to the identity matrix. Since then, the Log-Euclidean metric is exploited in the following of this chapter.

To benefit from both second-order statistics and deep learning architectures, different second-order convolutional neural network architectures have recently emerged [Ionescu *et al.* 2015, Cai *et al.* 2017, He *et al.* 2018, Huang & Gool 2017, Yu & Salzmann 2017, Acharya *et al.* 2018, Gao *et al.* 2019, He *et al.* 2020] for many applications including fine-grained classification. One first attempt was the pooled covariance matrix from CNN outputs [Ionescu *et al.* 2015]. Later, He et al., presented in [He *et al.* 2018] a multi-layer version: the multi-layer stacked covariance pooling (MSCP). Another way to exploit second-order statistics in a deep neural network is the Riemannian SPD matrix network (SPDNet) [Huang & Gool 2017]. The idea behind this network is to mimick the classical CNN fully connected convolution-like layers and rectified linear units (ReLU)-like layers to data, which lie in an Riemannian manifold. For that, the bilinear mapping (BiMap) layers and eigenvalue rectification (ReEig) layers were proposed. Inspired by this work, Yu et al. have introduced in [Yu & Salzmann 2017] a second-order CNN (SO-CNN), which is trained in an end-to-end manner. However, for these models, second-order representation is introduced only for the deepest layers. To overcome this issue, Gao et al. [Gao *et al.* 2019] have proposed the global second-order pooling (GSoP) convolutional networks which permit to introduce higher-order representation in earlier layers. Nevertheless, training such a deep CNN model from scratch requires a huge labeled training set. Recently, the remote sensing community has started to build large scale datasets that can serve as pre-training, such as the BigEarthNet composed by Sentinel-2 image patches [Sumbul *et al.* 2019]. In addition, they provide on their website CNN models trained on the BigEarthNet dataset, the specificity of those CNN models, compared to the classical models such as the VGG-16 network, lies in the nature of the input images. In fact, they operate on multispectral images where BigEarthNet images are constitued of 10 spectral bands. When it comes to classify remote sensing datasets, successful results were obtained with those models where, for the same model, classification performance are higher than with the one trained on ImageNet dataset. However, for many practical applications, most of the remote sensing datasets are quite small.

Many authors have proposed several ideas to overcome this issue such as using a new kind of neural network called capsule network [Souleyman *et al.* 2019] which has the ability to work with a small amount of training data. Compared to convolutional neural network, capsule network allows to address the “Picasso problem" in image recognition, i.e. images that show the right components but have not the right spatial relationships. For example, for a face image,

the location of the eye and ear are swapped. For our application of remote sensing scene classification, this is not critical. For instance, in a harbour scene, the location of the scene elements (boats, pontoon, ...) in the image is not so important. The key point is that the network is able to recognize them. Another effective solution for limited training set consists of transfer learning. In that case, CNN models are considered as feature extractors. Classically, deep CNN models pre-trained on the ImageNet dataset are used. Then, features are extracted from a single or multiple layers, and are processed with some machine learning algorithms. This technique has been proved to be efficient and permits to outperform traditional hand-crafted feature-based methods [Krizhevsky *et al.* 2012]. In a recent paper, Pires de Lima et. al. have shown that transfer learning strategies based on feature extraction are among the best approaches for remote sensing scene classification, especially for dataset with a low number of training samples [Pires de Lima & Marfurt 2019]. In this context, in order to benefit of pre-trained deep neural networks and second-order representations, this chapter aims to propose a novel ensemble learning approach based on covariance pooling of CNN features for remote sensing scene classification. It consists of a combination of two hybrid architectures exploiting second-order features. The former is based on the log-Euclidean Fisher vector encoding of region covariance matrices computed locally on the first layers of a CNN [Akodad *et al.* 2018b] and its extension to the use of an ensemble learning strategy to combine multiple classifiers. The latter concerns an ensemble learning approach based on the covariance pooling of CNN features extracted from deeper layers [Akodad *et al.* 2019c]. All the discussed strategies can be summarized in the following timeline in Figure 2.1. It highlights the three families of image classification approaches in a chronological order, first based on hand-crafted feature extraction and encoding methods, then followed by the deep learning based methods. The emergence of deep convolutional layers leads to the introduction of hybrid strategies and their extension to second-order statistics.

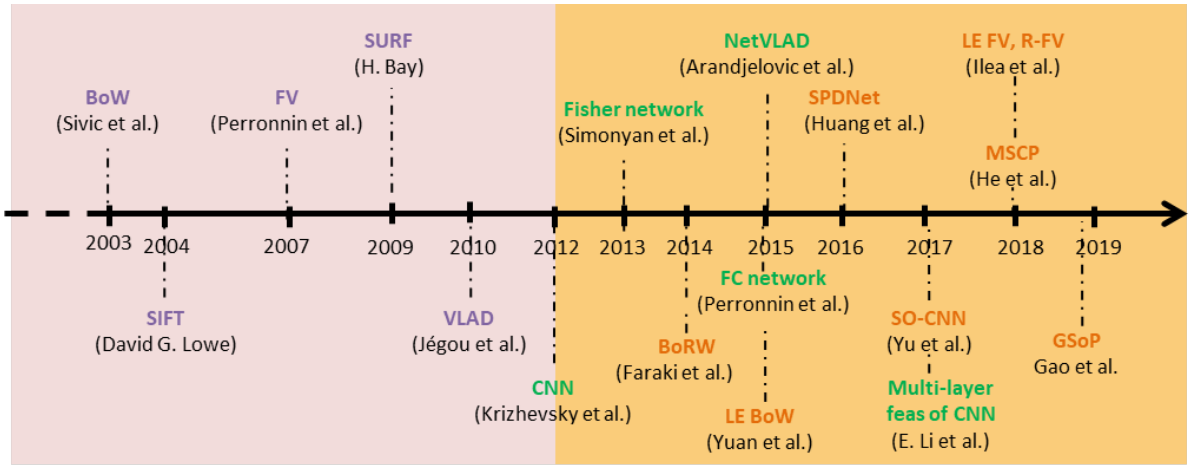


Figure 2.1: Time line of machine learning and deep learning based methods. Purple: machine learning methods including features extraction and encoding strategies. Green: Hybrid methods combining machine learning and deep learning approaches. Orange: Methods exploiting second-order statistics.

In summary, second-order representation (i.e. covariance pooling) has been shown to be useful for many signal and image processing tasks. Recently, in the remote sensing community, some works have shown interest in these second-order features for various remote sensing applications (*e.g.* remote sensing scene classification, texture recognition) [He *et al.* 2018, He *et al.* 2020, Rosu *et al.* 2017, Pham *et al.* 2017]. Motivated by these works and the success of deep neural networks, this chapter introduces two hybrid transfer learning approaches based on covariance

pooling of CNN features [Akodad *et al.* 2018b, Akodad *et al.* 2019c]. These two methods use either local or global second-order representation of CNN features. The main motivation of this chapter is to unify these works by presenting a transfer learning approach which benefit from these approaches. The main contributions of this chapter can be summarized as follows:

- We propose **a transfer learning approach, which efficiently combine local and global second-order representation of CNN features**. For the local one, an ensemble learning extension of our log-Euclidean Fisher vector encoding of region covariance matrices [Akodad *et al.* 2018b] is introduced. For the global one, our covariance pooling of deepest CNN features is considered [Akodad *et al.* 2019c]. In addition, **the use of saliency maps** is adopted to enhance classification performance regarding small objects of interest.

- **An ensemble learning approach based on the most diverse ensembles** is proposed to combine these decisions and enhance the classification performance [Akodad *et al.* 2020c].

- This transfer learning is **validated on different labeled remote sensing datasets** to illustrate its efficiency. Three are publicly available, namely UC Merced Land Use Land Cover [Yang & Newsam 2010], SIRI-WHU [Zafar & Ali 2019] and AID [Xia *et al.* 2017] datasets. Two others are internal datasets, oyster racks and maritime pine forest datasets, which are manually labeled by thematic experts [Regniers 2014].

The chapter is structured as follows. Section 2.2 introduces different machine learning and deep learning based methods dedicated to image classification tasks and some hybrid architectures that result from combining those latter two families. Then, since the second-order representation of CNN features is at the core of the study, and the mathematical background for the log-Euclidean representation of a covariance matrix is explained in chapter 1, Section 2.3 introduces the proposed ensemble learning approach based on the log-Euclidean Fisher vector encoding of region covariance matrices. Then, Section 2.4 recalls our ensemble learning approach based on covariance pooling (ELCP) of CNN features. In order to combine these two methods, Section 2.5 presents the fusion scheme based on the most diverse ensembles. Next, Section 2.6 summarizes a series of experiments performed on remote sensing scene classification. And finally, Section 2.7 provides conclusions and perspectives for this work.

2.2 Image classification algorithms based on traditional machine learning and deep learning methods

Image classification refers to the labeling of images into one predefined class according to its content. Before the emergence of deep learning methods, a first step called feature extraction was carried out for tasks such as image classification. Features, such as edges and interest points, provide rich information on the image permitting the representation of their content. Several algorithms, such as edge detection, corner detection or threshold segmentation may be involved in this step. A model is then learned on the space of those features. At the classifier stage, these features are searched for in other images using well-known algorithms such as decision trees, random forests or support vector machines.

The rise of deep learning methods over the last several years leads to drastic improvements for many computer vision tasks. The adjective "deep" refers to the use of multiple layers in the network which permit to progressively and automatically extract different levels of features from the raw input data.

Two different workflows are illustrated in Figure 2.2 to assess a comparison between traditional machine learning methods, involving handcrafted feature extraction and a visual codebook learning, and deep learning methods based on artificial neural networks (NN).

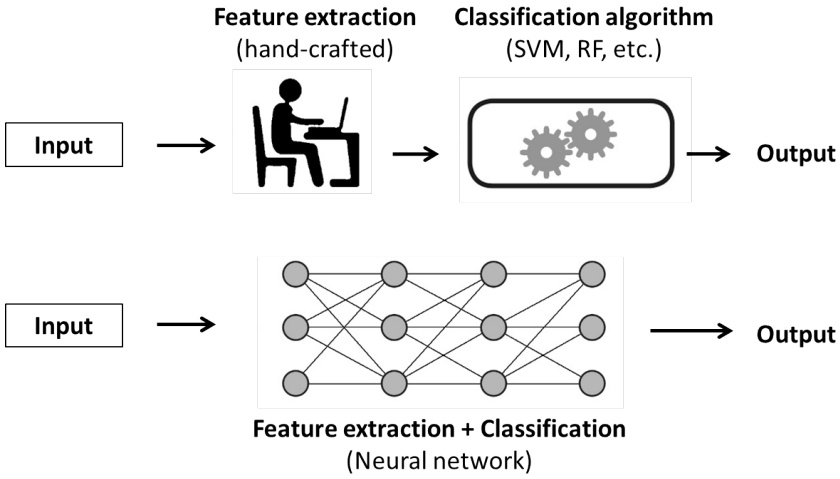


Figure 2.2: Difference between traditional machine learning and deep learning for image classification

In the following, the principal steps of an image classification problem whether using machine learning or deep learning strategies are detailed.

2.2.1 Machine learning strategies

The framework used to classify an image through machine learning methods for feature extraction and model training is illustrated in the Figure 2.3.

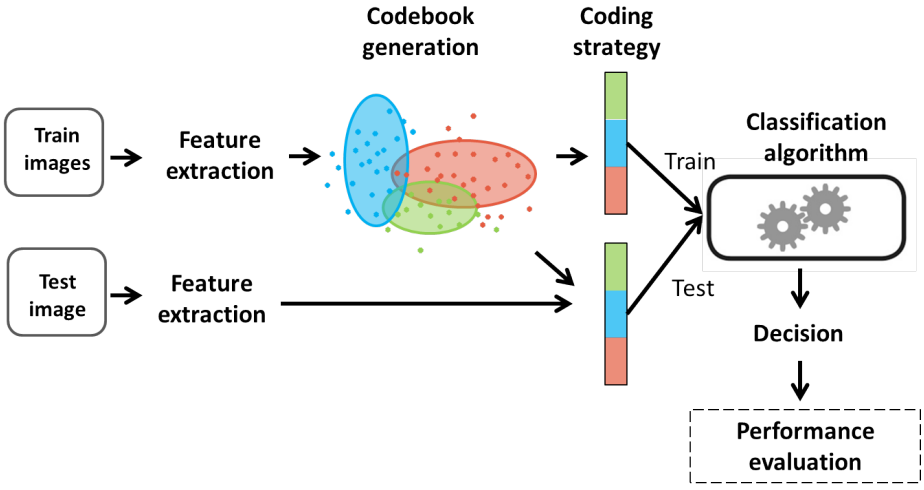


Figure 2.3: Classification workflow based on traditional machine learning strategies

The separation between training and testing sets is performed so that an equal number of images are randomly drawn from each class. In order to configure this separation, we have

defined for all simulations a factor p that corresponds to the percentage of learning images selected for each class. Then, the classification algorithm is performed, where it is composed of different steps: feature extraction, codebook generation, features encoding and a classification algorithm. Each of these steps are detailed on the following subsections.

2.2.1.1 Feature extraction

This is a crucial step where statistical methods are used to identify the most interesting patterns of the image, called features, that might be unique to distinguish a particular class and that will, later on, allow the model to differentiate between different classes. The traditional approach is to use well-established techniques such as feature descriptors (SIFT [Lowe 2004], SURF [Bay *et al.* 2006], BRIEF [Calonder *et al.* 2012], etc.) for object detection.

This first step is ensured by different methods either global; calculated on the whole image, or local which consist of mathematical transformations on a limited number of pixels around the points of interest of each image, *i.e.* a patch. The local description of an image is an approach applied in many areas of vision such as pattern recognition, tracking, reconstruction, calibration, etc. It is based on matching (or mapping) points of interest characterized by a local descriptor. Depending on the application, some invariances are necessary. In pattern recognition, one of the most commonly used feature is the SIFT descriptor [Lowe 2004].

The SIFT algorithm, which stands for Scale Invariant Feature Transform, is a method developed by David Lowe [Lowe 2004] and is used to identify similar points between images at different scales. This identification goes through two stages. First, extraction of SIFT descriptors by calculating a gradient orientation histogram. It involves transforming an image into a set of vectors of characteristics which are invariant by geometric transformations. Next comes the mapping step by comparing the descriptor vectors of two images in order to detect an object or to conclude on the transformation undergone. The focus here is around the calculation of SIFT descriptors. It consists of steps listed below:

- **Construction of the pyramid**: This step allows to analyze the images with a mutli-scale approach. To do this, the Gaussian pyramid is used to sub-sample and smooth the image gradually. This reduces the size of the image by four at each level. Concretely, this pyramid is set up by applying to the same image Gaussian filters of different variances σ^2. The operation is repeated on the different scales of the image. The convolution between the starting image I and a Gaussian filter G produces the smoothed image L, which is called: the gradient of a scale factor σ. It is given by:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.1}$$

  where $*$ is the convolution operator, and (x, y) are the pixel coordinates and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{\frac{-(x^2+y^2)}{2\sigma^2}}. \tag{2.2}$$

- **Difference of Gaussians (DoG)**: The difference between two consecutive images of each octave in the pyramid allows to obtain a pyramid of DoG and identify potential points of interest that are invariant to scale and rotation.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{2.3}$$

with σ and $k\sigma$ define two consecutive images of an octave.

These DoG images are great for finding out interesting keypoints in the image.

- **Local extrema detection**: This difference between two consecutive and smoothed images makes it possible to detect points of interest on several scales and different levels of resolution. Each pixel of a DoG is compared to its 8 neighbors and 9 neighbors in the next and previous DoG. The recovered point is therefore considered to be a local extremum. By setting a threshold, the low contrast points are rejected as well as the unstable points located at the edges.

- **Orientation Assignment**: Each keypoint is assigned one or more orientations calculated from the gradient distribution of the neighbouring points. The gradient, amplitude $m(x, y)$ and orientation $\theta(x, y)$ of each point of the Gaussian image $L(x, y, \sigma_0)$ of the neighborhood of $L(x_0, y_0, \sigma_0)$ are estimated by establishing a histogram of the orientations. As shown in Figure 2.4, the final SIFT descriptor is a 4×4 descriptor array which summarizes the contents over considered subregions. Each cell corresponds to the sum of the gradient magnitudes within the region. It actually shows orientation histograms of eight directions each, where the length of each arrow corresponds to the magnitude of that histogram entry.

Figure 2.4: The descriptor of a SIFT frame centered at a fixed pixel

Then, these features are fed into a learning algorithm for their classification. To do this, the set of descriptors is separated into two categories. One for learning the dictionary in order to encode it, using for example a Gaussian mixture model (GMM), and the other set, namely the testing set, is encoded based on the learnt codebook, it also allows assessing the classification performance.

2.2.1.2 Codebook generation

The purpose of this step is to cluster features into visual code words which means that the space of visual descriptors is divided into several regions. Usually, this procedure is performed by means of clustering algorithms, such as the k-means or expectation-maximization (EM) algorithm by considering a Gaussian mixture model (GMM). When performing the EM algorithm, modeling considers both cluster centers and covariances, which describe the location and shape

of clusters, whereas for k-means clustering, despite being relatively simple to implement and computationally fast, the method is limited by the underlying assumptions of homoscedasticity. Here, the interest is toward the GMM modeling involving the EM algorithm to estimate the Gaussian parameters in an unsupervised manner. Considering $\mathbf{X} = \{\mathbf{x_i}\}_{i=1,\dots,N}$ an N-sample of d dimensional observations modeled by a Gaussian mixture model with K components as follows:

$$p(\mathbf{X}|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \omega_k p(\mathbf{x}_i|\mu_k, \Sigma_k), \tag{2.4}$$

where $\theta = \{(\omega_k, \mu_k, \Sigma_k)_{1\leq k\leq K}\}$ is the parameter vector with $\omega_k \in [0, 1]$ the mixture weight, μ_k the mean vector and Σ_k the covariance matrix. Thus:

$$p(\mathbf{x}_i|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \mu_k)\right\}. \tag{2.5}$$

As detailed in chapter 1, the set of features $\theta = \{(\omega_k, \mu_k, \Sigma_k)_{1\leq k\leq K}\}$ is estimated using the EM algorithm and partitioned into a predefined number of clusters, where a description of each cluster k is made by computing the following parameters: the weight ω_k, the cluster's centroid μ_k and the dispersion Σ_k. Those obtained parameters are called codewords and are grouped on a codebook, also called a dictionary. Since the codebook is used as the basis for encoding feature vectors, several methods permitting the features encoding are described.

2.2.1.3 Descriptor encoding

Based on the created codebook, the descriptor encoding aims to transform collections of local image features into fixed-size vector representations. The general idea consists of projecting the extracted features onto the codebook. In this work, we investigate three different descriptor encoding schemes, namely the bag of visual words (BoW) [Csurka *et al.* 2004], vector of locally aggregated descriptors (VLAD) [Jégou *et al.* 2010] and Fisher vectors (FV) [Perronnin & Dance 2007, Perronnin *et al.* 2010b, Perronnin *et al.* 2010a].

- **Bag of Words (BoW):**

At first, the bag-of-words model, as its name suggests, has been employed in problems such as language modeling and document classification [Salton & Buckley 1988, Joachims 1998]. It is considered as a simple technique to represent a text by describing the occurrence of words within a document in a histogram as illustrated in Figure 2.5. It has been then extended to visual categorization such as image characterization [Csurka *et al.* 2004] where the "words" are replaced by the image features. Therefore, each image is described by the number of occurrences of these patterns. For that, it involves two steps. First, the codebook is created, using a clustering algorithm such as the k-means and data is partitioned in different regions by assigning each sample to the closest centroid. Then the computation of the histogram of occurrences of each codeword is performed. For classification purpose, a nearest neighbor classifier is classically employed to measure the distance between two histograms. For that, the χ^2 metric is generally employed. It encodes the zero-order statistics of the distribution of local descriptors.

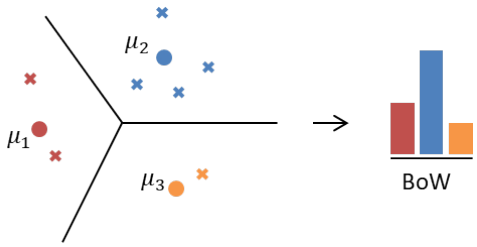


Figure 2.5: Visual word vector computation.

- **Vectors of locally aggregated descriptors (VLAD):**

VLAD [Jégou *et al.* 2010] has been originally proposed for image retrieval application and permits encoding a set of descriptors into a dictionary.

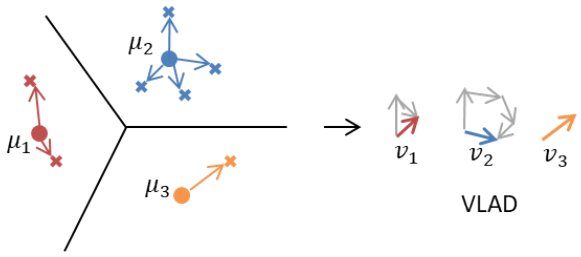


Figure 2.6: VLAD vectors computation.

Let's consider a set $\mathbf{X} = \{\mathbf{x_i}\}_{i=1,\dots,N}$ with $\mathbf{x_i} \in \mathbb{R}^d$ a d dimensional feature of N extracted observations from an image. The k-means algorithm is usually performed to partition the set into K clusters, defined by their centroids. The homoscedasticity assumption should also be made *i.e.* $\sigma_k = \sigma$. For each cluster c_k of the codebook, the vector $\mathbf{v}_k$ contains the sum of differences between the feature sample and the cluster centroid μ_k it is assigned to:

$$\mathbf{v}_k = \sum_{i=1}^{N} \gamma_k(\mathbf{x}_i)\ (\mathbf{x}_i - \mu_k). \tag{2.6}$$

where $\gamma_k(\mathbf{x}_i)$ denotes the membership of the descriptor $\mathbf{x}_i$ to k^{th} cluster. In this original version, it is a hard assignment with $\gamma_k(\mathbf{x}_i) = 1$ if cluster c_k is the closest cluster to observation $\mathbf{x}_i$ and $\gamma_k(\mathbf{x}_i) = 0$ otherwise. To summarize, the VLAD encoding of $\mathbf{X}$ is obtained as the concatenation of vectors $\mathbf{v}_k$:

$$\mathbf{VLAD} = [\mathbf{v}_1^T, \dots, \mathbf{v}_K^T]. \tag{2.7}$$

The obtained VLAD vector is of dimension $K \times d$.

- **Fisher vectors (FV):**

The Fisher vector encoding of a set of features, such as SIFT features, is based on fitting a parametric generative model, i.e. the Gaussian Mixture Model (GMM), to the features, and then encoding the derivatives of the log-likelihood of the model with respect to its parameters, i.e. the mixture weights, means and variances. The computation of Fisher vectors is based on Fisher kernels [Jaakkola & Haussler 1998] which represents methods allowing to assess samples fitting to models.

Intuitively, the encoding describes how the distribution of features of a particular image differs from the distribution fitted to the features of all training images. Let $\mathbf{X} = \{\mathbf{x_i}\}_{i=1,\dots,N}$

with $\mathbf{x_i} \in \mathbb{R}^d$ a d dimensional sample of N extracted features from an image. Suppose we have a generative model $p(\mathbf{X}|\theta)$. We can map $\mathbf{X}$ into a vector by computing the gradient vector of its log-likelihood function with respect to the model's parameters θ:

$$\mathcal{G}_\theta^{\mathbf{X}} = \mathbf{F}_\theta^{-\frac{1}{2}} \nabla_\theta \log p(\mathbf{X}|\theta), \tag{2.8}$$

where $\mathcal{G}_\theta^{\mathbf{X}}$ is the Fisher vector which can be seen as the deviation of the direction to make parameters θ fit better to $p(\mathbf{X}|\theta)$. Classically, the gradient of the log-likelihood is normalized by the square-root of the inverse of the Fisher Information Matrix $\mathbf{F}_\theta$ [Perronnin & Dance 2007].

In the particular case of GMMs with diagonal covariances, Fisher vectors lead to the representation which captures the average first and second-order differences between the observations and each of the GMM centers. Considering the GMM model defined in (2.5), diagonal covariance matrices are usually assumed to simplify the model and thus reduce the Fisher vector size. Then, $\sigma_k^2 = \text{diag}(\Sigma_k)$ is the variance vector. As a result, the derivatives of each dimension $j = 1, \ldots, d$ with respect to the parameters θ are computed.

$$\frac{\partial \log p(\mathbf{X}|\theta)}{\partial \mu_k(j)} = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left(\frac{\mathbf{x}_n(j) - \mu_k(j)}{\sigma_k^2(j)} \right), \tag{2.9}$$

$$\frac{\partial \log p(\mathbf{X}|\theta)}{\partial \sigma_k(j)} = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left(\frac{(\mathbf{x}_n(j) - \mu_k(j))^2}{\sigma_k^3(j)} - \frac{1}{\sigma_k(j)} \right), \tag{2.10}$$

$$\frac{\partial \log p(\mathbf{X}|\theta)}{\partial \omega_k(j)} = \sum_{n=1}^{N} (\gamma_k(\mathbf{x}_n) - \omega_k), \tag{2.11}$$

where $\gamma_k(\mathbf{x}_n)$ is the soft assignment of $\mathbf{x}_n$ to the k^{th} component defined as:

$$\gamma_k(\mathbf{x}_n) = \frac{\omega_k \; p(\mathbf{x}_n|\mu_k, \sigma_k)}{\sum_{j=1}^{K} \omega_j \; p(\mathbf{x}_n|\mu_j, \sigma_j)}. \tag{2.12}$$

The diagonal Fisher Information Matrix (FIM) [Perronnin & Dance 2007] can be taken into account by a coordinate-wise normalization of the obtained gradient vectors, which yields the following normalized gradients:

$$\mathcal{G}_{\mu_k(j)}^{\mathbf{X}} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left(\frac{\mathbf{x}_n(j) - \mu_k(j)}{\sigma_k(j)} \right), \tag{2.13}$$

$$\mathcal{G}_{\sigma_k(j)}^{\mathbf{X}} = \frac{1}{\sqrt{2\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n) \left(\frac{(\mathbf{x}_n(j) - \mu_k(j))^2}{\sigma_k^2(j)} - 1 \right), \tag{2.14}$$

$$\mathcal{G}_{\omega_k(j)}^{\mathbf{X}} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} (\gamma_k(\mathbf{x}_n) - \omega_k). \tag{2.15}$$

The final Fisher vector is the concatenation of the Fisher vectors $\mathcal{G}_{\mu_k}^{\mathbf{X}}$, $\mathcal{G}_{\sigma_k}^{\mathbf{X}}$ and $\mathcal{G}_{\omega_k}^{\mathbf{X}}$ for $k = 1, \ldots, K$. It leads to a dimension of $(2d+1)K$ where $\mathcal{G}_{\omega_k}^{\mathbf{X}}$ is a scalar while $\mathcal{G}_{\mu_k}^{\mathbf{X}}$ and $\mathcal{G}_{\sigma_k}^{\mathbf{X}}$ are d-dimensional vectors. In [Sánchez *et al.* 2013], it has been demonstrated that the combination of $\mathcal{G}_{\mu_k}^{\mathbf{X}}$ and $\mathcal{G}_{\sigma_k}^{\mathbf{X}}$ are the most discriminating descriptors compared to $\mathcal{G}_{\omega_k}^{\mathbf{X}}$.

To summarize, BoW can be considered as a special case of the FV encoding where the gradient computation is restricted to the mixture weight parameters of the GMM defined in (2.15) under the hypothesis of hard thresholding. In addition, the VLAD encodes the first-order information rather than the zero-th order information (counts) and gains in recognition accuracy. It also can be thought as a simplified version of the FV encoding where it uses k-means clustering, and switches from soft to hard assignment. Starting from (2.13), the VLAD vector is defined for a binary assignment $\gamma_k(\mathbf{x}_i)$ where it is equal to 1 if cluster c_k is the closest cluster to descriptor $\mathbf{x}_i$ and 0 otherwise and σ_k^2 a constant. As such, (2.13) induced to (2.6) for the hard assignment hypothesis.

2.2.1.4 Classification algorithms

This step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns. In fact, the dataset of interest is divided into training and testing dataset. The chosen model uses the training dataset and calculate how to best map samples of input data to specific class labels. As such, the training dataset must be sufficiently representative of the problem. There are many different types of classification algorithms where the most popular are: the support vector machine (SVM), the decision trees, random forests and the K-nearest neighbors (KNN). These four methods are briefly detailed in the following.

- **Support vector machine (SVM):**

It is a supervised machine learning algorithm used for both regression and classification problems. When used for classification purposes, it separates the classes using a linear boundary by building a hyper-plane. The separation between two classes is achieved by the hyperplane that has the largest distance to the nearest training data point of any class, such distance being called margin. In the Figure 2.7, SVM needs to find the optimal line with the constraint of correctly classifying

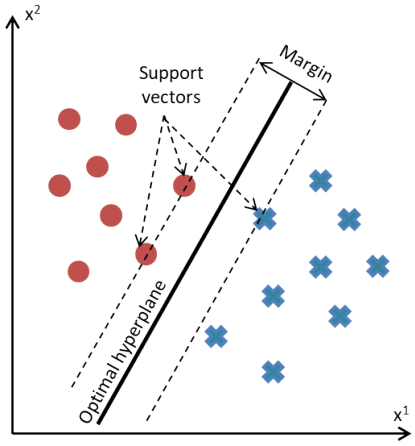


Figure 2.7: Illustration of the SVM hyperplane which best separates the two classes.

either class of red dots or blue crosses. In real life, the training data can be rarely separated using an hyperplane and there is a need to transform the data into a higher dimensional space in order to fit a support vector classifier. This transformation is made by functions that can map the data to any number of higher dimensions. This illustrates the power of this algorithm, where different kernel functions can be used. The most commonly used kernels are: linear kernel, Gaussian kernel and polynomial kernel.

- **Decision trees and random forest (RF):**

Decision trees and random forests are also supervised machine learning algorithms based on decision tree data structure. They use a series of if/else statements on the feature space at each intermediary stage/level. They proceed generally in three steps: Partitioning the nodes, finding the terminal nodes and allocating the corresponding class label to terminal node.

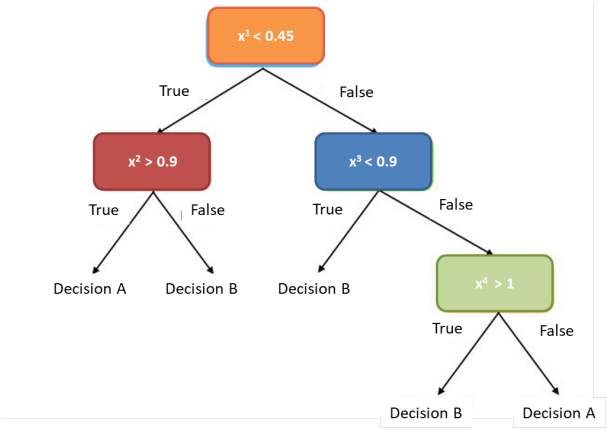


Figure 2.8: A simple decision tree classifier with 4 features.

In most supervised problems, a single tree is not sufficient to produce stable results. This is where the Random Forest algorithm comes into the picture. The name random forest refers to Breiman's work in [Breiman 2001] and the method aims at combining several decision trees where each node in the decision tree works on a random subset of features, called bootstrap sample, to calculate the output. During this phase, at each node of the tree, a splitting rule is designed by selecting a feature over used features chosen uniformly at random among the initial features. This selection can be performed by maximizing the well-known Gini impurity criterion [Raileanu & Stoffel 2004]. The algorithm operates until a stopping criterion is achieved and the output of individual decision trees are fused to generate the final decision which would be more accurate and stable.

- **K-Nearest Neighbor (K-NN):**

The k-nearest neighbor is by far the most simple machine learning algorithm. This algorithm simply relies on the distance between feature vectors, classically the Euclidean distance, and classifies unknown data points by finding the most common class among the k-closest examples. Figure 2.9 below illustrates the classification algorithm for the 1 nearest neighbor (1-NN) where the nearest observation to the new sample is from the red dots class. As such, the algorithm will classify it on the red dots class.

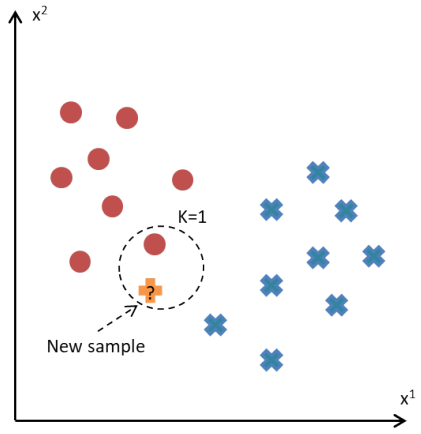


Figure 2.9: Illustration of 1-NN classifier for a two-class problem.

2.2.1.5 Performance evaluation

Once the model is built, the most important question that arises is how good is the model? A learned classifier has to be evaluated based on the testing set which hasn't been fed to the model during the training stage. The experimental performance on the testing data is related to the performance on unseen data and permit validating the classifier's generalization ability. Performance evaluation relies on the use of several performance indicators. Such indicators convey the qualities of an algorithm. Typical performance indicators include:

1. Accuracy: how well the algorithm has performed with respect to some reference;
2. Robustness: an algorithm's capacity for tolerating various conditions;
3. Sensitivity: how responsive an algorithm is to small changes in features;
4. Stability: the degree to which an algorithm, when repeated using the same stable data, yields the same result.

Several evaluation metrics are used in the literature to address classification quality. Table 2.1 provides the most commonly used metrics to evaluate classification performance for a two-class problem. Most of them, such as the precision and recall, are defined in terms of the cells in the confusion matrix.

Metric	Description	Formula
Confusion matrix (M)	It shows a detailed breakdown of correct and incorrect predictions for each class. For a two-class classification problem, each cell in the table has a specific and well-understood name.	$\mathrm{M} = \begin{bmatrix} \mathrm{TP} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TN} \end{bmatrix}$
Overall accuracy (OA)	It measures how often the classifier makes the correct prediction across all classes. In the general case, it is a ratio between the number of correct predictions and the total number of produced predictions	$\mathrm{OA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+FP+FN}}$
Precision (P)	It identifies how accurately the model predicted the positive classes.	$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
Recall (R)	It is a sensitivity metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. In this way, it provides an indication of missed positive predictions.	$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
F-score (F)	It provides a single measure which capture both properties of overall accuracy and recall metrics. It might be the most common metric used on imbalanced classification problems.	$\mathrm{F} = 2\,\frac{\mathrm{P\times R}}{\mathrm{P+R}}$
Kappa accuracy (K)	Also named Cohen's Kappa, is a useful measure for problems that have an imbalance in the classes. It permits identifying how well the model is predicting by considering the level of expected accuracy obtained by chance p_c	$K = \frac{OA-p_c}{1-p_c}$,

Table 2.1: Metrics for performance evaluation for a two-class problem. TP: True Positives, FP: False Positives (false alarms), FN: False Negatives (misses) and TN: True Negatives (correct rejections).

Overall accuracy is an easy way to evaluate classification performance. However, it doesn't make any distinction between classes. For example, when the classes are imbalanced, i.e., there are different numbers of samples per class, the accuracy will give a very distorted picture, because the class with more samples will dominate the statistic. In that case, the per-class average accuracy is well adapted. It returns a metric corresponding to each class, which is the precision for each class. For the conducted experiment, the overall accuracy is often exploited to quantify classification performance, especially when it comes to balanced datasets. Moreover, per-class average accuracy and kappa accuracy are also evaluated

The difficulty with traditional machine learning approaches is that it is necessary to choose which features are important in each given image. Furthermore, as the number of classes to classify increases, feature extraction becomes more and more cumbersome whereas deep learning

strategies have the ability to automatically adapt to changes by constant feedback and improve the model. It collects data, learns from it, and optimises the model. As such, deep learning algorithms can be regarded as a sophisticated and mathematically complex evolution of traditional machine learning algorithms.

2.2.2 Deep learning based methods

Inspired by the properties of biological neural networks, artificial neural networks are statistical learning algorithms and are used for a variety of tasks, from relatively simple classification tasks to computer vision and speech recognition. Like the human brain, it is composed of many computing cells, namely "neurons" where each performs a simple operation and interacts with each other to make a decision. Deep Learning (DL) techniques are conquering over the prevailing traditional machine learning approaches. They are relatively important when it comes to the large amount of images, applications requiring complex functions and demanding increased accuracy with lower time complexities.

2.2.2.1 Convolutional neural networks (CNN)

Convolutional neural networks (CNNs) introduce a special architecture of artificial neural networks, and were first introduced in the 1980s by Kunihiko Fukushima [Fukushima 1988] which designed an artificial neural network, namely the Neocognitron, for visual pattern recognition. This paper was followed by the work of Yann Le Cun which proposed the LeNet model [Le Cun *et al.* 1998]. It represents a first modern application of convolutional neural networks for handwritten digits recognition. Then, with the availability of large sets of data, namely the ImageNet dataset with millions of labeled pictures, and the increase of computer resources, a multi-layered neural network version appeared in 2012 under the name AlexNet. It refers to its main creator, Alex Krizhevsky. CNNs are composed of multiple layers of artificial neurons, inspired by their biological counterparts, which are mathematical functions that calculate the weighted sum of multiple inputs and output an activation value. A typical CNN is constituted of two parts:

- The convolutional part is composed by a stack of convolutional and pooling layers permitting the extraction of image features.
- The classifier is usually composed by fully connected layers. The main goal of the classifier is to classify the image based on the extracted features.

Figure 2.10 shows the architecture of a model based on CNN. Contrary to machine learning methods, deep learning models can automatically learn hierarchical feature representations. Features computed by the first layer are general, while features computed by the last layer are specific and depend on the chosen dataset and task.

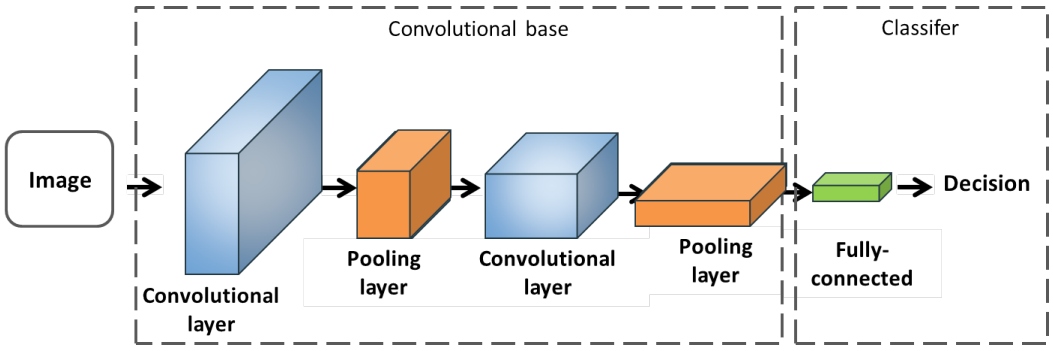


Figure 2.10: Convolutional neural network (CNN) architecture

A CNN is basically composed of elementary layers, namely convolutional, pooling, activation and fully-connected layers. They are explained in the following.

Convolutional layer:

Convolution basically means a pointwise multiplication of two functions to produce a third function, which is then summed. Here, the first function is the input image pixels matrix while the second one is the filter. The convolutional layer performs by sliding the filter over the image and get the dot product of the two matrices as displayed in Figure 2.11.

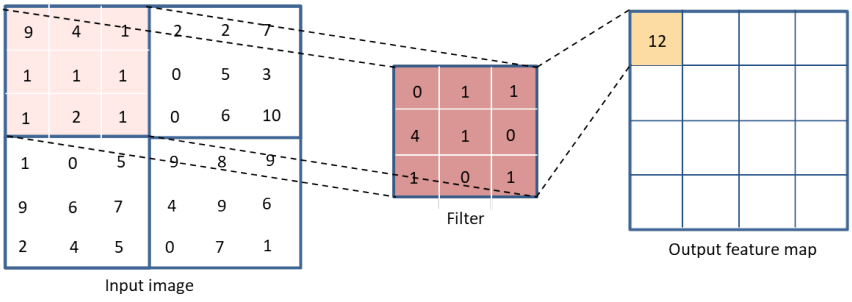


Figure 2.11: Example of a convolution filter on the convolutional layer.

The resulting matrix is called an "Activation Map" or "Feature Map". The innovation of convolutional neural networks is to learn the filters during training which permits updating filter weights. In fact, the learning phase allows adjusting its weights in order to minimize a cost function.

Pooling layer:

Similar to the convolutional layer, the pooling layer is responsible for reducing the spatial size of the extracted activation map. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model. There are basically two types of pooling: max pooling and average pooling which operate locally among the feature map. As illustrated in Figure 2.12, max pooling returns the maximum value from the portion of the image covered by the kernel, whereas average pooling returns the average of all the values from the portion of the image covered by the kernel.

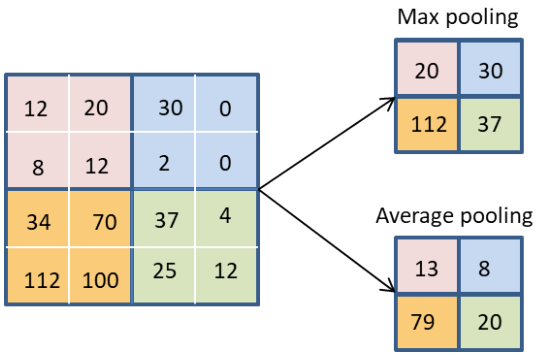


Figure 2.12: Example of max and average pooling.

The main purpose of a pooling layer is to apply a first-order pooling and reduce the number of parameters of the input tensor and thus:

- Helps reduce over-fitting and insures small invariance to translation.
- Extracts representative features from the input tensor.
- Reduces model complexity and computational time and thus aids efficiency.

Activation layer:

After each convolutional layer, a CNN applies an activation function that transforms each pixel value. Usually, a ReLU (Rectified Linear Unit) activation function is performed. For an input x, ReLU function simply turns all negative values into 0's (black) and keeps x for all values of $x > 0$. An activation function also introduces non-linearity into a model, and this means that the CNN will be able to find nonlinear boundaries that effectively separate data.

Fully-connected layer:

The input to the fully connected layer is the output from the final pooling or convolutional layer, which is flattened and then fed into the fully connected layer. As its name suggests, the neurons in this layer are connected to every neuron in the previous layer. After passing through the fully connected layers, the final layer uses the softmax activation function which is used to get probabilities of the input being in a particular class.

2.2.2.2 Transfer learning

Transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the target problem. With transfer learning, instead of starting the learning process from scratch, the significant part of the learned knowledge is transferred to the dataset of interest which reduces the computational cost.

Thus, the key motivation is the fact that most models which solve complex problems need a large set of data, and getting vast amounts of labeled data for supervised models can be really difficult, by considering the time and effort it takes to label data points. A simple example would be the ImageNet dataset [Russakovsky *et al.* 2014], which has millions of images and is commonly used to train convolutional neural networks.

From a practical perspective, the entire transfer learning process can be summarised as follows:

- Select a pre-trained model: A pre-trained model is chosen from available models that have been learned on large and challenging datasets. For image classification tasks, several models are available such as the AlexNet, CaffeNet or even deep models such as VGG models which were learned on the ImageNet dataset [Russakovsky *et al.* 2014]. This latter is the most widely used large scale dataset and contains more than 14 millions of labeled images. The pre-trained CNN models can be downloaded and incorporated directly into new models.

- Freeze the pre-trained model: The considered model is used as a feature extractor through the convolutional layers. In fact, the pre-trained model is kept at its original form and then its outputs are fed on a selected classifier. This option is usually adopted for problems with low computational power, and/or when dataset is small, and/or the pre-trained model solves a problem very similar to the task of interest.

- Fine-tune the pre-trained model: In this case, some layers of the pre-trained model are trained according to the dataset of interest. As shallow layers refer to basic features, such as horizontal, vertical, and diagonal edges, and deeper layers refer to specific features, such as corners and combinations of edges. The choose of frozen layers and retrained layers comes down to play with that dichotomy in order to improve the model and avoid overfitting.

In recent years, in order to benefit from both CNN architectures and encoding methods, many authors have focused on proposing hybrid architectures that consist of combining deep neural network architecture with traditional machine learning methods such as those based on FV/VLAD descriptors.

2.2.3 Hybrid architectures

A scene image is composed by a set of visual elements. For example, an harbour scene is formed by many objects such as boat, water, pontoon, etc. In this context, coding based methods such as Fisher vectors (FV) or Vector of Locally Aggregated (VLAD) descriptors have reached the state-of-the-art at the beginning of the 2000's [Perronnin & Dance 2007, Arandjelović & Zisserman 2013, Jégou *et al.* 2010]. More recently, the development of CNNs has had a tremendous influence in the field of computer vision and is responsible for a big jump in the ability to recognize objects. In fact, they have shown to outperform coding methods by a significant margin. For instance, on the ImageNet large scale visual recognition challenge, deep learning based methods have won since 2012 [Krizhevsky *et al.* 2012]. In order to benefit from both strategies, in the recent literature on scene classification, many authors have introduced hybrid architectures that combine CNN with some coding methods. For example, Perronnin et al. [Perronnin & Larlus 2015] have proposed a network of fully connected layers trained on the FV descriptors. Simonyan et al., introduced in [Simonyan *et al.* 2013] the Fisher network, which is composed of several stacked FV layers. Later, Arandjelovic et al. [Arandjelovic *et al.* 2015] proposed the NetVLAD layer, which mimicks the VLAD layer. To benefit of multi-layer representation, other strategies include the FV or VLAD encoding of CNN features from different layers of the network [Ng *et al.* 2015, Cimpoi *et al.* 2016, Diba *et al.* 2017, Li *et al.* 2017]. The next subsections briefly present these hybrid architectures.

2.2.3.1 Fisher vectors meet fully-connected layers network

Perronnin et al. proposed in [Perronnin & Larlus 2015] a first hybrid architecture allowing the combination of the best of both worlds, the leading architectures on machine learning based on Fisher vectors encoding and neural networks. For that, the introduced architecture, illustrated in Figure 2.13 is constituted of unsupervised layers involving the feature extraction, the computation and the dimensionality reduction of FVs. Then, instead of using a standard classifier, such as the SVM algorithm, a set of fully-connected neural network (NN) layers are plugged in the following. However, this architecture can not be trained on an end-to-end manner.

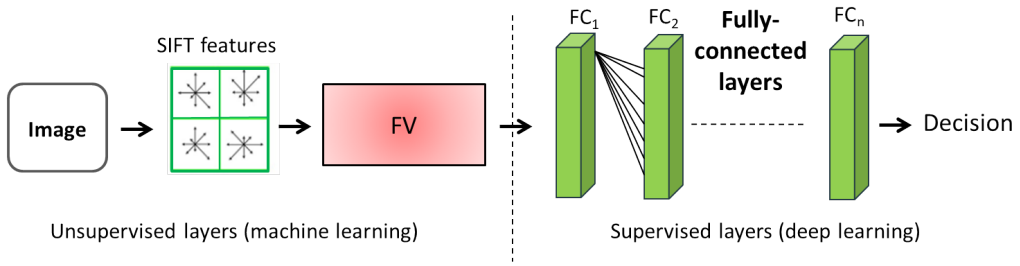


Figure 2.13: Fully-connected network model.

2.2.3.2 NetVLAD

The authors in [Arandjelovic *et al.* 2015] proposed a convolutional neural network (CNN) architecture that is trainable in an end-to-end manner directly for the place recognition task. The main component of this architecture is the NetVLAD as illustrated in Figure 2.14. It is a new generalized VLAD layer, inspired by the VLAD encoding method and is trained using the last convolutional feature of the CNN model.

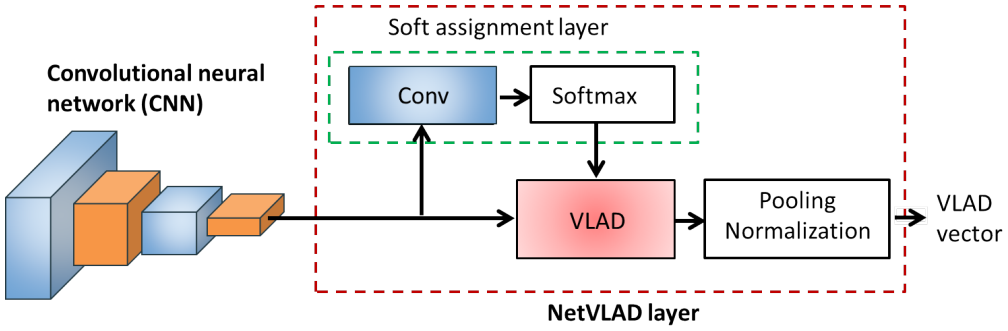


Figure 2.14: NetVLAD model architecture.

The NetVLAD layer permits to mimic a VLAD in a CNN framework and design a trainable generalized VLAD layer. By doing that, it results on a powerful image representation which has the ability to be trainable on an end-to-end manner regarding the target task. The specificity of the VLAD layer is that it is easily pluggable into any CNN architecture as it is amenable to back-propagation. To make operations differentiable and avoid discontinuities in the VLAD due to the use of hard assignment's of descriptors to cluster centers, the soft assignment layer is performed which allows to assign the weight of each descriptor to a specific cluster proportional to their proximity, but relative to proximities to other clusters. This comes down to replace the hard assignment introduced in (2.6) with a soft assignment described by:

$$\gamma_k(\mathbf{x}_i) = \frac{e^{\alpha||\mathbf{x}_i - c_k||^2}}{\sum_{k'} e^{\alpha||\mathbf{x}_i - c_{k'}||^2}}. \tag{2.16}$$

Note that $\gamma_k(\mathbf{x}_i)$ ranges between 0 and 1 and α is a positive parameter that controls the decay of the response where a very large α approximates to hard assignment of the original VLAD. The form of (2.16) can result on the form in (2.17) by expanding the squares and cancelling the term $e^{\alpha||\mathbf{x}_i||^2}$ between the numerator and the denominator. It gives:

$$\gamma_k(\mathbf{x}_i) = \frac{e^{w_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{w_{k'}^T \mathbf{x}_i + b_{k'}}}. \tag{2.17}$$

where $w_k = 2\alpha c_k$ and $b_k = -\alpha||c_k||^2$. Then the final form of the NetVLAD layer is given by the VLAD descriptor (2.6) weighted by the soft-assignment obtained in (2.17). It results in:

$$v_k = \sum_{i=1}^{N} \frac{e^{w_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{w_{k'}^T \mathbf{x}_i + b_{k'}}} (\mathbf{x}_i - c_k). \tag{2.18}$$

As a consequence, the NetVLAD layer has three independent sets of parameters for each cluster k (weight $\{w_k\}_{k=1,\dots,K}$, biases $\{b_k\}_{k=1,\dots,K}$ and cluster centers $\{c_k\}_{k=1,\dots,K}$) which enables greater flexibility than the original VLAD. As illustrated in Figure 2.14, the soft-assignment can be seen as a softmax function such as:

$$\gamma_k(\mathbf{x}_i) = \text{softmax}(w_k^T \mathbf{x}_i + b_k), \tag{2.19}$$

where $w_k^T \mathbf{x}_i + b_k$ is obtained as an output of a convolution with a set of 1×1 filters $\{w_k\}_{k=1,\dots,K}$ and biases $\{b_k\}_{k=1,\dots,K}$. At the end, a normalization step is performed and the obtained descriptor is of dimension $(K \times D) \times 1$.

In [Arandjelovic *et al.* 2015], the NetVLAD model significantly outperforms off-the-shelf CNN models and improves over the state-of-the-art on challenging image datasets, specially for place recognition benchmarks.

2.2.3.3 Fisher network

To take advantages of local features encoding and inspired by the multi-layer architecture of convolutional neural networks which allows to capture complex image structures, Simonyan et al. [Simonyan *et al.* 2013] defined the Fisher network which is constitued of stack of Fisher layers as illustrated in Figure 2.15.

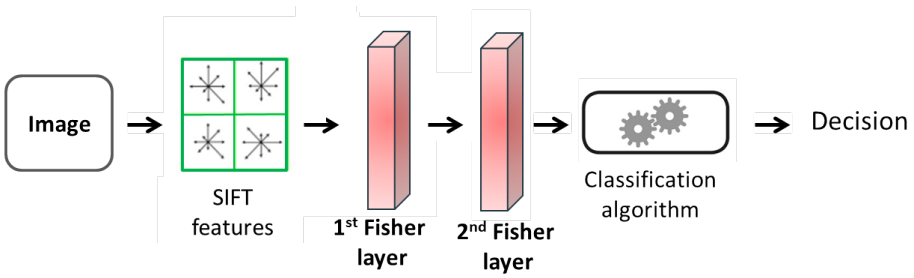


Figure 2.15: Fisher network architecture: Stack of Fisher layers on top of each other.

Such Fisher layers consists of the operations illustrated in Figure 2.16. On the input, the Fisher layer at each level receives the d-dimensional SIFT features after being decorrelated using PCA, densely computed over multiple scales on a regular image grid. Then, Fisher vector encoding is performed by pooling only a set of semi-local regions of the image. The idea is to

consider more than a unique FV vector to describe each image by considering semi-local FVs on square regions of the image where more complex image statistics are captured. For the FV encoding, a GMM generative model of K components is considered to produce a FV of dimension $2Kd$. To reduce FV dimensionality, a discriminatively trained linear projection is applied. The spatially adjacent features are stacked in a 2×2 window. Finally, to enhance invariance properties, reduce dimension and decorrelate, the produced features are L2-normalized and PCA-projected before being passed as the input to the following Fisher layer.

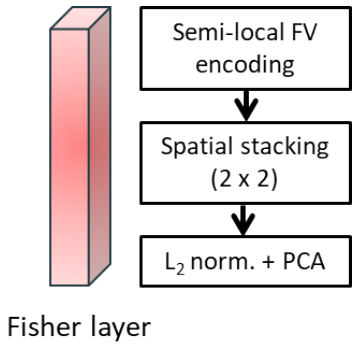


Figure 2.16: Architecture of a Fisher layer.

As illustrated, the Fisher network is constructed by stacking several Fisher layers on the basis of dense features, such as SIFT features. In analogy to the state-of-the-art deep neural network architecture, the Fisher network is trained in a supervised manner, since each Fisher layer depends on discriminative dimensionality reduction.

The hybrid architectures presented above provided significant improvement. However, they do not exploit second-order statistics, i.e., dependencies between features, which have been shown to be important in the human visual recognition process. Since then, we introduce in the following sections different models based on covariance pooling of CNN features to exploit second-order statistics.

2.3 Local covariance pooling: Ensemble log-Euclidean Fisher vector architecture

Building on the success of those latter hybrid architectures, more attention is given to a particular approach introduced in [Li *et al.* 2017]. In that paper, Li et al. have proposed a hybrid structure, which consists of encoding each output of the convolutional layers of a pre-trained neural network with FV. This technique has demonstrated competitive results for remote sensing scene classification. To capture various scale phenomenons when applying the FV encoding, a Gaussian pyramid is considered. This permits generating multiscale images by using a Gaussian smoothing and sub-sampling at different scales as detailed in [Li *et al.* 2017]. Classification results have demonstrated the interest of using multiscale images compared to a single input image. Therefore, a pyramid of three scale levels is retained in the following. Those multiscale images are fed into the CNN model, allowing the extraction of convolutional features which are then concatenated before being encoded with FV. Note that CNN models are used only to extract deep features without any retraining from scratch or fine-tuning. In fact, once the multiscale features are extracted from each convolutional layer, an individual codebook is generated. In this approach, the dimension K of the codebook is the same for all the layers. The

CNN features are then encoded with the improved FV [Perronnin *et al.* 2010b]. Next, those FVs are fused to represent the mid-level feature vectors of a scene image. Therefore, this approach does not consider second-order features, which have proved to be efficient in many classification problems and have shown to outperform first-order features for many image processing applications, including material recognition and person re-identification. To this aim, we have proposed in [Akodad *et al.* 2018b] a novel hybrid architecture named Hybrid LE FV, which integrates second-order features in the classification algorithm,

2.3.1 Hybrid log-Euclidean Fisher vector (Hybrid LE FV)

As illustrated in Figure 2.17, the hybrid LE FV architecture consists of a local covariance pooling where the log-Euclidean Fisher Vector (LE FV) encoding of the covariance matrices of CNN features are computed locally on layers output. The next Section 2.3.1.1 presents in details the principle of this Hybrid LE FV approach starting from the extraction of region covariance matrices to the FV encoding with the learned codebook [Akodad *et al.* 2018b]. Then, aiming at improving the classification performance, a proposition of an ensemble learning version of Hybrid LE FV strategy is detailed in Section 2.3.2.

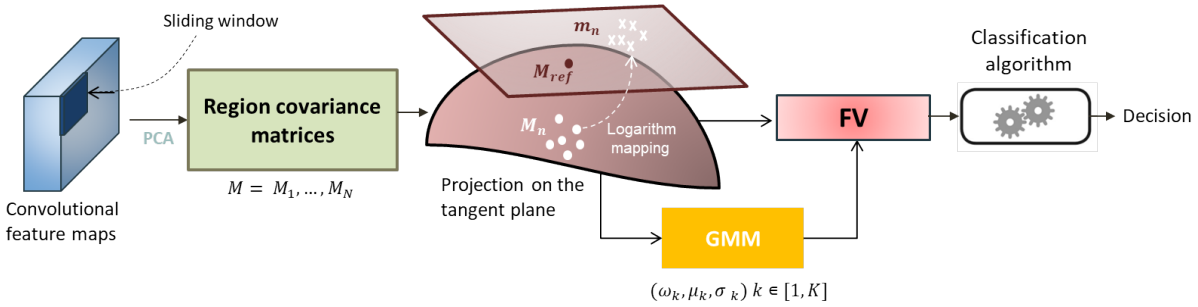


Figure 2.17: Principle of the proposed log-Euclidean Fisher vector encoding of region covariance matrices (Hybrid LE FV).

2.3.1.1 Region covariance matrices

The first step is to extract the region covariance matrices computed on a sliding window on the CNN feature maps. Hence, each image is represented by a set $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_n\}_{n=1:N}$ of covariance matrices $\mathbf{M}_n \in \mathcal{P}_d$. As the size of the output CNN layer depends on layer depth, only the first and second layers of a CNN are considered for computing local covariance matrices. Indeed, for the deepest layers, the feature maps are of small spatial dimension which does not allow the extraction of a large set of covariance matrices. For this purpose, a particular attention is given to the choice of the CNN model. Here, the CNN model adopted is a very deep convolutional network named VGG-16 [Simonyan & Zisserman 2014]. It is composed of 16 weight layers and is characterized by using a simple 3×3 convolutional layer stack with a stride fixed to 1 pixel and a spatial padding of 1 pixel. Therefore, the size of the output feature map is preserved through the first two layers that permit the extraction of a sufficient set of region covariance matrices. Then, according to the log-Euclidean framework detailed in section 1.3.3, these region covariance matrices are encoded with the LE FV. For that, a codebook is first learned by considering a Gaussian mixture model on the manifold of SPD matrices.

2.3.1.2 Gaussian mixture model and codebook creation

Referring to chapter 1, let's consider the following GMM model:

$$p(\mathbf{M}|\omega, \bar{\mathbf{M}}, \Sigma) = \sum_{k=1}^{K} \omega_k \; p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k), \tag{2.20}$$

where $p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k)$ is a multivariate Gaussian distribution defined on the tangent space of the identity matrix. In the context of log-Euclidean framework detailed in section 1.4.4 of chapter 1, the probability density function is given by:

$$p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k) = \frac{\exp\left\{-\frac{1}{2}\left(\text{Vec}\left(\log(\mathbf{M})\right) - \text{Vec}\left(\log(\bar{\mathbf{M}}_k)\right)\right)^T \Sigma_k^{-1} \left(\text{Vec}\left(\log(\mathbf{M})\right) - \text{Vec}\left(\log(\bar{\mathbf{M}}_k)\right)\right)\right\}}{(2\pi)^{\frac{d(d+1)}{4}} |\Sigma_k|^{1/2}}. \tag{2.21}$$

$\omega_k \in [0, 1]$, $\bar{\mathbf{M}}_k \in \mathcal{P}_d$ and $\Sigma_k \in \mathcal{P}_{\frac{d(d+1)}{2}}$ are respectively the weight, mean and covariance matrices for the k^{th} component of the GMM model. In addition, the classical assumption of diagonal covariance matrices Σ_k is made, i.e. $\sigma_k^2 = \text{diag}(\Sigma_k) \in \mathbb{R}^{\frac{d(d+1)}{2}}$ is the variance vector [Perronnin & Dance 2007].

Moreover, Equation (2.21) can be rewritten as:

$$p(\mathbf{M}|\bar{\mathbf{M}}_k, \Sigma_k) = p(\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}}|\mu_k, \Sigma_k) = \frac{\exp\{-\frac{1}{2}(\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} - \mu_k)^T \Sigma_k^{-1} (\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}} - \mu_k)\}}{(2\pi)^{\frac{d(d+1)}{4}} |\Sigma_k|^{1/2}}, \tag{2.22}$$

where $\mu_k = \text{Vec}\left(\log(\bar{\mathbf{M}}_k)\right) \in \mathbb{R}^{\frac{d(d+1)}{2}}$ is the log-Euclidean mean vector for the k^{th} component of the GMM model, and $\mathbf{m}^{\mathcal{T}_{\mathbf{I}_d}}$ is the LE vector representation of $\mathbf{M}$ given by Equation (1.31) and Equation (1.32). Since covariance matrices are projected into the tangent space and represented by their corresponding LE vectors, all the algorithms developed on a vector space can be used. In particular, the EM algorithm for parameter estimation of a GMM model is used to estimate the weights, means, and dispersions parameters. The set of these estimated parameters represents the codebook that will further be used to encode the set of region covariance matrices extracted from each image.

2.3.1.3 Log-Euclidean Fisher vector encoding

Considering $\boldsymbol{\mathcal{X}} = (\mathbf{m}_1^{\mathcal{T}_{\mathbf{I}_d}}, \mathbf{m}_2^{\mathcal{T}_{\mathbf{I}_d}}, \ldots, \mathbf{m}_N^{\mathcal{T}_{\mathbf{I}_d}})$ be a set of $d(d+1)/2$-dimensional log-Euclidean vectors extracted locally from the first convolutional layers of an image. The LE FV encoding consists of projecting these local vectors onto the codebook defined in the previous subsection. The LE FV descriptor assigned to $\boldsymbol{\mathcal{X}}$ is obtained by computing the gradient of the log-likelihood with respect to GMM model parameters, scaled by the inverse square root of the Fisher Information Matrix (FIM) $\mathbf{F}_\theta$ [Perronnin & Dance 2007]:

$$\mathscr{G}_\theta^{\boldsymbol{\mathcal{X}}} = \mathbf{F}_\theta^{-\frac{1}{2}} \nabla_\theta \log p(\boldsymbol{\mathcal{X}}|\theta). \tag{2.23}$$

Here, θ represents each of the distribution parameters (ω_k, μ_k and σ_k). In practice, the derivatives with respect to the mean $\mu_k(j)$ and standard deviation $\sigma_k(j)$ have been found to be the most useful [Perronnin & Dance 2007]. Hence, the following two FVs are obtained after

deriving with respect to these two elements

$$\mathscr{G}^{\boldsymbol{\mathcal{X}}}_{\mu_k(j)} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) \left(\frac{\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}(j) - \mu_k(j)}{\sigma_k(j)} \right), \tag{2.24}$$

$$\mathscr{G}^{\boldsymbol{\mathcal{X}}}_{\sigma_k(j)} = \frac{1}{\sqrt{2\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) \left(\frac{\left[\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}(j) - \mu_k(j)\right]^2}{\sigma_k^2(j)} - 1 \right), \tag{2.25}$$

where $\mu_k(j)$ (resp. $\sigma_k(j)$) is the j^{th} element of vector μ_k (resp. σ_k) and $\gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}})$ is the occupancy probability of $\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}$ to the k^{th} Gaussian component of the GMM, also named the posterior probability, and is defined as:

$$\gamma_k(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}}) = \frac{\omega_k \; p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \omega_j \; p(\mathbf{m}_n^{\mathcal{T}_{\mathbf{I}_d}} | \mu_j, \Sigma_j)}. \tag{2.26}$$

Once FV descriptors are obtained, a post-processing step is conventionally used to enhance the classification accuracy [Perronnin *et al.* 2010b, Sánchez *et al.* 2013]. This consists of a power and an ℓ_2 normalization. Furthermore, to avoid the curse of the dimensionality phenomenon when the dimensionality of the FV descriptor is high, a dimension reduction step can be used. In the following, the Kernel Discriminant Analysis (KDA) as suggested in [Mika *et al.* 1999]. Finally, a classification with a linear SVM is performed to make the decision for each test image depending on the information contained in the FV vector representation.

2.3.1.4 Dimensionality reduction

Working in high-dimensional spaces can be challenging for many reasons; raw data are often sparse as a consequence of the curse of dimensionality which degrade algorithms performance, and analyzing the data is usually computationally expensive. Machine learning problems that involve many features make training extremely slow. Dimensionality reduction aims at solving this issue. It can be defined as the process of transforming data from a high-dimensional space into a low-dimensional space while preserving some meaningful properties of the original data. More precisely, the reduction refers to techniques that reduce the number of feature variables in a dataset. There are many techniques that can be used for dimensionality reduction. The following diagram in Figure 2.18 categorizes some of the principal dimension reduction methods.

Feature selection aims at building a model of high accuracy by selecting the optimal features from the input dataset and leaving out the irrelevant features. Forward and backward selection methods [Weisberg 2005, Kutner 2005] are based on evaluation an machine learning model performance using dataset features. The features are kept or discarded, depending on the model accuracy. Furthermore, feature extraction methods aims at transforming the high-dimensional into space into a low-dimensional space. This category is divided in two families: linear and non-linear methods.

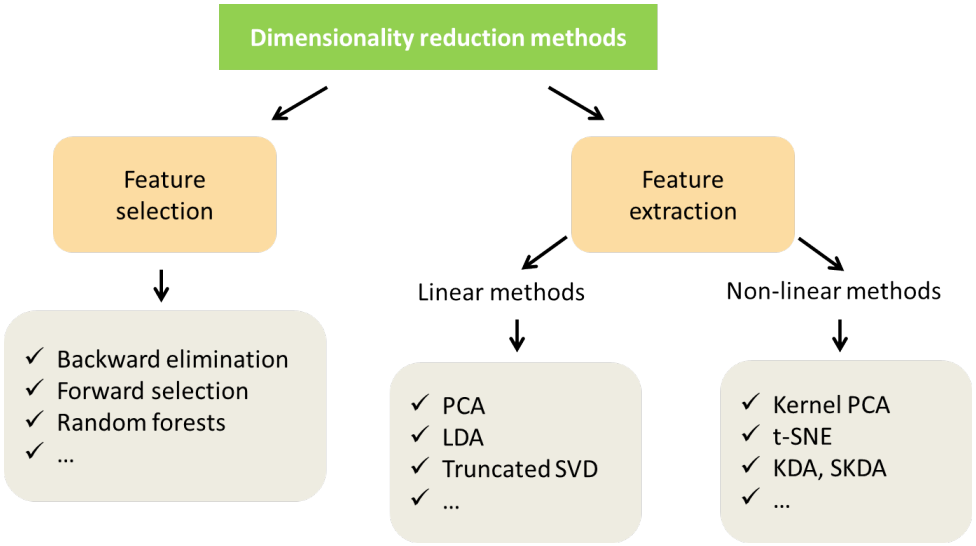


Figure 2.18: Chart summarizing some dimensionality reduction methods.

Linear methods involve linearly projecting the original data onto a low-dimensional space. The most common method is the principal component analysis (PCA). PCA proceeds in an unsupervised manner to transform a set of correlated variables d into a smaller ($k \leq d$) number of uncorrelated variables called principal components, while preserving as much as possible the variance of the original dataset. In contrast, the linear discriminant analysis (LDA) is a supervised method which attempts to find a feature subspace that maximizes class separability. For an empirical comparison between PCA and LDA techniques, the interested reader is referred to [Martinez & Kak 2001]. Singular Value Decomposition (SVD) is also used to decompose the original dataset into its constituents, resulting in dimensionality reduction [Halko *et al.* 2010]. Non-linear methods permit to use more advanced techniques for dimensionality reduction, such as the t-Distributed Stochastic Neighbor Embedding (t-SNE) [van der Maaten & Hinton 2008] algorithm which is one of the few algorithms capable of capturing both local and global structure of the data at the same time. Also, the Kernel Discriminant Analysis (KDA) [Cai *et al.* 2007] is an extension of the LDA based on the introduction of a kernel function which corresponds to the non-linear mapping.

For our work, PCA and LDA as well as its variant KDA are adopted for the aim of dimensionality reduction and feature decorrelation.

2.3.1.5 Sensitivity analysis

As explained in subsection 2.3.1.3, two parameters have to be tuned for the proposed Hybrid LE FV method, namely the number of components K in the GMM model and the dimension d of the covariance matrices. To evaluate the influence of each parameter on classification accuracy, some experiments are carried out on the UC Merced Land Use Land Cover dataset [Yang & Newsam 2010]. This dataset is composed of 21 classes where each class contains 100 remote sensing images of dimension 256 $\times$ 256 pixels. Figure 2.19 shows some examples of the UC Merced dataset. In order to prove the efficiency of the proposed approaches in challenging conditions, only a small set of $p = 10\%$ images is retrained for training for all experiments and the remaining images are used for testing. Classification results are evaluated in terms of overall accuracy averaged on five runs. Moreover, to allow a fair comparison between the models, the same images are used for training.

Figure 2.19: Samples from the UC Merced dataset.

Figure 2.20 draws the evolution of the classification accuracy of the proposed Hybrid LE FV approach for the first convolutional layer as a function of the dimension d of the covariance matrix. Here, the number of GMM components is fixed equal to 30. The dimension d is the number of selected principal components. If d is too small, a low number of principal components is retained. All the variability is not well explained, which leads to low classification accuracy. When d increases, more variability is explained, and the classification performance also increases. But after a certain value ($d = 5$ in our experiments), the variance gain is not so important and the classification performance remains quite stable. Hence, it is recommended to consider a covariance matrix size greater than a value of $d = 5$.

To evaluate the sensitivity of the proposed approach to the number of GMM components, Table 2.2 shows the classification accuracy using three values of K in the GMM model. As observed, the approach is not very sensitive to the codebook dimension.

Method	**K = 10**	**K = 30**	**K = 60**
Hybrid LE FV (conv 1)	60.5 ± 1.0 %	61.2 ± 0.8 %	61.2 ± 0.8 %

Table 2.2: Classification accuracy of Hybrid LE FV using three codebook dimensions K (d=5).

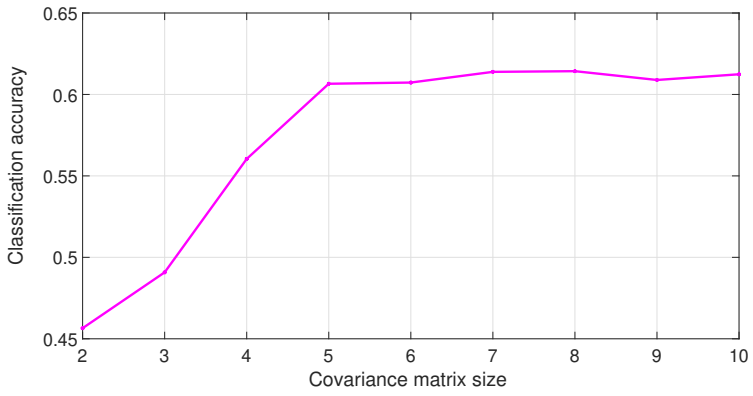


Figure 2.20: Influence of dimension d of covariance matrices for Hybrid LE FV (conv 1) on the UC Merced dataset (K=30).

2.3.2 Ensemble hybrid log-Euclidean Fisher vector (Ens. Hybrid LE FV)

In machine learning, ensemble learning strategies have become more and more popular [Breiman 2001, Kuncheva & Whitaker 2003]. They rely on the combination of multiple weak classifiers to form a stronger one, hence allowing improvements to the classification performance. The most basic approaches are boosting and bagging [Freund & Schapire 1996, Breiman 2005]. Boosting is based on an iterative re-training process where the weakness is identified, related to the misclassified samples, and is considered to focus on the learning process. In contrast, the bagging strategy, also called bootstrap aggregation, consists in randomly generating samples with replacement, of a fixed size, called bootstrap samples, from an initial dataset. Those samples are used to fit several independent models in order to obtain a final model with a lower variance. As an example, random forest algorithm is an ensemble learning method that seeks to train each model, through decision trees, on a different sample of the same training dataset. The predictions made by the ensemble members are then combined together to elect the final decision using simple statistics, such as voting for classification or averaging for regression problems. The diversity in the ensemble is actually ensured by the variations within the data in which each decision tree is trained on.

2.3.2.1 Proposed architecture

Inspired by this idea, we introduce an ensemble learning approach for the hybrid log-Euclidean Fisher vector presented in the previous subsection. The workflow of this method named “Ens. Hybrid LE FV", is shown in Figure 2.21. As observed, for each convolutional layer (conv 1 and/or conv 2), N' subsets are considered. For each subset, d feature maps are randomly selected without replacement[1]. Actually, some conditions are necessary to ensure a good decision making, in particular, the following criteria:

- Decision independence of considered classifiers
- Diversity of features in each classifier

As detailed in [Surowiecki 2004], diversity of opinions bring meaningful differences rather than minor variations, as such, chances that decisions end up to the good decision increase. Also, diversity helps to preserve the independence, where each classifier acts of its own motion without any effect of other classifiers. To satisfy those conditions, there is a trade-off when choosing the number of features d for each subset. The size d of the initial subset should be small enough to ensure the diversity from one subset to another and decreases correlation to make classifiers act independently. Also, the covariance matrices are well estimated. Therefore, the classifier looses its stability due to the poor explanatory power of considered d feature maps. In contrast, a high value of d improves the features explanatory and the model stability, whereas it results on a poor diversity between classifiers, and a loss of independence where features have a higher probability to figure in each subset. In addition, it leads to a not well-conditioned covariance matrix.

[1]Replacement means that if a feature is selected, it is returned to the training dataset for potential re-selection in the same training dataset.

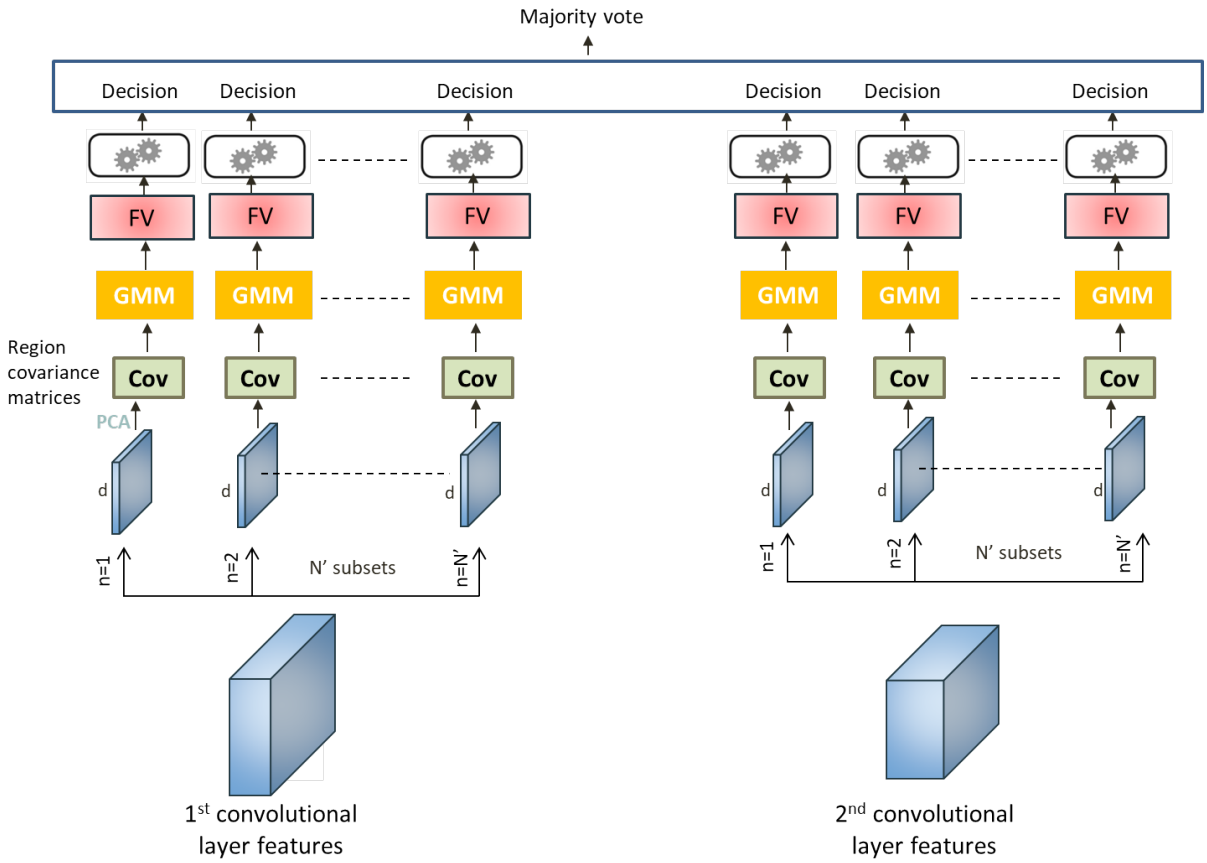


Figure 2.21: Ensemble Hybrid LE FV workflow.

A first experiment is conducted in order to evaluate the sensitivity of the proposed approach. This consists of evaluating the influence of the number of subsets N'. Table 2.3 shows the classification accuracy of the "Ens. Hybrid LE FV strategy regarding the first convolutional layer of VGG-16 model. Five values of N' are experimented (5, 7, 9, 11, and 13) for $d = 5$ and $p = 10\%$ of training images of the UC Merced dataset.

N'	Ens. Hybrid LE FV
5	63.7 ± 0.6 %
7	64.0 ± 0.3 %
9	64.0 ± 0.3 %
11	63.9 ± 0.1 %
13	64.0 ± 0.5 %

Table 2.3: Classification accuracy of "Ens. Hybrid LE FV" using different number of subsets N'

One can observe that results remain quite stable for the different considered N'. For further experiments, the number of subsets N' will be fixed to 7.

2.3.2.2 Results for GMM modeling involving a unique tangent space

Table 2.4 highlights the classification results obtained on the UC Merced dataset for the first (conv 1) and second (conv 2) convolutional layers of VGG-16 network. The proposed ensemble learning approach, "Ens. Hybrid LE FV", is compared to two closely related state-of-the-art strategies. The first one, named "Hybrid FV", consists of encoding the output of the

convolutional layers with FV [Li *et al.* 2017]. Note that this approach considers only first-order statistics. The second one, named "Hybrid LE FV" is the one presented in Section 2.3.1.1. It exploits second-order statistics but not in an ensemble learning approach [Akodad *et al.* 2018b].

Method	Conv 1	Conv 2
Hybrid FV [Li *et al.* 2017]	41.4 ± 0.2 %	43.7 ± 1.1 %
Hybrid LE FV [Akodad *et al.* 2018b]	61.2 ± 0.8 %	65.1 ± 1.6 %
Ens. Hybrid LE FV	**62.4 ± 0.9 %**	**68.1 ± 1.7 %**

Table 2.4: Classification results on the UC Merced dataset for the first and second convolutional layers of the VGG-16 network ($p = 10\%$).

As observed in Table 2.4, the benefit of exploiting second-order statistics is clearly demonstrated for the first and second CNN convolutional layers. A significant gain of 20% to 25% is reported for the proposed "Hybrid LE FV and Ens. Hybrid LE FV methods compared to the conventional Hybrid FV approach. In addition, for these first two layers, a significant gain is observed when exploiting an ensemble learning strategy compared to the use of a single classifier.

2.3.2.3 Results for GMM modeling involving multiple tangent spaces

In chapter 1, we have discussed in section 1.3.3 the choice of reference point for constructing the tangent plane and have proposed to extend the GMM modeling to multiple tangent planes defined at different reference points. As explained, this would be a better alternative to preserve the specific geometry of the SPD matrices and limit distortion when projecting covariance matrices at the same tangent plane, especially when the set of covariance matrices is located far from the considered reference point. For that, in order to define a reference point close to the covariance matrices, the centroid $\bar{\mathbf{M}}_k$ of each GMM cluster k has been estimated and a tangent space is defined for each GMM cluster at the estimated centroid.

As detailed in chapter 1, experimental results on synthetic data demonstrated the benefit of using adapted tangent planes to preserve the correct fitting of Gaussian models and avoid projection distortions, especially when data are located far from the reference point. Here, the objective is to extend the proposed architectures, Hybrid LE FV and Ens. Hybrid LE FV, to a GMM modeling with multiple tangent planes for a classification problem. For that, the EM algorithm is used to learn the codebook on the training set, and this latter one is used to encode a set $\boldsymbol{\mathcal{M}} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ of N covariance matrices with Fisher vectors. By combining (2.23) and (1.66), the FV associated to a GMM model with K reference points are obtained as:

$$\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{\bar{\mathbf{M}}_k(j)} = \frac{1}{\sqrt{\omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{\mathbf{z}^{\mathcal{T}_{\mathbf{I}_d}}_{(n,k)}(j)}{\eta_k(j)} \right), \tag{2.27}$$

$$\mathscr{G}^{\boldsymbol{\mathcal{M}}}_{\eta_k(j)} = \frac{1}{\sqrt{2\ \omega_k}} \sum_{n=1}^{N} \gamma_k(\mathbf{M}_n) \left(\frac{\left[\mathbf{z}^{\mathcal{T}_{\mathbf{I}_d}}_{(n,k)}(j)\right]^2}{\eta^2_k(j)} - 1 \right), \tag{2.28}$$

where $\mathbf{z}^{\mathcal{T}_{\mathbf{I}_d}}_{(n,k)}$ is the vector version of $\mathbf{Z}_{(n,k)}$ computed at the identity matrix $\mathbf{I_d}$. As detailed in chapter 1, Section 1.4.6.3, $\mathbf{Z}_{(n,k)}$ is the result of parallel transport of covariance matrices $\mathbf{M}_n$ to the identity matrix $\mathbf{I_d}$ and $\eta^2_k(j)$ is the variance of the transported set. Table 2.5 highlights the

classification results obtained on a single convolutional layer of the VGG-16 model, for instance the second layer (conv 2). In the proposed approaches, each layer is represented by a set of region covariance matrices which are further encoded with Hybrid LE $FV_{\mathcal{T}_{\mathbf{I}_d}}$ when using a single tangent plane at the identity matrix whereas for $FV_{\mathcal{T}_{\tilde{\mathbf{M}}_k}}$, multiple tangent planes are considered.

Method	Conv 2
Hybrid LE $FV_{\mathcal{T}_{\mathbf{I}_d}}$ (VGG-16)	**65.1 ± 1.6 %**
Hybrid LE $FV_{\mathcal{T}_{\tilde{\mathbf{M}}_k}}$ (VGG-16)	**66.7 ± 0.9 %**

Table 2.5: Classification results on the UC Merced dataset for the first and second convolutional layers of the VGG-16 model ($p = 10\%$).

As observed, since the two strategies are comparable, results are quite similar. To explain that, one hypothesis may be the fact that the induced distortion does not affect the classification results. In fact, covariance matrix features extracted from the UC Merced dataset are relatively discriminant and provide a sufficient separation between classes regardless the distorted data shape. As such, the encoding step between the two models, whether using a single tangent plane or multiple planes, gives similar results. Furthermore, as using multiple tangent planes requires high computational time, it is preferred to use the single tangent plane approach for the following.

In this approach, only covariance matrices computed on the first layers of a CNN have been encoded with the LE FV. Indeed, as the deepest convolutional layers of the VGG-16 network are of relatively small spatial dimensions, it is irrelevant to compute a sufficient number of region covariance matrices. Nevertheless, the deepest layers may provide useful features for the classification. To alleviate this issue, instead of considering a local approach, the covariance matrix will be computed globally for the deepest feature maps. For that, Section 2.4 introduces first a multilayer stacked covariance pooling approach, namely MSCP, proposed by He *et al.* in [He *et al.* 2018] then our extension to an ensemble learning approach based on a global covariance pooling of CNN features [Akodad *et al.* 2019c].

2.4 Global covariance pooling: Ensemble learning based on covariance pooling of CNN features (ELCP)

2.4.1 Multilayer stacked covariance pooling (MSCP)

Willing to exploit second-order statistics on deep convolutional layers of a CNN, He et al. have proposed in [He *et al.* 2018] a strategy named multilayer stacked covariance pooling (MSCP). The originality lies in the replacement of the usual first-order pooling (i.e. average or max pooling) in a CNN by a second-order pooling (i.e. covariance pooling). Note also that, in contrast with the ensemble hybrid LE FV method introduced in section 2.3.2, where each layer is represented by a set of covariance matrices computed locally on the feature maps, a single covariance matrix is computed for MSCP, which can significantly speed up the computation time.

The general principle of MSCP is summarized in Figure 2.22.

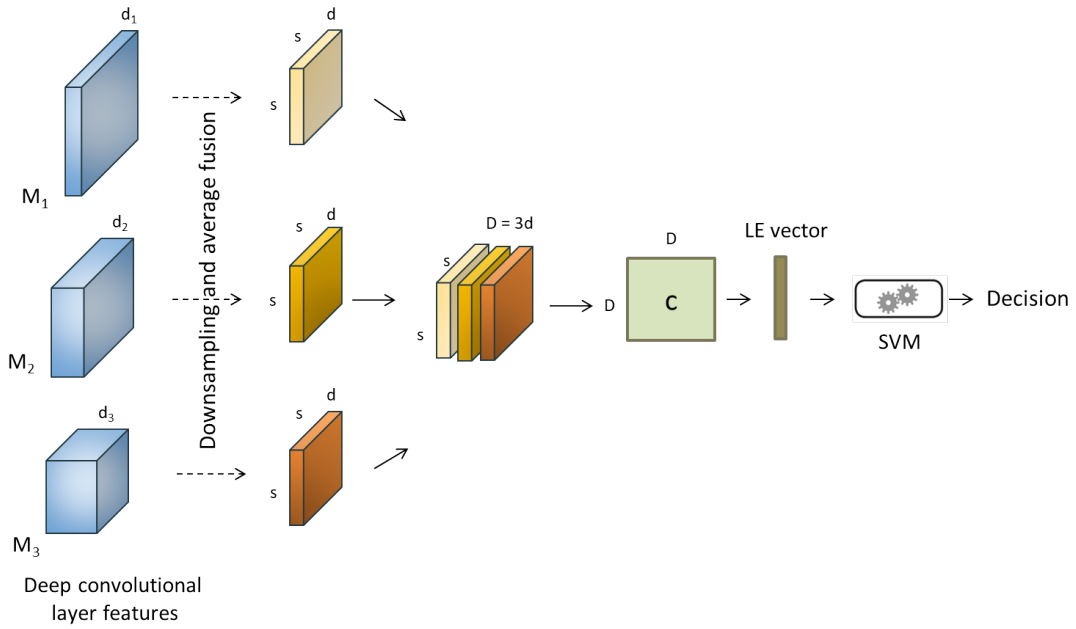


Figure 2.22: Architecture of the multilayer stacked covariance pooling strategy (MSCP).

First, three convolutional layers, with different depth, are considered and analyzed separately. For a given convolutional layer, an ensemble learning approach is considered by splitting the convolutional features into d subsets. For each subset, k features are selected without replacement. These latter are next downsampled and averaged in order to obtain only one descriptor by subset (see Figure 2.23). Then, the d average descriptors for each convolutional layer are concatenated allowing to obtain a tensor of dimension $s \times s \times 3d$. The covariance pooling operator is next applied, it consists in computing the $3d \times 3d$ covariance matrix descriptor. Finally, the log-Euclidean metric is considered for classification. For that, the LE vector representation is computed and an SVM classifier is adopted. MSCP has successfully been validated for remote sensing scene classification, but it suffers from two main drawbacks.

First, a single decision is obtained at the end. Second, the main drawback of MSCP concerns the averaging operator presented in Figure 2.23. In fact, it may lead to a not well-conditioned covariance matrices: there is no practical reason that the average descriptor obtained on one subset is different from the one calculated on another subset.

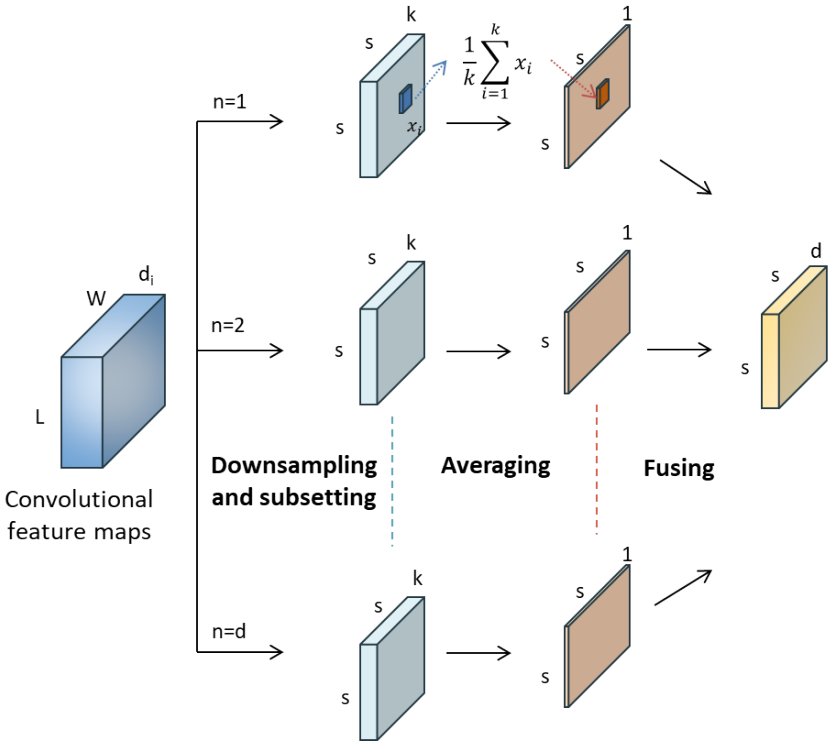


Figure 2.23: Description of the downsampling and averaging fusion operations over convolutional feature maps in the MSCP algorithm.

To overcome these problems, we have introduced in [Akodad *et al.* 2019c] a novel hybrid approach named ELCP, which consists of an ensemble learning approach based on covariance pooling of CNN features. This architecture is detailed in the next subsection.

2.4.2 Ensemble learning approach based on covariance pooling (ELCP)

The global principle of the proposed architecture [Akodad *et al.* 2019c] is shown in Figure 2.24. First, the feature maps M_1, M_2 and M_3 produced by three deep convolutional layers ($conv_{3-3}$, $conv_{4-3}$ and $conv_{5-3}$) are considered. Commonly, CNN layers have different spatial dimensions. For example, for the VGG-16 model, dimensions are $M_1 \in \mathbb{R}^{56\times56\times256}$, $M_2 \in \mathbb{R}^{28\times28\times512}$ and $M_3 \in \mathbb{R}^{14\times14\times512}$. A downsampling to the smallest spatial dimension is performed using a bilinear interpolation to stack the feature maps of these latter layers. Furthermore, for each image, an ensemble learning approach is considered where the stacked feature maps generated by the convolutional layers are split into N subsets of k features each. This splitting is achieved for each subset by random sampling without replacement. Then, for each subset n, a global covariance pooling strategy is adopted. It consists in computing the $k \times k$ covariance matrix $\mathbf{C}_n$. The log-Euclidean framework presented in chapter 1 is then adopted to represent $\mathbf{C}_n$ in the tangent plane of the identity matrix by $\mathbf{c}_n^{\mathcal{T}_{\mathbf{I}_d}}$ according to equation (1.30). Then, for each subset, these log-Euclidean vectors are fed to a base linear SVM classifier allowing them to obtain a decision. The final prediction is obtained as the most represented decision among the N subsets.

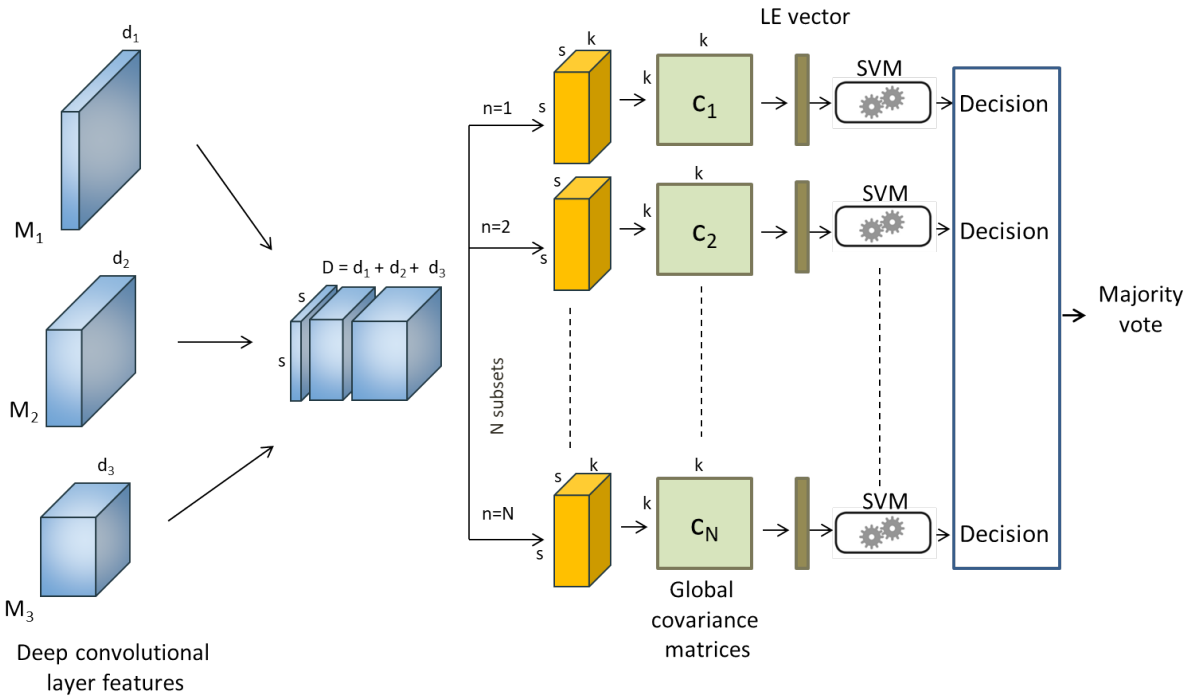


Figure 2.24: Ensemble learning approach based on covariance pooling of CNN features (ELCP) workflow.

As explained, two parameters should be tuned: the number N of subsets and the number k of selected features in each subset. In order to evaluate the sensitivity of the proposed ELCP approach, an experiment is conducted. Figure 2.25 draws the evolution of the classification accuracy as a function of k for different values of N ($N = 10$ to $N = 30$).

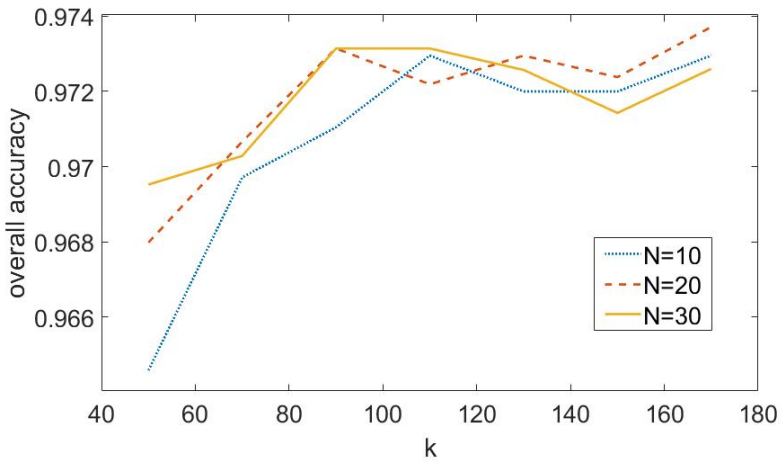


Figure 2.25: Influence of the number N of subsets and the number k of selected features in each subset on the classification accuracy.

As observed, the classification results for this method are stable and not so sensitive to parameter tuning, the number of subsets N and the number of feature maps k per subset retained in the following will, respectively be equal to 20 and 170 as suggested on [Akodad *et al.* 2019c]. These parameters are set to these values in the following.

2.4.3 Experimental results

2.4.3.1 Classification results for a single convolutional layer

This section introduces an application to large scale remote sensing scene classification. For that, the UC Merced Land Use is considered to evaluate the performance of the proposed supervised classification algorithm. Images are randomly separated into training and testing sets according to a fixed rate. 10 % of images are retained for training. In the following, two CNN models pretrained on ImageNet are considered: AlexNet [Krizhevsky *et al.* 2012] and VGG-16 [Simonyan & Zisserman 2014]. Note also that the final classification step in Figure 2.24 is performed by the linear SVM classifier.

The proposed ELCP approach is tested when CNN feature maps are issued from a single layer. Some comparisons are carried out with two other strategies: (1) an hybrid architecture based on the FV encoding of CNN features (Hybrid FV) [Li *et al.* 2017] and (2) the MSCP algorithm [He *et al.* 2018] detailed in Section 2.4.1. Table 2.6 summarizes the classification results obtained on the UC Merced dataset for three convolutional layers for AlexNet and VGG-16 CNN models.

	AlexNet		
	$Conv_3$	$Conv_4$	$Conv_5$
Hybrid FV [Li *et al.* 2017]	77.2 ± 0.5 %	79.9 ± 0.7 %	81.0 ± 1.2 %
MSCP [He *et al.* 2018]	79.3 ± 0.8 %	82.8 ± 1.2 %	80.6 ± 0.7 %
ELCP	**81.7 ± 1.1 %**	**83.8 ± 1.4 %**	**83.6 ± 1.4 %**
	VGG-16		
	$Conv_{3,3}$	$Conv_{4,3}$	$Conv_{5,3}$
Hybrid FV [Li *et al.* 2017]	73.5 ± 2.3 %	85.0 ± 0.7 %	86.5 ± 0.5 %
MSCP [He *et al.* 2018]	73.5 ± 1.2 %	**86.2 ± 1.3 %**	84.4 ± 1.0 %
ELCP	**76.7 ± 1.1 %**	**86.7 ± 1.1 %**	**88.8 ± 1.1 %**

Table 2.6: Classification performance obtained on UC Merced dataset using Hybrid FV, MSCP and the proposed ELCP approaches.

As observed in Table 2.6, the proposed ELCP architecture allows to improve the classification accuracy compared to Hybrid FV and MSCP architectures when a single layer is considered. A mean average gain of about 1.5 % and 2.6 % are respectively observed for AlexNet and VGG-16 models. Note also that the best results are obtained for the VGG-16 model. In the following, only this CNN model will be considered.

2.4.3.2 Classification results for multilayer features

Now that the proposed ELCP approach has successfully been validated for a single layer, the potential of a multilayer version is investigated. The proposed ELCP approach is compared with some standard and recent state-of-the-art approaches on the UC Merced dataset. A first approach is the FV encoding of handcrafted SIFT features (FV SIFT) [Perronnin *et al.* 2010b]. The next approaches are transfer learning methods based on the VGG-16 pre-trained CNN model on the ImageNet dataset. A fine-tuning of this model is first considered (CNN (VGG-16 fine-tuned)). For that, the convolutional layers are frozen, and a fully connected layer is added

and trained on the UC Merced dataset. The second transfer learning approach (VGG-16 feat. extraction + SVM) consists in considering the CNN model as a feature extractor. CNN features are then fed to an SVM classifier. Finally, the two second-order based methods, namely MSCP and the proposed ELCP approaches, are compared. Table 2.7 summarizes the classification results obtained for these five methods.

Method	OA (Mean ± sd)
FV (SIFT) [Perronnin *et al.* 2010b]	62.3 ± 1.1 %
CNN (VGG-16 fine-tuned)	62.7 ± 1.8 %
CNN (VGG-16 feat. extraction + SVM) [Chatfield *et al.* 2014]	82.7 ± 0.6 %
MSCP (VGG-16) [He *et al.* 2018]	86.3 ± 1.0 %
ELCP (VGG-16)	**88.4 ± 1.4 %**

Table 2.7: Classification performance of the proposed multi-layer architecture compared to some state-of-the-art algorithms on the UC Merced dataset ($p = 10\%$).

As observed in Table 2.7, several conclusions can be drawn. First, deep learning-based methods outperform traditional handcrafted based ones. Second, since a low number of samples is used for training in this experiment, a fine-tuning strategy does not provide the best results. It is better to consider a pre-trained CNN model as a feature extractor [Pires de Lima & Marfurt 2019, Cheng *et al.* 2020]. A gain of more than 20% is observed between these strategies. Third, among the transfer learning strategies based on feature extraction, methods exploiting second-order statistics of CNN features (MSCP and ELCP) outperform the first-order one. Fourth, by exploiting an ensemble strategy, the proposed ELCP significantly outperforms MSCP. A gain of about 2% is observed.

2.4.4 Ensemble learning covariance pooling guided by saliency maps (EL-SCP)

Visual saliency has been investigated in the computer vision literature in many different tasks, such as image classification and retrieval, semantic segmentation and object recognition [Moosmann *et al.* 2006, Gao & Vasconcelos 2004]. It permits to identify parts of the input image which are the most important for classification.

In the ELCP approach, all the pixels of the image contribute equally during the estimation of the sample covariance matrix. This might be problematic when the objects of interest are of small dimension compared to the surrounding environment. In order to give more strength to those elements, several approaches can be considered, especially by exploiting saliency maps. The simplest one consists in using the ELCP approach where the input image is multiplied by the saliency map.

But this approach loses the contextual information. To circumvent this drawback, we propose to exploit the saliency map on the CNN features. For that, we introduce a weighted covariance matrix estimator where the weights depend on the visual saliency. Larger weights will be given to more salient regions. In practice, each branch related to each subset in the ELCP approach shown in Figure 2.24 is replaced by the one shown in Figure 2.26 to form the proposed EL-SCP method.

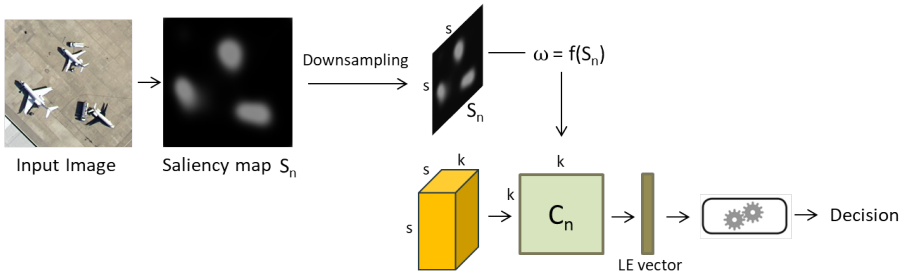


Figure 2.26: Covariance pooling of CNN feature bloc guided by saliency maps.

2.4.4.1 Saliency maps

Visual saliency describes the spatial locations in an image that attract the human attention. According to a bottom-up process, it consists in an exploration of the image by a human observer during a small duration and without any prior about its content. The progress in convolutional neural networks opened a possibility to leverage a new family of models to extract saliency maps where Pan *et al.*, inspired by generative adversarial networks (GANs), introduced in [Pan *et al.* 2017] the SalGAN architecture to estimate the saliency map of an input image. SalGAN, as illustrated in Figure 2.27, is composed of two competing convolutional neural networks: a generator which allows the generation of saliency maps using a convolutional encoder-decoder architecture and a discriminator which tells whether the generated saliency map is real or fake.

The training of SalGAN is made through two competing convolutional neural networks:

- **Generator:**
  Permits the generation of saliency maps using a convolutional encoder-decoder architecture. The encoder part is similar to a pretrained VGG-16 network. Filter weights of the last layers are modified and trained for saliency prediction. The decoder is structured in a reversed order than the encoder. The final output of the generator is a saliency map having the same size as the input image. The values are normalized such that each pixel is in range $[0, 1]$.

- **Discriminator:**
  The convolutional network is constituted by kernel convolutions, pooling, fully connected layers and activation functions. It produces an output score which tells whether the input saliency map is real or fake. shown in the Figure 2.27 this network is initially trained with a binary cross entropy (BCE) loss over the saliency maps.

In this work, the SalGAN strategy is employed to produce the saliency maps of the considered image datasets. In fact, only the trained generative part of the architecture is considered for prediction. Figure 2.28 illustrates an example of saliency map obtained with SalGAN for an image belonging to UC Merced airplane class. As expected, the most salient regions correspond to the three airplanes.

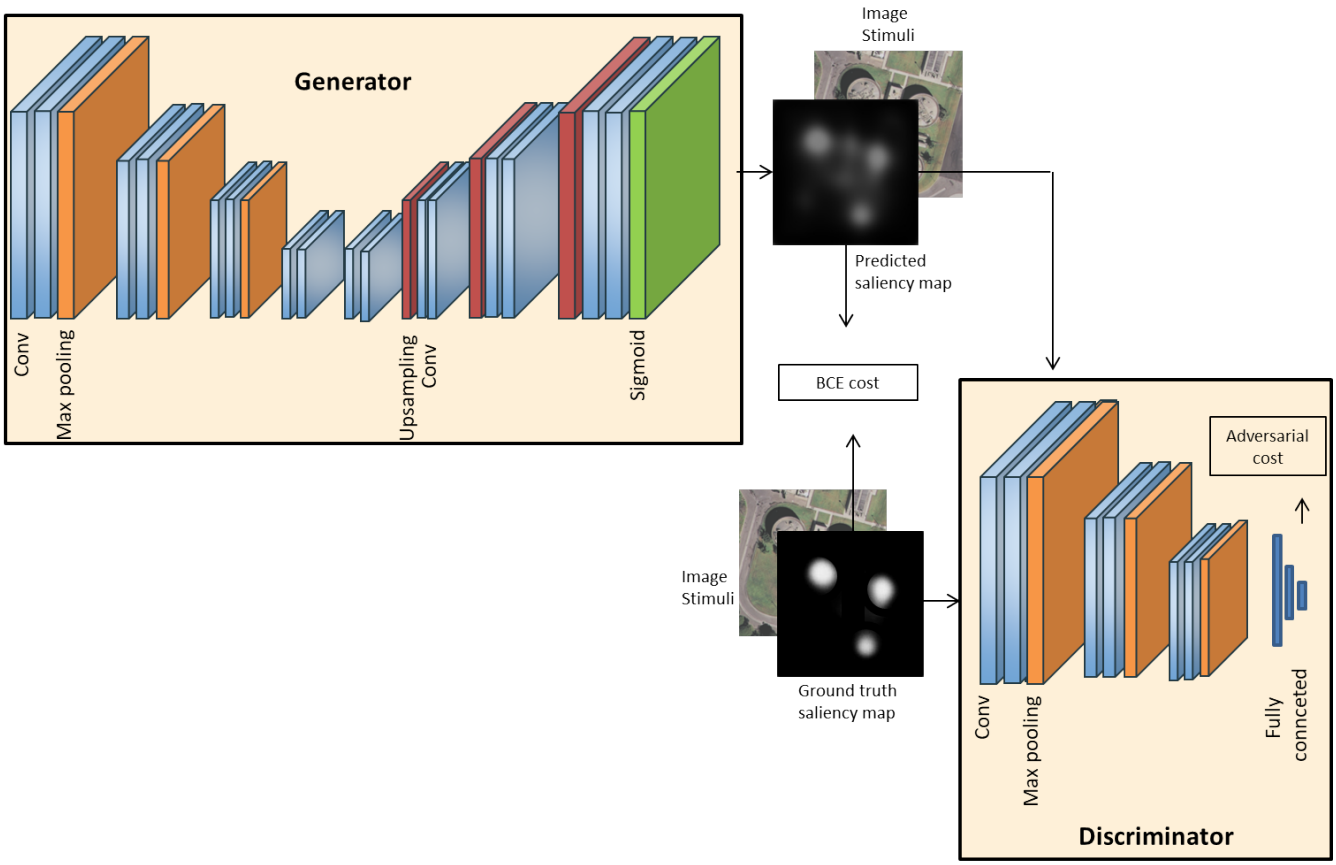
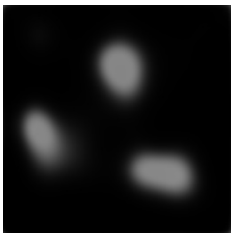


Figure 2.27: Overall architecture of the SalGAN network involving a generator and a discriminator for producing a saliency map.

(a) Input image



(b) Corresponding saliency map

Figure 2.28: Example of saliency map obtained with SalGAN for an image belonging to airplane class.

2.4.4.2 Weighted covariance matrix estimator

Inspired by the theory of robust statistics [Maronna *et al.* 2006], we propose to consider a weighted covariance matrix estimator during the covariance pooling step. The idea is similar to the fixed point algorithm explained in the previous chapter to enhance robustness of covariance estimation. For a given set of k dimensional CNN features $\{\mathbf{x}_i\}_{i=1...N}$, the $k \times k$ weighted covariance matrix is:

$$\mathbf{C} = \sum_{i=1}^{N} \omega_i (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T, \tag{2.29}$$

where μ is the weighted mean vector, *i.e.* $\mu = \sum_{i=1}^{N} \omega_i \mathbf{x}_i$, and ω_i is the weight assigned to pixel i and N is the number of pixels. In order to give more strength to salient regions, we propose

to define weights by:

$$\omega_i = \frac{\exp\left(\frac{s_i}{\sigma}\right)}{\sum_{j=1}^{N} \exp\left(\frac{s_j}{\sigma}\right)}, \tag{2.30}$$

where s_i is the saliency information obtained by SalGAN for pixel i and σ is a positive scalar parameter which controls the importance given to the saliency information. Note that when σ tends toward infinity, the weights ω_i are equal to $\frac{1}{N}$ and (2.29) reduces to the sample covariance matrix. Hence, in this case, the proposed EL-SCP approach reduces to ELCP [Akodad *et al.* 2019c].

2.4.4.3 Classification results

This section summarizes some classification experiments on large scale scene remote sensing images on the UC Merced land use land cover dataset [Yang & Newsam 2010]. Figure 2.29 draws the evolution of overall accuracy as a function of σ in (2.30). As expected, when σ tends toward infinity, the proposed EL-SCP approach is equivalent to ELCP [Akodad *et al.* 2019c]. Note that the best results are obtained for $\sigma = 50$. This value will be retained in the following.

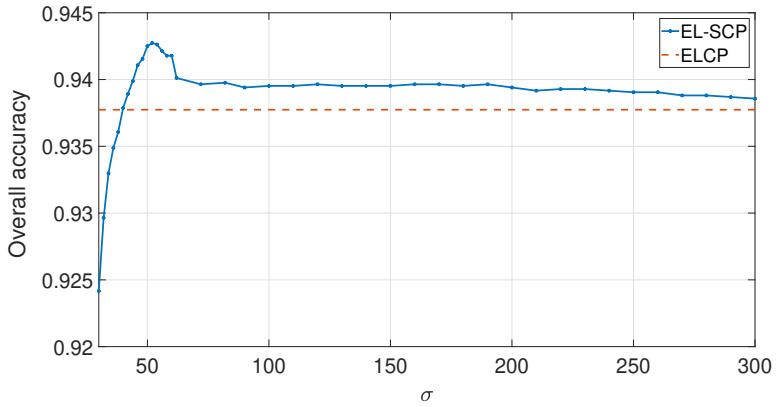


Figure 2.29: Influence of σ parameter for 20 % of training images.

Figure 2.30 shows the evolution of the classification performance with the percentage of training samples. Five benchmark approaches are considered. The first one, named VGG-16, consists of a simple transfer learning approach where features are extracted from VGG-16 model and classified with an SVM. The four other approaches are based on second-order features, namely MSCP [He *et al.* 2018], ELCP [Akodad *et al.* 2019c], Hybrid LE FV [Akodad *et al.* 2018b] and the proposed EL-SCP. As observed, EL-SCP allows to obtain competitive results compared to these state-of-the-art methods.

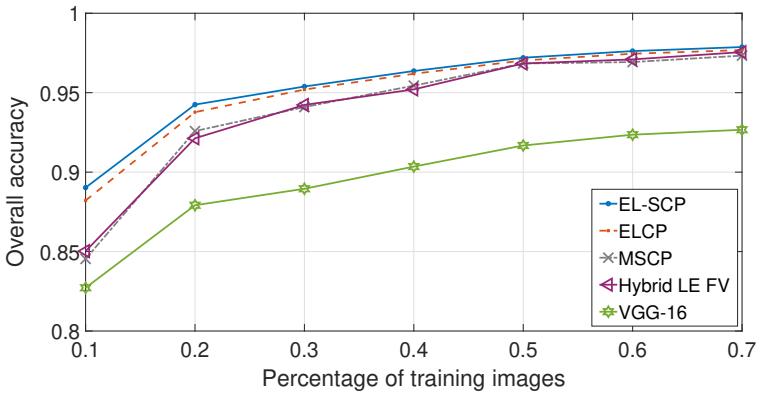
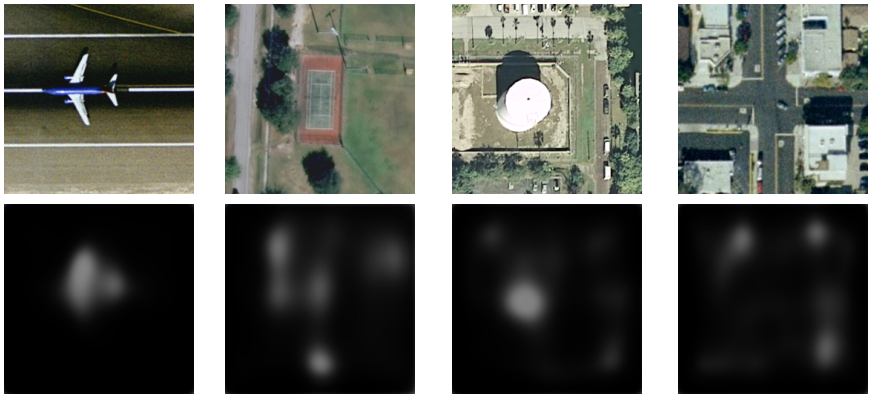


Figure 2.30: Influence of the percentage of training sample on the classification accuracy.

In order to analyze the pros and cons of the proposed EL-SCP approach over ELCP and MSCP, Table 2.8 shows classification performance per class on the UC Merced dataset where only 10 % of images are used for training. As it can be seen, EL-SCP performs better for most of the classes. Figure 2.31 shows some examples of images (with their corresponding saliency map). The three first images, belonging to (a) airplane, (b) tennis court and (c) storage tanks classes, are correctly classified only by EL-SCP, whereas the ELCP assigned them to runway, golf course and intersection classes, respectively. As observed, these images contain object of interest that are well captured by the saliency map. The proposed weighted covariance matrix estimator in EL-SCP allows hence to focus more on them, while for the ELCP, all pixels of the image contribute equally, the decision is therefore made in the basis of the whole image information and hence disregarding under-represented objects. In contrast, for very few images, there are some cases where the saliency map can be misguiding. This is the case of Figure 2.31(d) which belongs to intersection class. Here, the saliency does not focus on the intersection but on the surrounding area. This image is hence assigned to the buildings class for EL-SCP whereas it is correctly classified by ELCP.

Class	EL-SCP	ELCP	MSCP	Class	EL-SCP	ELCP	MSCP
1	86.8	**87.0**	80.8	12	78.8	**80.4**	66.8
2	**89.6**	88.2	85.4	13	74.0	**76.6**	66.4
3	**76.2**	75.4	66.2	14	72.4	**72.6**	66.6
4	86.0	**87.0**	81.2	15	**80.6**	80.4	66.6
5	**72.0**	71.6	66.2	16	89.6	**89.8**	88.8
6	**88.0**	**88.0**	85.4	17	79.8	**80.8**	69.8
7	**62.0**	60.4	57.6	18	**82.2**	81.6	78.0
8	**89.4**	86.2	88.4	19	**80.2**	79.8	66.0
9	**84.4**	83.4	66.0	20	**69.2**	67.6	58.2
10	**80.2**	80.0	79.4	21	**72.8**	67.8	59.8
11	**88.4**	88.2	84.4	**OA**	**89.0**	88.6	80.8

Table 2.8: Classification performance per class for 10% of training images: comparison between MSCP, ELCP and EL-SCP approaches.



	(a)	(b)	(c)	(d)
Ground truth	**Airplane**	**Tennis court**	**Storage tanks**	**Intersection**
ELCP	Runway	Golf course	Intersection	Intersection
EL-SCP	Airplane	Tennis court	Storage tanks	Buildings

Figure 2.31: Examples of images (and saliency maps) correctly classified : only by EL-SCP (a) airplane, (b) tennis court, (c) storage tanks; and only by ELCP (d) intersection.

Therefore, the proposed architecture still have some points that can be improved. In fact, the considered saliency maps are provided by a pre-trained model. The SalGAN model is not retrained on the dataset of interest but is only applied on a transfer learning procedure. To improve that, one can extend the proposed architecture to an end-to-end training process. In addition, the considered saliency map remains the same for each subset on the EL-SCP approach whatever the considered feature maps. This can be overcame by proposing a more adapted strategy where the saliency maps are estimated for each EL-SCP branch according to the considered subset features.

2.5 Decision combination

2.5.1 Comparison between Ens. Hybrid LE FV and ELCP methods

Two transfer learning approaches have been presented, namely Ens. Hybrid LE FV in Section 2.3 and ELCP in Section 2.4. There are some similarities between these two methods. Both are based on covariance pooling of CNN features, where the log-Euclidean framework presented in Section 1.3.3 is adopted. They also exploit an ensemble learning approach. The main difference is that second-order statistics of CNN feature maps are computed locally on the first layers for Ens. Hybrid LE FV, while they are computed globally on deeper layers for ELCP. Unsurprisingly, as observed in Tables 2.5 and 2.7, ELCP has better classification performance than Ens. Hybrid LE FV since it exploits deeper CNN features. A gain of 26 % and 20 % are respectively observed for ELCP compared to the first and second layers of Ens. Hybrid LE FV. However, by looking closely at the classification results, it is possible to find some images that are well classified only by Ens. Hybrid LE FV, whereas ELCP fails at this task. Figure 2.32 shows some images from the UC Merced dataset with the predicted class by these methods. As observed, the first two ones are correctly classified only by Ens. Hybrid LE FV, while for the last two ones, only ELCP succeeds. By taking a closer look at these results, it can be observed that, for the first two images which belong to the baseball diamond class, ELCP seems to focus on the road and building located at the top of the images. Since it exploits deeper layers of a CNN, ELCP learns high-level features that are not so useful for these particular images. Low-level features are sufficient for these images. On the other hand, the third and fourth images of

Figure 2.32, are well classified only by ELCP; since the scene is more complex, high-level features are helpful. It therefore seems natural to combine Ens. Hybrid LE FV and ELCP in order to benefit from both low-level and high-level features. Based on the principle of the most diverse ensembles, the next subsection presents a simple fusion scheme between these two approaches.

	(a)	(b)	(c)	(j)
Ground truth	**Baseball diamond**	**Baseball diamond**	**Airplane**	**River**
Ens. Hybrid LE FV (c1)	Baseball diamond	Baseball diamond	Runway	Forest
Ens. Hybrid LE FV (c2)	Baseball diamond	Baseball diamond	Runway	Forest
ELCP	Intersection	Building	Airplane	River

Figure 2.32: Samples from the UC Merced dataset. Below, ground truth and class prediction by Ens. Hybrid LE FV and ELCP approaches.

2.5.2 Fusion scheme

As previously mentioned, Ens. Hybrid LE FV and ELCP methods can be complementary since they exploit features extracted from different layers. To benefit from both strategies, many multiple classifier systems have been proposed in the literature, such as dynamic selection techniques [Cruz *et al.* 2018]. However, the goal here is not to provide the best way to combine Ens. Hybrid LE FV and ELCP methods but rather to show the potential of their fusion. For that, we will focus on two standard and straightforward strategies. The first one, denoted as Fusion Ens. Hybrid LE FV-ELCP (MV), is simply a majority vote (MV) on the decision obtained on the output of each subset of Ens. Hybrid LE FV and ELCP. The second one, denoted as Fusion Ens. Hybrid LE FV-ELCP (MDE+MV), selects the most diverse ensembles (MDE) from these methods according to the disagreement diversity measure and greedy optimization [Kuncheva & Whitaker 2003]. The disagreement measure was first used in [Skalak 1996] to characterize the diversity between a base classifier and a complementary classifier, and then in [Ho 1998] to measure diversity in decision forests. It is given by the ratio between the number of observations on which one classifier is correct and the other is incorrect to the total number of observations. Here, the objective is to select the most diverse classifiers based on the disagreement measure. More precisely, let's consider an example of a binary classification problem where $\mathbf{X} = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$ be a labeled data set, $\mathbf{x}_i \in \mathbb{R}^d$. An ensemble of L classifiers are used where a classifier D_i output is represented with an $N-$dimensional vector $\mathbf{y}_i = [y_{1,i}, ..., y_{N,i}]$, such that :

$$\begin{cases} y_{j,i} = 1 & \text{if } D_i \text{ recognizes correctly } \mathbf{x_j}, \\ y_{j,i} = 0 & \text{otherwise.} \end{cases} \tag{2.31}$$

The disagreement measure between a pair of classifiers D_i and D_k is assessed as follows:

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{2.32}$$

with N^{ab} is the number of elements $\mathbf{x_i}$ of $\mathbf{X}$ for which $y_{j,i} = a$ and $y_{j,k} = b$ according to the table 2.9 below.

	D_k **correct (1)**	D_k **wrong (0)**
D_i **correct (1)**	N^{11}	N^{10}
D_i **wrong (0)**	N^{01}	N^{00}

Table 2.9: A 2×2 contingency table of the relationship between a pair of classifiers.

This measure is computed for all pairs of classifiers to form the matrix Dis of size $L \times L$. The most diverse classifiers are then selected according to this disagreement measure where $K \leq L$ classifiers are retained providing the highest disagreement measurement. In the end, a majority vote on these selected ensembles is performed. Table 2.10 summarizes the main results obtained on the UC Merced dataset for the original Ens. Hybrid LE FV and ELCP approaches and their fused versions. As observed, since the classification performances are significantly better for ELCP than Ens. Hybrid LE FV, a simple majority vote is not adapted. The accuracy of this fusion scheme (MV) is profoundly affected by the Ens. Hybrid LE FV scheme. However, by selecting the most diverse ensembles (MDE+MV), a slight gain is observed compared to ELCP, illustrating its potential.

Dataset	**Method**	**OA (Mean ± sd)**
UC Merced $p = 10$ %	Ens. Hybrid LE FV (conv1)	62.4 ± 0.9 %
	Ens. Hybrid LE FV (conv2)	68.1 ± 1.7 %
	ELCP	**88.4 ± 1.4 %**
	Fusion Ens. Hybrid LE FV-ELCP (MV)	88.2 ± 1.2 %
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**88.7 ± 1.1 %**

Table 2.10: Classification accuracy on UC Merced dataset obtained using Ens. Hybrid LE FV, ELCP and their fusion version Ens. LE FV - ELCP methods (p = 10 %).

2.6 Experiments on other datasets

In this section, experiments on other remote sensing scene classification datasets are conducted to evaluate the effectiveness of the proposed approach. For that, the SIRI-WHU Google dataset [Zhao *et al.* 2016], the AID dataset and two real texture datasets, respectively, for maritime pine forest and on oyster fields [Regniers *et al.* 2015, Regniers *et al.* 2016] were tested. In order to prove the efficiency of the proposed approaches in challenging conditions, only 10 % of images were considered for training.

2.6.1 Image datasets

2.6.1.1 SIRI-WHU:

This is a 12-class Google image dataset [Zafar & Ali 2019], where each class contains 200 images of 200 × 200 pixels, with a 2-m spatial resolution. This dataset was acquired from Google Earth and covers urban areas in China. Figure 2.33 shows some image examples of the dataset.

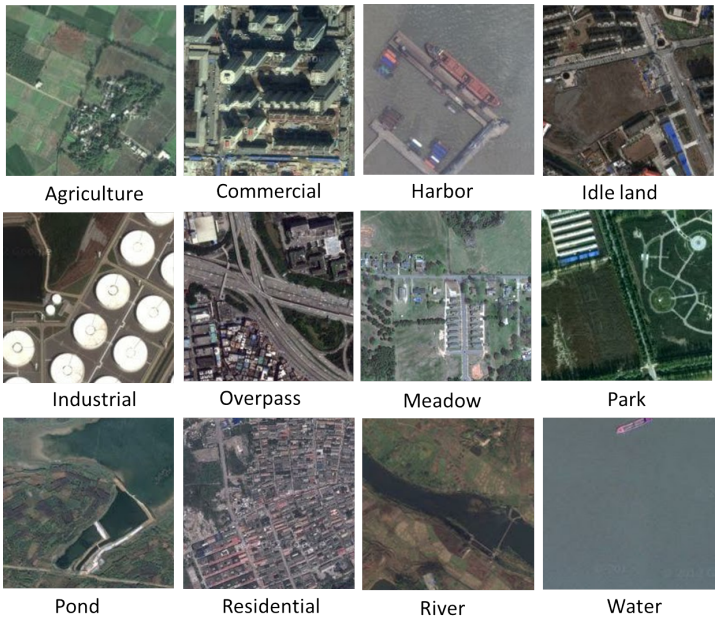


Figure 2.33: Samples from the Google image dataset of SIRI-WHU.

2.6.1.2 Maritime pine forest

This dataset, created in [Regniers 2014] comprises four classes of panchromatic Pléiades satellite images with a spatial resolution of 50 cm. It represents a monitoring of growing maritime pine tree stands located in the South-West of France. Figure 2.34 illustrates one image from each age class. The classes considered in this dataset correspond to three age classes to which is added a clear-cut class. As observed in the images, the textural information evolves from one class to another. In fact, crops are marked by a specific spatial arrangement due to cropping practices and row plantings.

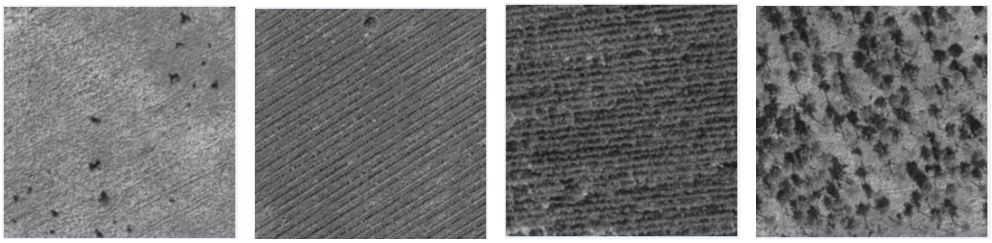


Figure 2.34: Samples from the maritime pine forest dataset.

2.6.1.3 Oyster racks

This five-class dataset [Regniers 2014] is also formed from panchromatic Pléiades satellite high-resolution images. The study site is located in the Arcachon bay, an intertidal lagoon with an area of approximately 180 km^2 located along the Atlantic coast in the South West of France. It is comprised, in particular, of images representing cultivated oyster racks and abandoned fields. Figure 2.35 shows one image of each class of the oyster dataset where it presents also a distinctive spatial organization.

Figure 2.35: Samples from the oyster racks dataset.

2.6.1.4 AID

This dataset contains 10,000 aerial images of dimension 600 × 600 pixels partitioned into 30 classes, with a 2-m spatial resolution. Figure 2.36 illustrates some dataset images.

Figure 2.36: Samples from the AID dataset.

Table 2.11 below summarizes the main characteristics of the considered datasets.

Dataset	Resolution (m)	Classes	Images	Image Size	Image Type
SIRI-WHU	2	12	2,400	200 × 200	Aerial
Maritime pine forests	0.5	4	471	256 × 256	Satellite (Pléiades)
Oyster racks	0.5	5	371	128 × 128	Satellite (Pléiades)
AID	2	30	10,000	600 × 600	Aerial

Table 2.11: Remote sensing scene dataset properties

2.6.2 Classification results

The experiments carried out consist of validating the proposed fusion scheme of the two proposed ensemble learning approaches, namely the Fusion Ens. Hybrid LE FV-ELCP (MDE+MV) strategy.

Table 2.12 summarizes the main results. As observed, a similar conclusion can be drawn from these four datasets. Firstly, the ELCP approach performs better than Ens. Hybrid LE FV on first and second CNN convolutional layers due to the considered convolutional layer depth. This clearly illustrates the interest of exploiting deep feature maps from CNN model, which characterizes high-level features compared to the first ones. Secondly, a similar conclusion can be drawn to the one obtained from the UC Merced dataset: the fusion of both local and global second-order statistics computation strategies permits enhancing classification performance, which illustrates the multi-layer fusion efficiency.

Database	Method	OA (Mean ± sd)
SIRI-WHU $p = 10\%$	Ens. Hybrid LE FV (conv1)	70.0 ± 0.8 %
	Ens. Hybrid LE FV (conv2)	79.1 ± 0.9 %
	ELCP	88.3 ± 1.2 %
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**89.9 ± 1.6 %**
Maritime pine forest $p = 10\%$	Ens. Hybrid LE FV (conv1)	86.5 ± 2.2 %
	Ens. Hybrid LE FV (conv2)	85.7 ± 0.4 %
	ELCP	87.8 ± 2.3 %
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**89.1 ± 1.3 %**
Oyster racks $p = 10\%$	Ens. Hybrid LE FV (conv1)	84. 1 ± 2.4 %
	Ens. Hybrid LE FV (conv2)	**86.1 ± 1.1 %**
	ELCP	85.7 ± 1.4 %
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**86.4 ± 1.4 %**
AID $p = 10\%$	Ens. Hybrid LE FV (conv1)	67.4 ± 0.4 %
	Ens. Hybrid LE FV (conv2)	70.9 ± 0.2 %
	ELCP	87.6 ± 0.2 %
	Fusion Ens. Hybrid LE FV-ELCP (MDE+MV)	**88.7 ± 0.3 %**

Table 2.12: Classification accuracy on different datasets obtained using Ens. Hybrid LE FV, ELCP and their fusion version Ens. LE FV - ELCP methods ($p = 10\%$).

2.7 Conclusions

Throughout this chapter, we started by exploring the most familiar machine learning and deep learning methods dedicated to image classification problems. In addition, the combination of machine and deep learning strategies has proved to be efficient and has shown relevant performance in many applications. In that context, hybrid architectures have been proposed, namely the Fisher network or the NetVLAD method. However, all these strategies exploit only first order statistics and do not take into account dependencies between features, which have been shown to be important in the human visual recognition process. Since then, we introduced different models based on covariance pooling of CNN features to exploit second-order statistics.

The proposed architectures consists of a new transfer learning approach based on the covariance pooling of CNN features maps. The first approach, published in [Akodad *et al.* 2018b], takes advantages of low-level features extracted from the first and second layers. It consists of the log-Euclidean Fisher vector encoding of region covariance matrices computed locally. The second architecture, published in [Akodad *et al.* 2019c], uses high-level features issued from deeper layers that are pooled together by computing their covariance matrix. In order to give

more importance to small objects of interest in the scene, the visual saliency map is computed with SalGAN and then used during covariance pooling. The largest weights are given to the most salient regions. These two strategies use features extracted from models pre-trained on the ImageNet dataset and share many other similarities. They are ensemble learning strategies based on the log-Euclidean representation of the covariance matrix of these CNN features. However, since they exploit feature maps extracted from different layers, they can be considered as complementary. As such, the last proposed architecture, described in [Akodad *et al.* 2020c] is an ensemble learning approach which consists of the fusion of the two previous hybrid architectures. The two ensemble learning strategies were hence combined together using the strategy of the most diverse ensembles. The proposed approach was then successfully validated on various dataset for remote sensing scene classification, illustrating its efficiency and the interest of second-order features. Competitive results have been obtained in challenging conditions where only 10% of images were used for the training process. As a result, a gain of about 1 to 2% were obtained in term of overall accuracy, compared to the recent state-of-the-art.

Since the proposed approach is based on covariance pooling of CNN features, any deep convolutional neural network can be used as backbone. Future works will concern the adaptation of the proposed strategy to multispectral or hyperspectral images dataset, where a CNN will be used for this kind of data [Hu *et al.* 2015, Paoletti *et al.* 2018]. Furthermore, the proposed architecture could be extended to an end-to-end learning strategy which permits developing forward and backward propagation regarding second-order pooling layers. In the same context, the exploited saliency maps can be also generated by a supervised trainable model instead of including them as a fixed input parameter. By doing that, the model will focus on object of interest and exclude useless information. Moreover, as the architecture is based on an ensemble strategy, the saliency map estimation can be fed on each subset.

Finally, another perspective of this work is to look deeper on the architectures to study their generalization capacities. In fact, deep learning has proven itself in several areas, so models are being introduced into increasingly critical applications, such as medical assistance and autonomous navigation. The learning process of neural networks requires fairly large databases, such as the ImageNet database for object recognition used to learn convolutional neural networks (AlexNet, VGG16, VGG19, etc). However, these types of approaches are subject to problems related to uncertainty and their generalization ability. In image classification, there is no control over the behavior of the network to predict objects that have never been included in the learning process. The values produced can therefore be arbitrary. Indeed, quantifying the uncertainty of the model in order to explain the predictions obtained is essential in order to study the confidence that can be given to the predictions, in particular for sensitive applications. This is therefore related to the degree of interpretability and explicability that an AI network is able to provide, in order to meet human needs to understand the reasons for decision-making and the variables involved. In other words, we are aiming to no longer consider deep learning models as black boxes but to try using bayesian approaches and human expertise to qualify the results and define their margin of uncertainty.

Chapter 3

Symmetric positive definite matrix time series classification

Contents

3.1 Introduction

A time series is a sequence of data points which have been acquired during an ordered time segment. Many real-world pattern recognition tasks deal with time-series analysis [Box & Jenkins 1994] such as biomedical signals (e.g. electroencephalography (EEG) and electrocardiography (ECG)), financial data (e.g. stock market and currency exchange rates), industrial devices (e.g. gas sensors and laser excitation), biometrics (e.g. voice, signature and gesture), video processing, data mining, cyber security [Rajkomar *et al.* 2018, Gogolou *et al.* 2019, Nwe *et al.* 2017, Yang & Wu 2006, Susto *et al.* 2018], etc. In fact, any classification problem, using recorded data with a specific order, can be converted into a time series classification problem. In addition, the diversity of the datasets in the UCR/UEA archive [Dau *et al.* 2019], which is the largest repository of time series datasets, shows the different applications of the time series classification problem, as a result of a collaborative effort between researchers at the University of California, Riverside (UCR) and the University of East Anglia (UEA).

Earth observation satellites are increasingly considered as devices that can provide sequences of images. Indeed, recently, the launch of the last generation of Earth observation satellites such as Sentinel-1 and Sentinel-2 has yield more recurrent acquisition of Earth surface images. These sensors allow the acquisition of multivariate time series such as spectral surface reflectance in several wavelengths. Being available for free, these multivariate time series has raised the interest of the remote sensing community to develop novel machine learning strategies for supervised classification. For example, [Courteille *et al.* 2021] proposed an attention-based model to classify Sentinel-2 land cover time series, where [López-Quiroz *et al.* 2009] focuses on ENVISAT radar time series images to analyze subsidence gradients affecting Mexico city.
To deal with time series, many models were proposed in the literature. In particular, many deep learning approaches have recently been proposed [Ismail Fawaz *et al.* 2019]. They include convolutional neural networks (CNN) [Krizhevsky *et al.* 2012] and deep recurrent neural networks (RNN) such as long short-term memory (LSTM) [Hochreiter & Schmidhuber 1997, Ienco *et al.* 2017] and gated recurrent units (GRU) [Cho *et al.* 2014b]. In the following, these approaches will not be considered since the training of such models involves large datasets for avoiding over-fitting and generalization purpose, while the aim here is to work with relatively small datasets. Other non deep learning approaches can be categorized into three main categories: distance-based time series classifiers, feature-based time series classifiers and ensemble classifiers. Distance-based strategies rely on a point-to-point distance between time series which is then fed in a conventional classifier such as a k-nearest neighbor or an SVM. In this family, the most popular approach is certainly the Euclidean distance, which measures the similarity (or dissimilarity) between two time series. As the Euclidean distance suffers from several limitations related to its sensitivity to transformations and to distortions in time dimension, many researchers oriented their work to improve distance-based methods. For example, the SRVF representation [Srivastava *et al.* 2011] ensures re-parameterization invariance while dynamic time warping (DTW) allows to measure the similarity between two time series by aligning them [Sakoe & Chiba 1978, Berndt & Clifford 1994]. It has the ability to match time series that are distorted and shifted along the temporal axis. Inspired by the principle of DTW, some closely related approaches have been proposed such as derivative DTW (DDTW) [Keogh & Pazzani 2001] or weighted DTW (WDTW) [Jeong *et al.* 2011].

Kernel based methods have also been introduced such as the global alignment kernel (GAK) [Cuturi 2011] for the family of DTW distances. More recently, many feature-based methods have been proposed. It consists in extracting features such as wavelet coefficients [Atto *et al.* 2016, D'Urso & Maharaj 2012] or 1D SIFT descriptors from multivariate time series before the classification step. Among these feature-based methods, codebook based representations have raised an interest [Wang *et al.* 2013]. For example, the bag-of-words (BoW) model has been used in [Bailly *et al.* 2016] to obtain an histogram representation of the time series by encoding SIFT features in a codebook. The third family concerns ensemble based classifier systems. The basic idea relies on the combination of multiple classifiers in order to obtain more accurate and robust decisions. Once again, many approaches can be considered. For example, a random forest classifier trains a single base classifier (*i.e.* decision tree) on different subsets of training data (sample, attributes and/or temporal subsets). Another strategy consists in using different classifiers on the same dataset such as in the collective of transformation-based ensembles (COTE) [Bagnall *et al.* 2016a] and its extension based on a hierarchical vote (HIVE-COTE) [Lines *et al.* 2016] where 35 and 37 standalone classifiers are respectively considered. Even if this latter has demonstrated successful results and is considered as the reference for time series classification [Bagnall *et al.* 2017], it suffers from a high computation cost since each classifier should be trained on the whole dataset. Since then, in order to get benefit of the advantages of kernel methods, codebook based representations and ensemble learning strategies, Mikalsen *et al.* have introduced the time series cluster kernel (TCK) method in [Mikalsen *et al.* 2018] which has demonstrated competitive results for times series classification.

At the same time, second-order features have shown a great interest for many image processing applications including person re-identification, texture recognition, material categorization or EEG classification in brain-computer interfaces to cite a few of them [Faraki *et al.* 2015a, Barachant *et al.* 2013, Said *et al.* 2015a]. For example, as shown in chapter 2, the use of covariance matrices has demonstrated to be successful in [Akodad *et al.* 2018b] for remote sensing scene classification. When it comes to time series, covariance matrices can be used to model dependencies between attributes at each timestep. To illustrate the idea, Figure 3.1 shows an example of two time series for two different applications. The first represents an action of running where sensors record the x, y and z coordinates of the hand and knee movements for action recognition application. The second shows the temporal evolution of spectral reflectance and vegetation indices (R, G, B, NDVI, etc.) for recognizing a rice plantation for a remote sensing application.

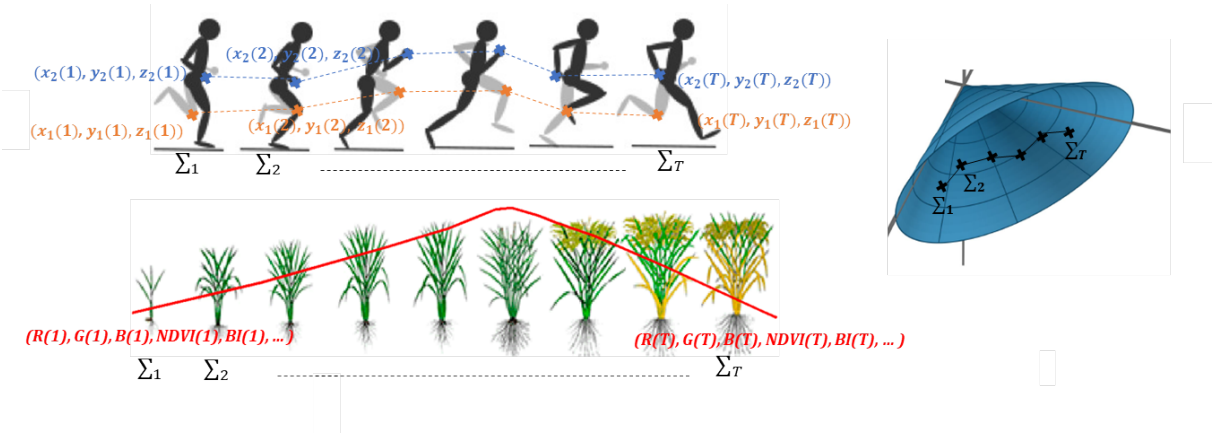


Figure 3.1: Examples of multivariate time series and covariance matrix trajectory.

As a person make an action, the arm and the knee may move in a correlated manner over time. Similarly in the bottom part of Figure 7, the temporal variation of spectral reflectance and vegetation indices of the rice crop may have a correlated behaviour. To capture those correlations between attributes, a covariance matrix Σ_t is computed at each time-step. In the right side of Figure 7, the computed covariance matrices form a trajectory on the space of SPD matrices, illustrated by the blue cone. As such, the multivariate time series classification problem is extended to a covariance matrix trajectory classification problem. Therefore, as explained in chapter 1 and 2, since this type of data lies in a Riemannian manifold, traditional time series classification methods need to be adapted.

Furthermore, to utilize second-order statistics in applications related to time series, second-order trajectories are computed in a way that comparisons between two trajectories are possible. For that, the transport square-root velocity function (TSRVF) representation is recently proposed [Su *et al.* 2014a] as a representation that provides a way to represent trajectories on Riemannian manifolds such that the distance between two trajectories is invariant to identical time-warpings. The method has been widely used for human action and visual-speech recognition applications [Anirudh *et al.* 2017, Zhang *et al.* 2015].

To summarize, the increasing volume of available time series data, covering a large area of applications, have prompted researchers to develop numerous methods and software packages for analyzing such data and extract meaningful information for a certain goal. Therefore, this comes with several issues that need to be addressed. In the case of remote sensing time series, satellite sensors offer a global coverage and different spectral and temporal characteristics but may suffer from shortcoming of sample availability and irregular temporal sampling and thus missing information in the datasets due to cloud contamination for example. In addition, vegetation cycles can be influenced by weather and soil conditions which results on temporal variabilities from one crop to another. More generally, many other applications may face the same issues. All of these challenges require the development of methods capable of dealing with time profile distortions. To that end, we have tried to improve some state of the art methods and the main contributions of this work are the following:

- We present **the dynamic time warping (DTW) framework** which allows aligning time series to ensure speed invariance and decrease distance distortions. As the main purpose is the use of second-order statistics as well as satisfying all desired properties, in particular

re-parameterization invariance, we focus on **the extension of the DTW to second-order feature time series by introducing the SRVF framework for second-order time series, that is the TSRVF**. The main particularity here is to exploit the multiple tangent plane framework introduced in chapter 1 involving the logarithm mapping and the use of parallel transport to bring the features on the same tangent space. Finally, the DTW is performed to align the TSRVF representations.

- We present **how TCK proposed in [Mikalsen *et al.* 2018] for multivariate time series can be extended to work with second-order feature time series, namely the SO-TCK for second-order time series cluster kernel** [Akodad *et al.* 2020a]. Regarding the algorithm complexity, which involves an ensemble learning strategy and a GMM modeling for each subset, the use of multiple tangent planes for handling second-order data is of a very high computational complexity. Thus, the log-Euclidean framework is employed to manipulate second-order statistics on the tangent plane at the identity matrix.

- The **proposed strategies are validated on various labeled time series datasets**. For generalization purposes, datasets of different applications, **from action recognition to remote sensing time series classification** are experimented.

The chapter is structured as follows. Section 3.2 assesses a brief literature review of different multivariate time series classification methods, including machine learning and deep learning based architectures that are commonly used. Section 3.3 introduces the dynamic time warping framework and its extension to align time series of second-order features using the TSRVF representation. Section 3.4.1 presents and discusses the principle of TCK, which can be considered as an ensemble learning strategy for classifying multivariate time series. Then, Section 3.4.2 introduces the proposed extension of TCK for the modeling of the time series of second-order statistics (SO-TCK). In addition, an application on different datasets is next presented in Section 3.5. Finally, Section 3.6 concludes this chapter and provides some perspectives to this work.

3.2 Multivariate time series classification

Time series classification is related to many different domains, such as health, finance, and bioinformatics. Due to its broad applications, researchers have developed many algorithms for these tasks where the overall goal is to identify a time series as coming from one of possibly many classes. For doing that, classification methods were proposed in the literature, whether using traditional machine learning strategies, involving distance metrics, feature based methods or ensemble learning algorithms. Also, with the success known by deep learning architectures in the last years, many researchers focus on proposing neural network based methods for time series classification problems.

Before introducing the different types of classification algorithms whether using machine learning or deep learning strategies, some formal definitions and notations of time series classification are given.

3.2.1 Definitions and notations

In statistics and signal processing, a time series is a sequence of data points exhibiting a temporal dependency, and measured typically at successive times. Time series can be univariate or multivariate depending if one or multiple variables (attributes) are available at each timestep.

- **A univariate time series (UTS):**
  A univariate time series $\mathbf{x} = [\ \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)\ ]$ with $\mathbf{x} \in \mathbb{R}^T$ refers to a time series that consists of a single observation recorded over time increments. The length of $\mathbf{x}$ is equal to the number of instances T.

- **A multivariate time series (MTS):**

  A V-dimensional multivariate time series $\mathbf{X}$, also denoted MTS, consists of a matrix of dimension $V \times T$ where V is the number of attributes. $\mathbf{X}$ is a finite sequence of V univariate time series where $\mathbf{X} = \{\mathbf{x_v} \in \mathbb{R}^T\}$ for $v = 1, \ldots, V$.

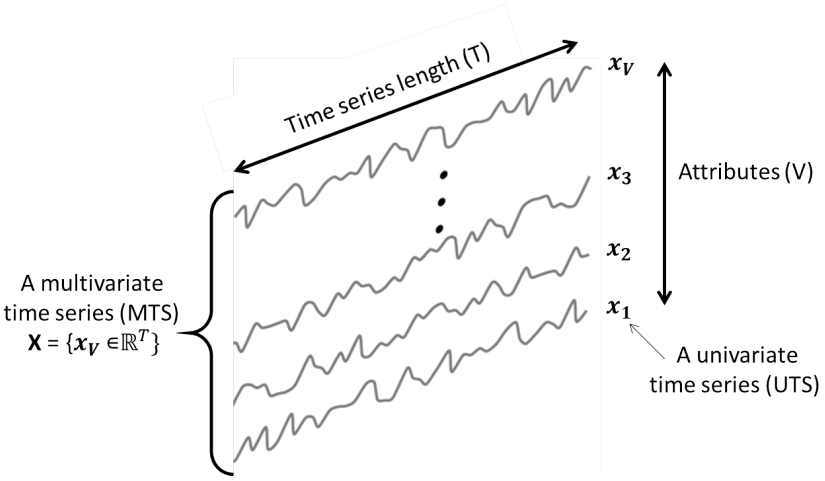


Figure 3.2: Illustration of a multivariate time series (MTS), T: time length, V: number of attributes.

Supervised time series classification (TSC) is a common time series analysis task which aims at restoring a functional dependence between the set of possible time series and the finite set of classes using a training set with known classes. Methods dedicated to TSC can be grouped into two categories: traditional machine learning based methods and deep learning based methods. A brief review of these two families is described in the following subsections.

3.2.2 Machine learning based methods

In this section, an overview of the state-of-the-art time series classification methods are given. This section is dedicated to the most common classification methods.

The practical objective of machine learning is to make correct predictions regarding series that were not seen before in an efficient and robust way. This is done based on many time series classification algorithms that can be categorized onto three families: distance-based, feature-based and ensemble-based methods. The purpose of the following is to introduce each time series classification family as well as detailing some of the most competitive and commonly used methods.

3.2.2.1 Distance-based algorithms

- Distance measurement

The distance-based methods reflects the techniques that measures the similarity between two series in a point-to-point manner. First, a similarity (respectively dissimilarity) measure is a real-valued function that measures how close (respectively dissimilar) two instances are to each other. As such, the closer the instances are, the larger the similarity is. Many similarity measures exist. We start by formally defining distance and similarity functions. For the sake of simplicity, let's consider $\mathbf{x}$ and $\mathbf{y}$ be two univariate time series. A distance is a pairwise function $d : \mathbb{R}^T \times \mathbb{R}^T \rightarrow \mathbb{R}^+$ which satisfies the conditions:

✓Non-negativity: $\mathrm{d}(\mathbf{x},\mathbf{y}) \geq 0$,
✓Identity: $\mathrm{d}(\mathbf{x},\mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$,
✓Symmetry: $\mathrm{d}(\mathbf{x},\mathbf{y}) = \mathrm{d}(\mathbf{y},\mathbf{x})$,
✓Triangle inequality: $\mathrm{d}(\mathbf{x},\mathbf{z}) \leq \mathrm{d}(\mathbf{x},\mathbf{y}) + \mathrm{d}(\mathbf{y},\mathbf{z})$.

The most commonly used distances are based on $\mathcal{L}_p$ distance, which is defined for $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^T$ and $p \geq 0$ as:

$$d_p(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_p = \left(\sum_{t=1}^{T} |\mathbf{x}(t) - \mathbf{y}(t)|^p \right)^{\frac{1}{p}}. \tag{3.1}$$

The $\mathcal{L}_p$ distance, also called Minkowski distance, is a generalization of Manhattan distance for $p = 1$, Euclidean distance for $p = 2$ and Chebyshev distance for $p = \infty$. Regarding time series applications, the definition of suitable distances is essential in order to determine the closeness or common patterns between two time series, among which the most common is the Euclidean distance, also called $\mathcal{L}_2$ norm. The Euclidean distance between two univariate time series $\mathbf{x}$ and $\mathbf{y}$ treats time instances as different features such as:

$$d_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{t=1}^{T} |\mathbf{x}(t) - \mathbf{y}(t)|^2 \right)^{\frac{1}{2}}. \tag{3.2}$$

The Euclidean distance has the advantage of being easy to implement and parameter-free, however, this distance measure is very sensitive to noise and misalignments in time, and is unable to handle local time shifting. In fact, it requires that the two series $\mathbf{x}$ and $\mathbf{y}$ have the same phase and time length T where, in practice, there is no one-to-one correspondence between time sample, and temporal distortions and shifts should be taken into account in distance measurements. To this end, time series need to be realigned before being compared.

- Dynamic time warping (DTW)

For that, the dynamic time warping (DTW) [Kruskal & Liberman 1999] technique, introduced in 1978 in the context of speech recognition [Sakoe & Chiba 1978], is the most well-known algorithm for computing the optimal alignment for a given pair of time series. Intuitively, it is seen as an elastic measure of similarity between two time series which seeks to provide the best alignment between them. As illustrated in Figure 3.3, the idea behind DTW technique allows, unlike the Euclidean distance, to build one-to-many and many-to-one matches between two time series. As such, the effects of shifting and distortion in time are minimized. To do that, the

DTW calculates an optimal match between two given time series where sequences are warped by stretching or shrinking the time dimension. This technique is detailed in section 3.3.1.

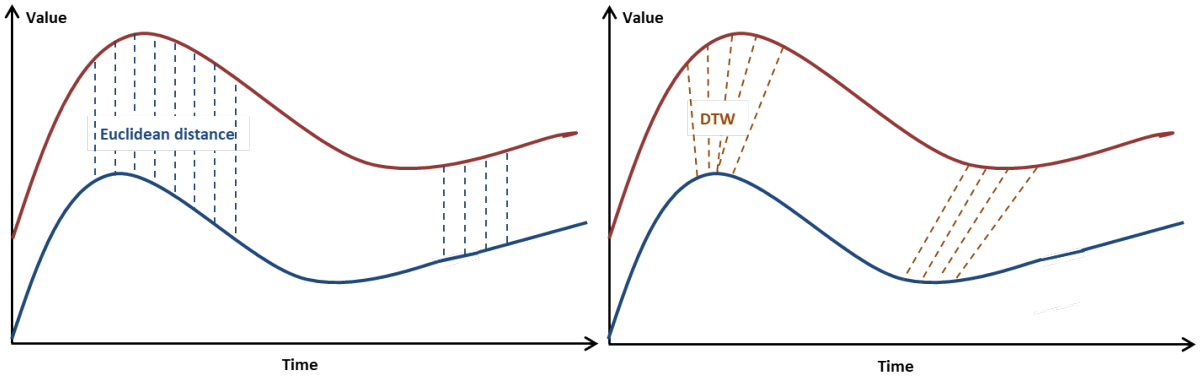


Figure 3.3: Comparison between Euclidean distance and DTW distance of two time series. The latter applies an elastic transformation to the time axis.

Once time series are warped, the classification step is carried out through usual classifiers. One of the simplest and effective ways to exploit a distance measure within a classification process is by using k-NN classifiers where it classifies a time series from the test set with the most given label of its closest time series in the training set. Although approaches from this category provide accurate results where k-NN classifier is fast for small data sets, it becomes slow for large ones, since it needs storing and searching the entire set to compare each observation of the test set with every observation in the training set.

3.2.2.2 Feature-based algorithms

In this category, a high-level representation of time series is constructed, called features, and permits to find a compact description of the considered time series before the classification phase. The main idea is to transform the time series into feature vectors and then use a conventional classifier in this feature space.

Diverse techniques were employed such as Discrete Wavelet Transform [Popivanov & Miller 2002], Discrete Fourier Transform [Faloutsos *et al.* 2000], where features of frequency domain are considered. In these representations, each time series is represented as a feature vector, then, the set of feature vectors are fed together to a classification model such as a Support Vector Machine (SVM) for time series classification.

Other works have investigated the extraction of local and global features in time series. Among these works, Baydogan et al. [Baydogan *et al.* 2013] proposed a framework to classify time series based on bag-of-features representation, denoted TSBF, where local features such as mean, variance and extremum values are computed on sliding windows, another strategy, introduced in [Wang *et al.* 2013], aims at extracting discrete wavelet coefficients on sliding windows and then quantizing them into words using k-means strategy. In another standard approach proposed in [Bailly *et al.* 2016], authors introduced the Bag-of-Temporal-SIFT-Words where the Bag-of-Words (BoW) approach has been extended to time series data. It consists in representing time series using a histogram of word occurrences where words correspond to local SIFT features adapted to mono-dimensional signals. The method starts by extracting

keypoints in time series following the SIFT framework. The difference with the original SIFT method lies on the detection of local extrema in terms of both scale and temporal location using the DoG function adapted to time series. More details about SIFT features extraction are given in chapter 2. This step allows to represent each time series by a collection of feature vectors (SIFT features). The second step is dedicated to the codebook learning of k words using k-means clustering. The words represent different local behaviours in time series and each feature vector is assigned to the closest word in the codebook. At the end, the number of occurrences of each word in the time series is computed. A linear SVM is next used for the classification. In addition, normalization schemes were added to improve the informative power by reducing the influence of frequent codewords. Experiments were conducted over 86 UCR repository available datasets, including a wide variety of problems such as sensor reading of ECG signals, human motion (GunPoint), and were compared with 1NN classifier combined with Euclidean distance and DTW. The classification results and parameters selection are detailed in [Bailly *et al.* 2016]. The method yields interesting performance and improve classification accuracy compared to state-of-the-art methods.

3.2.2.3 Ensemble-based algorithms

Another category that have been explored over the last years for solving time series classification (TSC) is based on ensemble learning strategies. Ensemble learning, based on combining multiple classifiers, have advanced the field by significantly outperforming the other strategies. Many ensemble classifiers methods have been proposed over the last years in the context of time series classification problems where they can be grouped according to those following categories:

- The use of a single model with different input data, where training data is divided into several subsets, such as the random forest method;
- The use of a single model with different training parameters and initializations, as is the case of initial weights for neural networks;
- The use of different models, also called stacking models.

For example, the Collective Of Transformation based Ensembles (COTE) [Bagnall *et al.* 2015], belongs to the third family where it combines 35 classifiers built on four representations of TSC problems: time, frequency, change, and shapelet transformation domains.

Another effective ensemble strategy focuses on combining all classifiers into a flat hierarchy (Flat-COTE) [Bagnall *et al.* 2015]. This latter strategy is an ensemble that combines 35 different classifiers over four data representations where each classifier is built independently and produces separate training accuracies. Given a test sample, the output class of each classifier is weighted by the training accuracy. After pooling operation over all weighted outputs, the class with the highest combined vote is retained. Therefore, due to the imbalanced number of classifiers in each data representation, Flat-COTE can be biased. In fact, time domain classifiers may give higher weights compared to other domains simply because more classifiers are built in the time domain. To overcome this issue, an improved version was proposed, namely the heterogeneous meta-ensemble Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE). It allows to modularise the elements of each group of classifiers. As such, components of a module are the ensemble of classifiers on a certain type. It was first proposed in 2016 [Lines *et al.* 2018], and is defined including two other ensemble classifiers

built in existing feature spaces and allows only a single probabilistic prediction from each domain. This approach, and its updated versions, proved state-of-the-art for accuracy on the UCR benchmark time series datasets. Regarding classification performance, it has been demonstrated that HIVE-COTE is significantly more accurate than Flat-COTE and became the new state-of-the-art for time series classification. In fact, it is characterized by a modulable structure and has the ability to capture more sources of possible discriminatory features in time series. However, as the complexity increases, it involves high resources consumption.

3.2.3 Deep learning based methods

Recently, deep learning methods become commonly used to classify time series and improve the performance of traditional machine learning based approaches. Recurrent neural networks (RNN) such as long short-term memory (LSTMs) and convolutional neural networks (CNNs) are capable of mining dynamical characteristics of time series, hence their success.

3.2.3.1 Recurrent neural networks (RNN) based methods

Recurrent neural networks (RNNs) are dynamical systems that make efficient use of temporal information in the input sequence, both for classification [Ruffini *et al.* 2016, Malhotra *et al.* 2017] and regression [Williams *et al.* 2002, Dunis & Huang 2002]. The key feature of an RNN is that the network has feedback connections which allows to model the effects of the earlier parts of the sequence on the later part of the sequence. As such, its output, at each time step, depends on previous inputs and past computations and hence allows the network to develop a memory of previous events.

The basic structure of a simple RNN consists of a feed-forward part and a memory part; the latter one stores the activations of the feed-forward neurons from the previous time step and serves as additional input for the feed-forward part. This simple RNN is known under the name of the Elman Recurrent Neural Network (ERNN), also as Vanilla RNN. A typical architecture of a simple RNN is depicted in Figure 3.4. The left side of the image is a graphical illustration of the recurrence relation. The right part illustrates how the network unfolds through time over a time series.

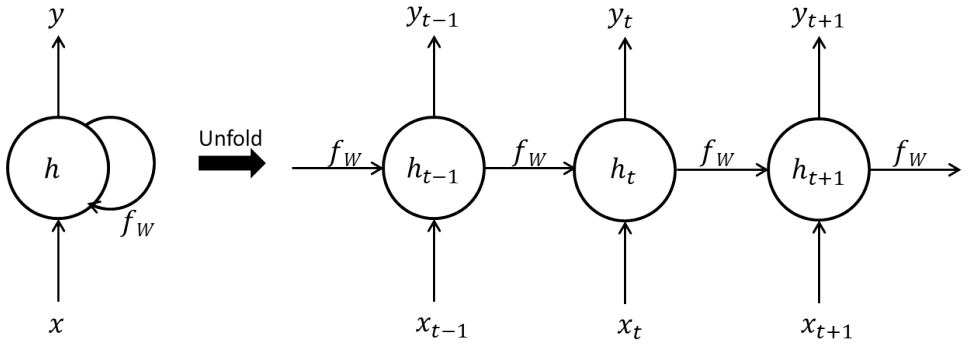


Figure 3.4: The diagram depicts the RNN being unfolded

The RNN is trained on input temporal data $\mathbf{x}_t$ in order to reproduce a desired temporal output $\mathbf{y}_t$. The hidden internal state gets updated every time it reads the input data. The mathematical representation is given below:

$$h_t = f_W(h_{t-1}, \mathbf{x}_t), \tag{3.3}$$

where h_t is the new state, f_W is some function with parameters W, h_{t-1} the old state and $\mathbf{x}_t$ is the input vector at some time step t. While processing, it passes the previous hidden state to the next step of the sequence. As such, the hidden state acts as the neural networks memory where it holds information on previous data the network has seen before.

Theoretically, RNNs can remember long sequences. However, their memory is in practice limited by their finite size. As a result, it has a very short-term memory. In fact, during the learning process, in particular the back propagation, recurrent neural networks suffer from the vanishing gradient problem. It means that earlier layers get small gradient update and thus stop learning. As a consequence, RNNs can forget what it has seen in longer sequences. To overcome memory limitations, recent researchers have led to the design of novel RNN architectures, which are equipped with a permanent memory capable of storing information for long amount of time such as Long Short-Term Memory (LSTM) and Gated Reccurent Unit (GRU), which can be interpreted as a variation or as an extension of RNNs and are commonly used for deep learning applications such as speech recognition, speech analysis, natural language understanding, etc.

The LSTM networks, introduced in [Hochreiter & Schmidhuber 1997] are a modified version of recurrent neural networks, which can handle the information in memory for a longer period of time compared to RNN. It is capable to learn more than 1,000 time steps, depending on the complexity of the built network. A basic LSTM is built of a cell state and it's various gates. Cell state is considered as the memory which carry relevant information during the processing of the series by the use of internal mechanisms called gates. Those latter are the different neural network elements, namely the sigmoïd activation, forget gate, input gate and output gate, that are learned to regulate the flow of information and thus make the decision whether to allow information on the cell state or forget it. Moreover, the Gated Recurrent Units (GRU) [Cho *et al.* 2014a] is similar to an LSTM. It has almost the same gates but has fewer tensor operations which makes him little speedier to train than LSTM.

Furthermore, another deep learning based model has been introduced in the last few years to outperform the RNN, LSTM and GRU models: the transformer [Vaswani *et al.* 2017]. It is a model that uses attention in an encoder-decoder architecture to significantly improve the performance of deep learning models. The key to the transformers performance is its use of attention where impressive results were demonstrated, in particular for applications related to text translation [Bahdanau *et al.* 2016]. It has also been exploited in remote sensing application such as satellite image time series classification [Garnot *et al.* 2020, Sainte Fare Garnot & Landrieu 2020].

3.2.3.2 Convolutional neural networks (CNN) based methods

Motivated by the success of deep CNN architectures in various domains, such as reaching human level performance in image recognition problems [Szegedy *et al.* 2014] as well as different natural language processing tasks [Sutskever *et al.* 2014], many researchers have started exploiting them for time series analysis [Gamboa 2017]. For that, the convolutional filters exhibit only one dimension over time instead of two dimensions corresponding to image width and length. As such, the convolution is applied as a sliding window over the time series.

In this context, many architectures were proposed to deal with multivariate time series, such as the multi-channel deep CNN (MCDCNN) [Zheng *et al.* 2014], where it is considered

as a traditional CNN with an adaptation to multivariate time series data: it is constituted of two convolutional stages, with 8 filters each, a ReLU activation function and a max pooling operation. The convolutions are applied independently on each dimension of the input MTS. The output of the second convolutional stage for all dimensions is concatenated over the channels axis and then fed to a fully-connected layer followed by a softmax classifier.

Similarly, the time convolutional neural network (Time-CNN), proposed in [Zhao *et al.* 2017], is designed for univariate and multivariate time series classification. The main difference with the previous architecture lies in the use of fully-connected layer with a sigmoid activation function instead of the softmax classifier. Another difference to traditional CNNs resides in the use of local average pooling instead of the local max pooling operation. In addition, unlike MCDCNN, for MTS data it applies one convolution for all the dimensions of a multivariate classification task.

3.2.3.3 Hybrid architectures combining RNN and CNN models

Convolutional neural networks (CNNs) are a type of deep neural networks with the ability to act as feature extractors, stacking several convolutional operators to create a hierarchy of more abstract features. Such models are able to learn multiple layers of feature hierarchies automatically. Recurrent neural networks, in particular long-short-term memory (LSTMs) neural networks, are characterized by a memory allowing to model temporal dependencies in time series problems. The combination of CNNs and LSTMs in a unified framework is able to capture time dependencies on features extracted by convolutional operations. For example, DeepConvLSTM, introduced in [Ordóñez & Roggen 2016], has already offered state-of-the-art results in the speech recognition domain. The main difference between DeepConvLSTM and a classical CNN is the topology of the dense layers. In the case of DeepConvLSTM, the units of these layers are LSTM recurrent cells, and in the case of the baseline model, the units are non-recurrent and fully connected. Moreover, the input to the network consists of a data sequence, which is a short time series extracted from the initial time series data using a sliding window. Then, convolutional layers process the input only along the axis representing time. For more details about model implementation, the interested reader may refer to [Ordóñez & Roggen 2016].

In the following, the approaches based on deep neural networks will not be considered since the training process involves large data for avoiding overfitting and generalization purpose, while the aim here is to work with relatively small datasets. Moreover, the following two sections focus on two state-of-the-art approaches for time series classification and their extension to second-order trajectories. First, the dynamic time warping (DTW) and the square-root velocity function (SRVF) representation are detailed as well as their extension to the use of covariance matrices, by applying the transported square-root velocity function (TSRVF). Then, the time series cluster kernel (TCK) architecture is introduced followed by its extension to covariance matrices trajectories by proposing the second-order TCK, called the SO-TCK.

3.3 Dynamic time warping for second-order statistical features

3.3.1 Dynamic time warping (DTW)

When treating time series, the similarity between two sequences of the same length can be calculated by summing the ordered point-to-point distance between them. To do that, the most common distance function is the Euclidean distance [Bagnall *et al.* 2016b], which corresponds to the $\mathcal{L}_2$-norm. Therefore, Euclidean distance and its variants present several drawbacks. In fact, Euclidean distance is sensitive to signal transformations as time shifting which induces inaccurate results in certain applications. This problem of distortion in the time axis can be addressed by Dynamic Time Warping (DTW). DTW was originally designed to treat automatic speech recognition [Sakoe & Chiba 1978] and it looks for the optimal and global alignment between two time series, exploiting temporal distortions between them.

To illustrate that, Figure 3.5 shows the temporal evolution of a rice crop in two different fields. Due to differences among those two regions, such as air temperature, soil drainage and other environmental characteristics, the same crop may have different temporal behaviour while providing the same information, and thus belonging to the same class. In fact, rice growing passes through three different crop stages. It starts by a vegetation phase to reach its highest NDVI value then goes through a reproductive phase, followed by the ripening phase until its decline. The comparison of NDVI among different crop stages shows this temporal shifting between the two series. Indeed, as observed in Figure 3.5, the vegetative phase for crop (b) is longer than the one for crop (a). Hence, the Euclidean distance is not adapted to measure the similarity between the two curves. The irregular rice growth NDVI time series data needs to be separated from the variability of classes.

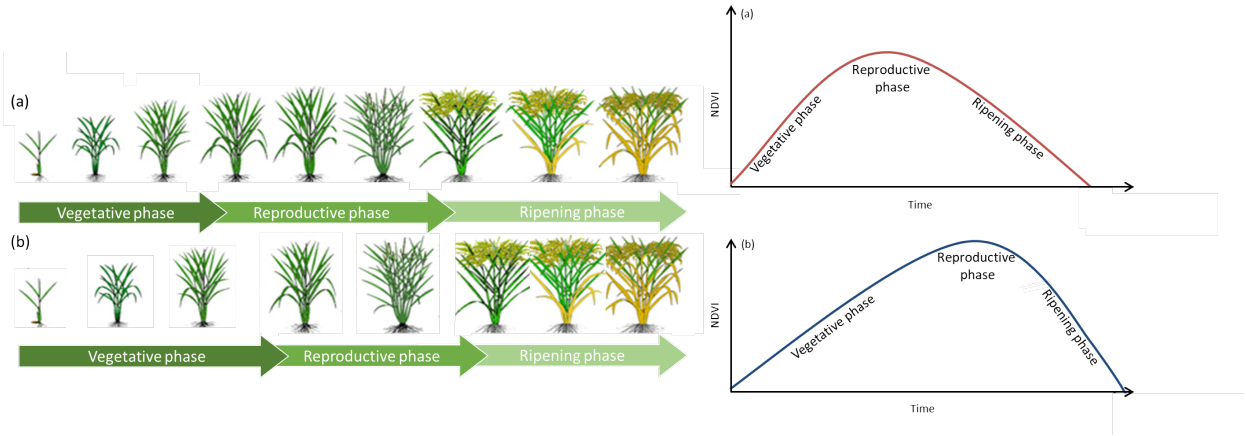


Figure 3.5: Temporal evolution of a rice plant. Left: two different behaviours (a) and (b) illustrating a speed variation inducing a time shifting between the two series. Right: their corresponding normalized difference vegetation index (NDVI).

To address that, the use of an adapted similarity measure is required. Here comes the benefit of using DTW, where it offers an optimal alignment between the two sequences by matching their temporal patterns in which rice planting schedules are flexible. That optimal warping path γ that minimizes the global cost between pairs $\mathbf{x}_1$ and $\mathbf{x}_2$ is found by minimizing the following optimization problem:

$$\mathrm{d}_w(\mathbf{x}_1, \mathbf{x}_2) = \inf_{\gamma}(||\mathbf{x}_1 - \mathbf{x}_2 \circ \gamma||_2), \tag{3.4}$$

where the composition $\mathbf{x}_2 \circ \gamma$ means warping the series $\mathbf{x}_2$ by γ. In the following, we first define

the cost between points of time series, which is used later to generate the optimal warping path.

3.3.1.1 Cumulative cost matrix computation

Formally, given two time series represented by the sequences $\mathbf{x}_1 = [\, \mathbf{x}_1(1), \mathbf{x}_1(2), \ldots, \mathbf{x}_1(T)\,]$ and $\mathbf{x}_2 = [\, \mathbf{x}_2(1), \mathbf{x}_2(2), \ldots, \mathbf{x}_2(T')\,]$ with $T, T' \in \mathbb{N}$, the first step consists of constructing a cost matrix $\mathbf{C} \in \mathbb{R}^{T \times T'}$ where the input at indices (i, j) represents the distance between the time instance i of the series $\mathbf{x}_1$ and the time moment j in $\mathbf{x}_2$. The cost matrix is usually computed using the Euclidean distance of all pairwise observations between $\mathbf{x}_1$ and $\mathbf{x}_2$ as:

$$\mathbf{C} \in \mathbb{R}^{T \times T'} \;:\; \mathbf{C}_{i,j} = ||\mathbf{x}_1(i) - \mathbf{x}_2(j)||,\; i \in [1:T],\; j \in [1:T'] \tag{3.5}$$

It is needed to compute a cumulative cost matrix $\mathbf{C}^+ \in \mathbb{R}^{T \times T'}$ where each element $\mathbf{C}^+_{i,j}$ is the minimum cost for aligning the series $\mathbf{x}_1$ up to point i with series $\mathbf{x}_2$ up to point j. The matrix $\mathbf{C}^+$ is obtained as follows:

$$\mathbf{C}^+_{i,j} = \mathbf{C}_{i,j} + \min(\mathbf{C}^+_{i-1,j-1}, \mathbf{C}^+_{i,j-1}, \mathbf{C}^+_{i-1,j}), \tag{3.6}$$

with $\mathbf{C}^+_{1,1} = \mathbf{C}_{1,1}$. Then, the minimum distance under the best alignment is found in $\mathbf{C}^+_{T,T'}$. The pseudo-code for computing the cumulative cost matrix is detailed in Algorithm 5.

Algorithm 5 Computation of the cumulative cost matrix for DTW

Input: Time series $\mathbf{x}_1$ of length T and $\mathbf{x}_2$ of length T', a distance $d()$.
Initialize: $\mathbf{C}^+$ an empty array.

for $i = 1 : T$ **do**
  $C^+(i, 0) \leftarrow \infty$
end for
for $j = 1 : T'$ **do**
  $C^+(0, j) \leftarrow \infty$
end for
for $i = 1 : T$ **do**
  for $j = 1 : T'$ **do**
    $C^+(i, j) \leftarrow d(\mathbf{x}_1(i), \mathbf{x}_2(j)) + \min\left(C^+(i-1, j-1), C^+(i-1, j), C^+(i, j-1)\right)$
  end for
end for

Output: The cumulative cost matrix C^+.

3.3.1.2 Optimal warping path

Once the accumulated cost matrix is built, the next step aims to find the best match between these two sequences. For that, we can find a path through the matrix that minimizes the total cumulative distance between them, namely the warping path. It is the minimal cost that can be found by backtracking from the end point (T, T') to the start point $(1, 1)$ following the strategy described in Algorithm 6.

Algorithm 6 Optimal warping path

Input: $i = T$ and $j = T'$.
Initialize: path as an empty array.

while $i > 1$ & $j > 1$ **do**
  if $i == 1$ **then**
    $j = j - 1$
  else if $j == 1$ **then**
    $i = i - 1$
  else
    if $\mathbf{C}^+(i-1,j) == \min\{\mathbf{C}^+(i-1,j); \mathbf{C}^+(i,j-1); \mathbf{C}^+(i-1,j-1)\}$ **then**
      $i = i - 1$
    else if $\mathbf{C}^+(i,j-1) == \min\{\mathbf{C}^+(i-1,j); \mathbf{C}^+(i,j-1); \mathbf{C}^+(i-1,j-1)\}$ **then**
      $j = j - 1$
    else
      $i = i - 1; j = j - 1$
    end if
    path.add((i,j))
  end if
end while

Output: path.

To evaluate classification performance of the proposed approach, one time series dataset of the UCI/UCR benchmark datasets is experimented: the Libras dataset. This latter, acronym of the Portuguese name "Lingua BRAsileira de Sinais", represents hand movement of Brazilian official language. It contains 15 classes of 24 instances each, where each class refers to a hand movement type. The hand movement is represented as a bi-dimensional curve performed by the hand in a period of time recorded by a video. As an example, Figure 3.6 illustrates two time series of the Libras dataset. As shown, the two sequences have a similar shape while there is a time shift.

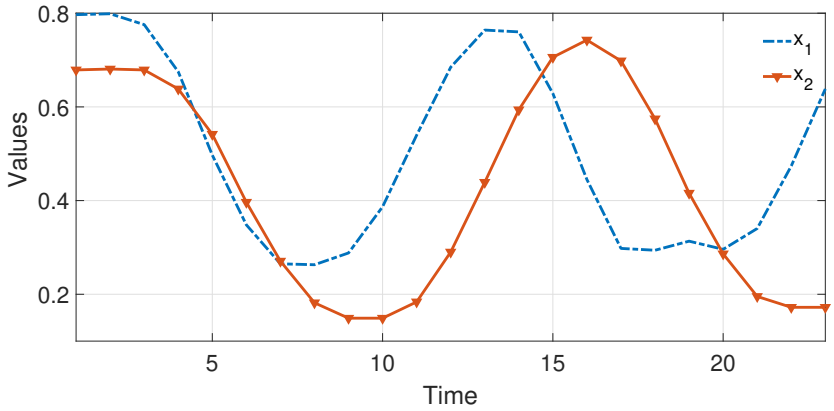


Figure 3.6: Example of two time series $\mathbf{x}_1$ and $\mathbf{x}_2$ from the Libras dataset.

To evaluate the potential of the DTW strategy over the usual Euclidean distance, the dynamic programming is performed to compute the cumulative cost matrix and find the optimal path according to Algorithms 5 and 6. Figure 3.7 shows the resulting cumulative cost between two series $\mathbf{x}_1$ and $\mathbf{x}_2$ and the obtained optimal path is highlighted in white color.

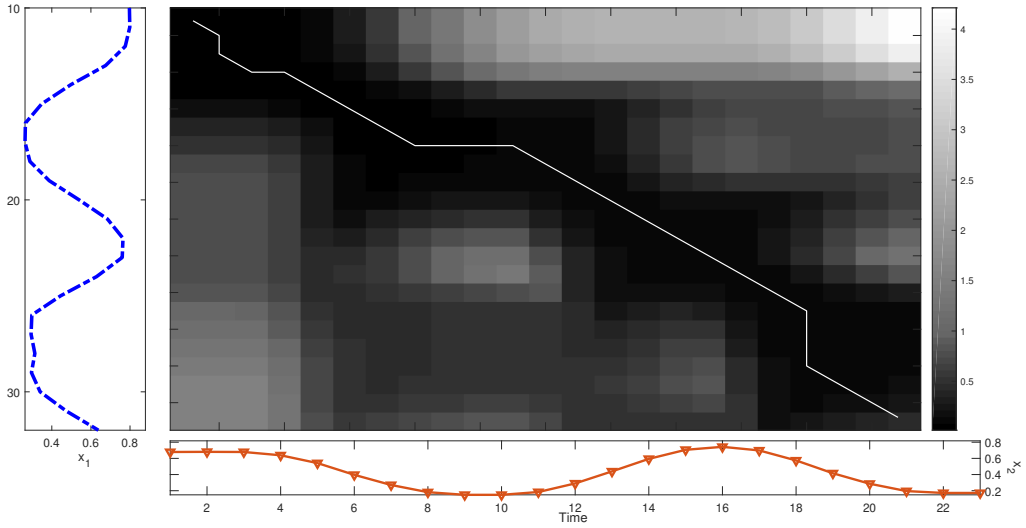


Figure 3.7: Illustration of the cumulative cost matrix between two sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ of the Libras dataset, the optimal warping path is shown in white line.

Once the optimal warping path γ is produced, the warping operation can start in order to align $\mathbf{x}_1$ and $\mathbf{x}_2$ by applying a time dilation or a time contraction. It is performed by matching between corresponding points of one series to another. Figure 3.8 illustrates the original signals in the left and the warped signals in the right.

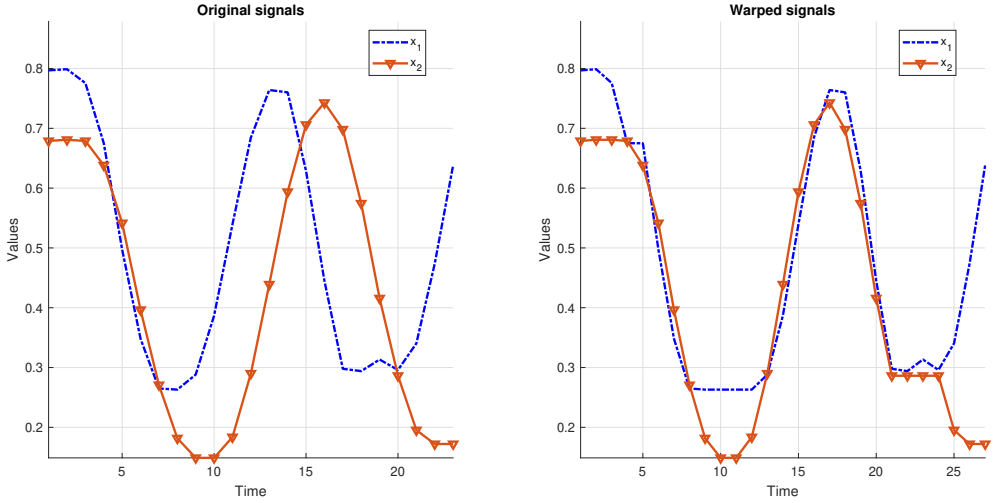


Figure 3.8: Two temporal signals $\mathbf{x}_1$ and $\mathbf{x}_2$ before and after time warping using the DTW method.

The mapping of the points from the first sequence $\mathbf{x}_1$ to points from the other sequence $\mathbf{x}_2$ enables a fair comparison regardless the initial time shifting.

To quantify the efficiency of DTW, the Euclidean distance is computed point-to-point over the original sequences $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$. Then, the series are aligned using the previously explained DTW algorithm, to match the corresponding points, before distance computation. Numerically, it gives the results in Table 3.1.

Euclidean	DTW	$d(\mathbf{x}_1, \mathbf{x}_2)$
✓	×	1.21
✓	✓	0.42

Table 3.1: Distance measurements between $\mathbf{x}_1$ and $\mathbf{x}_2$ with and without using DTW alignment.

Euclidean distance, which assumes the i^{th} point in one sequence is aligned with the i^{th} point in the other, will produce a pessimistic dissimilarity measure. The non-linear dynamic time warped alignment allows a more intuitive distance measure to be calculated.

3.3.1.3 Variations of the DTW

Inspired by the principle of DTW, some closely related approaches have been proposed such as derivative DTW (DDTW) [Keogh & Pazzani 2001] or weighted DTW (WDTW) [Jeong *et al.* 2011]. Kernel based methods have also been introduced such as the global alignment kernel (GAK) [Cuturi 2011] for the family of DTW distances.

One of the investigated ideas in this work is the constrained DTW, also called LDTW [Zhang *et al.* 2017]. In fact, the simple DTW can bring undesired effects when a large number of points from the first time series is mapped to a single point of another time series. To avoid this problem, a common way is to restrict the warping path in the sense that it has to follow the diagonal direction. For that, a threshold β is added for the calculation of the cumulative distance matrix as follows:

$$\mathbf{C}^{+}_{i,j} = \begin{cases} \mathbf{C}_{i,j} + \min(\mathbf{C}^{+}_{i-1,j-1}, \mathbf{C}^{+}_{i,j-1}, \mathbf{C}^{+}_{i-1,j}) & \text{if } |i-j| < \beta \\ \infty & \text{otherwise} \end{cases} \tag{3.7}$$

By doing that, two special cases may arise when choosing the time-constraint on the warping path. First, when $\beta = 0$ the computation is limited to the diagonal elements, which reduces to the Euclidean distance. Second, when β is higher than the temporal length of the longest sequence, the entire DTW matrix is computed which comes to the original DTW method. Since then, the value of β needs to be adjusted to the characteristics of the times series. Therefore, despite being useful for speeding up the processing, when it comes to the considered datasets of this work, this version may not be helpful. In fact, as illustrated in Figure 3.7, the produced warping path is located around the diagonal of the matrix, the condition of the constrained DTW is naturally verified, thus the procedure remains unchanged whether using the original DTW or the improved version of constrained DTW.

A key problem in distance-based methods is to ensure invariance to time-domain warping, also known as re-parameterization. In the following, we present the definition of square-root velocity function representation (SRVF), which enables to evaluate the difference between two curves, and show that it satisfies the re-parameterization invariance. Furthermore, since the interest of second-order features has been demonstrated, such as covariance matrices, and are naturally constrained to be symmetric positive-definite matrices, adapted tools are required for comparing trajectories of covariance matrices. In the following, the transported square-root vector field (TSRVF) representation is introduced as an extension of SRVF to temporal trajectories of second-order statistical features.

3.3.2 Transported square-root vector field (TSRVF)

A main drawback of the Euclidean distance is that it is not invariant to re-parametrisation of the time series. To illustrate that, let's consider γ a transformation function and $\mathbf{x}_1$ and $\mathbf{x}_2$ two time series. It means that, if the trajectories $\mathbf{x}_1$ and $\mathbf{x}_2$ are warped by γ, to result in the composition $\mathbf{x}_1 \circ \gamma$ and $\mathbf{x}_2 \circ \gamma$, also noted $\mathbf{x}_1(\gamma)$ and $\mathbf{x}_2(\gamma)$, which are a time-warped or re-parameterized versions of $\mathbf{x}_1$ and $\mathbf{x}_2$, the Euclidean distance is not preserved, *i.e.*, $\mathrm{d}(\mathbf{x}_1 \circ \gamma, \mathbf{x}_2 \circ \gamma) \neq \mathrm{d}(\mathbf{x}_1, \mathbf{x}_2)$.

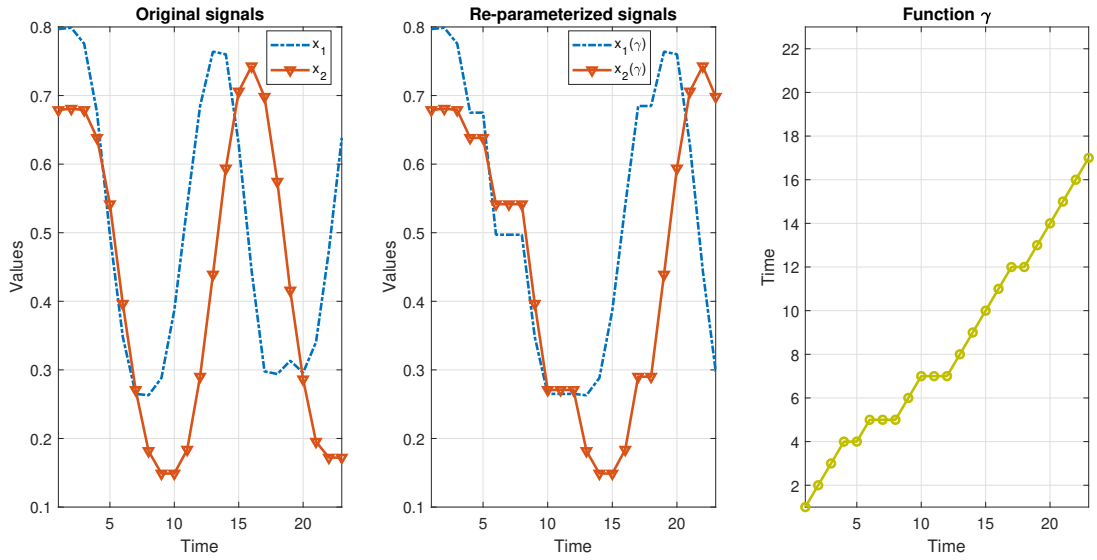


Figure 3.9: Illustration of the original signals, the composition $\mathbf{x}_1(\gamma)$ and $\mathbf{x}_2(\gamma)$ and function γ.

Figure 3.9 illustrates this. On the left, the two times series $\mathbf{x}_1$ and $\mathbf{x}_2$ are displayed respectively in blue and red. By applying a distortion γ (shown on the right of Figure 3.9) on these two times series, the re-parametrized signals $\mathbf{x}_1 \circ \gamma$ and $\mathbf{x}_2 \circ \gamma$ are obtained (on the middle of Figure 3.9). So far we compute the distance $\mathrm{d}(\mathbf{x}_1, \mathbf{x}_2)$ between $\mathbf{x}_1$ and $\mathbf{x}_2$, it will not be preserved under composition. Numerically, the resulting distances $\mathrm{d}(\mathbf{x}_1,\mathbf{x}_2)$ and $\mathrm{d}(\mathbf{x}_1 \circ \gamma,\ \mathbf{x}_2 \circ \gamma)$ are given in Table 3.2.

	$\mathrm{d}(\mathbf{x}_1,\mathbf{x}_2)$	$\mathrm{d}(\mathbf{x}_1 \circ \gamma,\ \mathbf{x}_2 \circ \gamma)$
$\mathcal{L}_2$-norm	1.21	1.06

Table 3.2: Distance comparison before and after warping $\mathbf{x}_1$ and $\mathbf{x}_2$ by γ.

Recent methods from the field of functional analysis and elastic shape analysis [Srivastava *et al.* 2011] were proposed in the literature to overcome those limitations. The square-root velocity function (SRVF) [Joshi *et al.* 2007] method allows development of more efficient solution while providing a rigorous mathematical framework, especially in imposing invariance to rotation, translation, scaling, and re-parameterization. The SRVF associates to each time series its velocity normalized by the square root of its norm and allows to well capture the matching relation between considered sequences. This framework uses the Fisher-Rao metric. That is, it permits to satisfy:

$$\mathrm{d}(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{d}(\mathbf{x}_1 \circ \gamma, \mathbf{x}_2 \circ \gamma), \tag{3.8}$$

3.3.2.1 Computation of SRVF representation

Let $\mathbf{x}$ be a time series, the square-root velocity function (SRVF) is defined as:

$$\mathbf{h}(t) = \frac{\dot{\mathbf{x}}(t)}{\sqrt{|\dot{\mathbf{x}}(t)|}} \in \mathbb{R}. \tag{3.9}$$

This can be seen as a mapping of the sequence onto some feature space. Series comparison is then reduced to the computation of distance in the feature space. To illustrate the produced SRVF transformation, Figure 3.10 shows in left side the series $\mathbf{x}_1$ and $\mathbf{x}_2$ of the Libras dataset, and their corresponding SRVF representation in the right side.

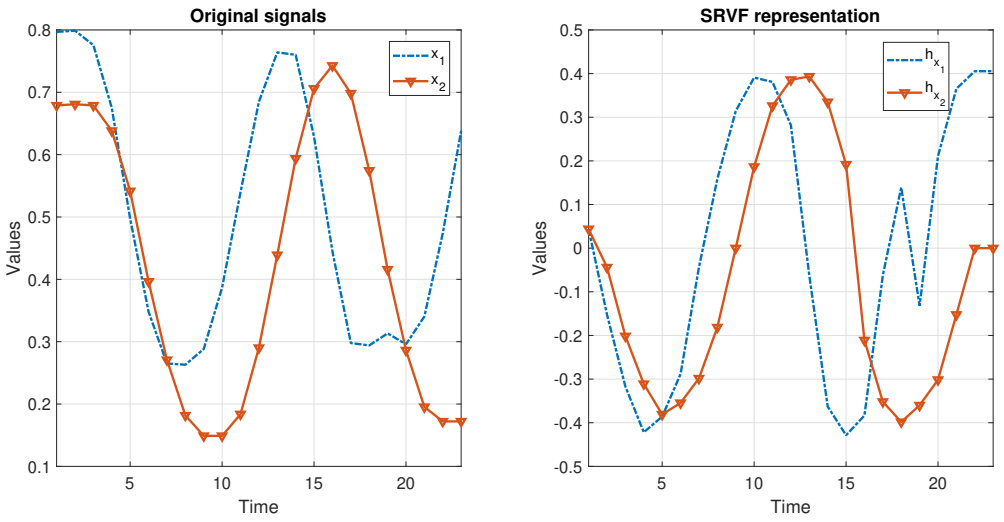


Figure 3.10: Time series $\mathbf{x}_1$ and $\mathbf{x}_2$ and their SRVF representations.

By using this representation, the SRVF offers a natural and efficient framework for aligning the series where distance computation (3.8) is simplified and preserved under warping. To summarize, those following properties are verified. The interested reader is refered to [Srivastava *et al.* 2011] for more details.

- If $\mathbf{x}$ is warped by γ, the SRVF of $\mathbf{x} \circ \gamma$ is given by:

$$\mathbf{h}_{\mathbf{x}\circ\gamma}(t) = \mathbf{h}_{\mathbf{x}}(\gamma(\mathbf{t}))\sqrt{\dot{\gamma}(t)}. \tag{3.10}$$

- Under the SRVF representation, the distance between two sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by the standard $\mathcal{L}_2$ norm, which is the Euclidean distance between their corresponding SRVFs, it satisfies:

$$\mathrm{d}_{SRVF}(\mathbf{x}_1, \mathbf{x}_2) = ||\mathbf{h}_{\mathbf{x}_1} - \mathbf{h}_{\mathbf{x}_2}||_2 = \mathrm{d}(\mathbf{h}_{\mathbf{x}_1}, \mathbf{h}_{\mathbf{x}_2}), \tag{3.11}$$

where d_{SRVF} is the Euclidean distance between the SRVF representations of $\mathbf{x}_1$ and $\mathbf{x}_2$, that are $\mathbf{h}_{\mathbf{x}_1}$and $\mathbf{h}_{\mathbf{x}_2}$.

- For any two SRVFs $\mathbf{h}_{\mathbf{x}_1}$, $\mathbf{h}_{\mathbf{x}_2} \in \mathbb{R}^T$ and a warping function γ. The distance between SRVFs remains unchanged to warping such that:

$$\mathrm{d}_{SRVF}(\mathbf{x}_1 \circ \gamma, \mathbf{x}_2 \circ \gamma) = ||\mathbf{h}_{\mathbf{x}_1\circ\gamma} - \mathbf{h}_{\mathbf{x}_2\circ\gamma}||_2 = ||\mathbf{h}_{\mathbf{x}_1} - \mathbf{h}_{\mathbf{x}_2}||_2 = \mathrm{d}(\mathbf{h}_{\mathbf{x}_1}, \mathbf{h}_{\mathbf{x}_2}). \tag{3.12}$$

To illustrate the efficiency of the SRVF transformation, and its re-parameterization invariance, we refer to the previous example of the two observations $\mathbf{x}_1$ and $\mathbf{x}_2$ from the Libras dataset. The induced distance after computing SRVFs are shown in Table 3.3.

	$\mathrm{d}(\mathbf{h}_{\mathbf{x}_1},\mathbf{h}_{\mathbf{x}_2})$	$\mathrm{d}(\mathbf{h}_{\mathbf{x}_1\circ\gamma}, \mathbf{h}_{\mathbf{x}_2\circ\gamma})$
$\mathcal{L}_2$-norm	2.76	2.76

Table 3.3: Distance comparison before and after re-parameterizing $\mathbf{x}_1$ and $\mathbf{x}_2$ by γ.

As observed, the invariance is well ensured. Moreover, since we assign to each curve its square root velocity function (SRVF), the optimal warping path which aligns $\mathbf{x}_1$ and $\mathbf{x}_2$ is found as the function γ that minimizes the following optimization problem.

$$\mathrm{d}_w(\mathbf{h}_{\mathbf{x_1}}, \mathbf{h}_{\mathbf{x_2}}) = \inf_{\gamma}(||\mathbf{h}_{\mathbf{x_1}} - \mathbf{h}_{\mathbf{x_2}\circ\gamma}||_2) = \inf_{\gamma} \mathrm{d}(\mathbf{h}_{\mathbf{x}_1}, \mathbf{h}_{\mathbf{x}_2}) \tag{3.13}$$

After warping, the distance between $\mathbf{h}_{\mathbf{x}_1}$ and $\mathbf{h}_{\mathbf{x}_2}$ is given by d_w. Computation of γ which better aligns $\mathbf{h}_{x_1}$ and $\mathbf{h}_{x_2}$ can be efficiently done using a dynamic programming method (DTW). For that, Algorithms 5 and 6 are applied.

Furthermore, to extend this framework to second-order trajectories, the TSRVF representation was introduced in [Su *et al.* 2014a]. It offers a way to represent trajectories on Riemannian manifolds and is defined by a parallel transport of scaled-velocity vectors SRVFs, defined in (3.9), of trajectories to a reference tangent space on the manifold, including Riemannian metric and vector space representations. The objective of the TSRVF representation is twofold. The first is related to the data geometry, since they do not obey to conventional Euclidean properties, the TSRVF permits to represent trajectories on a tangent space. The second is related to the need of warping for speed invariance, which causes two sequences to be mis-aligned in time inducing distortions in distance computation and thus classification performance losses.

We focus here on the problem of classification of time series by treating them as second-order trajectories introduced in the following subsection. To start, the computation of an SPD matrix trajectory of the multivariate time series is introduced, then the TSRVF representation framework is defined as well as the distance measurement between two TSRVFs. The theoretical framework used here has been introduced in statistics literature [Su *et al.* 2014a], but our goal here is to show its applicability to multivariate time series classification, in particular for a remote sensing application.

3.3.2.2 SPD matrix time series (SPD-MTS)

In this work, each considered time series is represented by a set of time-dependent second-order features which constitutes a SPD matrix time series (SPD-MTS) as shown in Figure 3.11. Those latter are computed on a sliding temporal window of dimension Δt. Specifically second-order features are determined for overlapping sub-sequences of time series. Overlapping sub-sequences are selected in order to cover all discriminative portions of the time series. Moreover, a shrinkage estimator of the covariance matrix can be used. The simplest way is to add a small ridge $\varepsilon\mathbf{I}$ to each covariance matrix where $\mathbf{I}$ is the identity matrix and $\varepsilon = 10^{-6}$. This regularization is performed to ensure the positive definiteness of the computed covariance matrix. Positive definiteness is important for the LE representation used in the following which involves a logarithm operation over the eigenvalues. We associate for each MTS of dimension $V \times T$ a SPD-MTS of dimension $V \times V \times (T - \Delta t + 1)$.

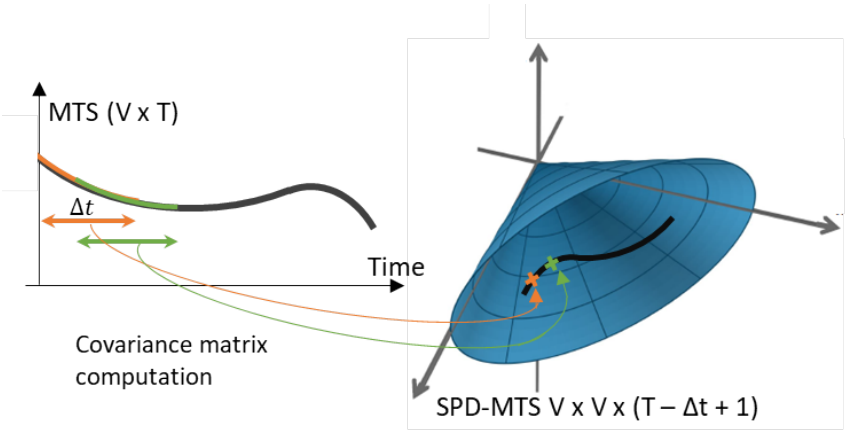


Figure 3.11: Illustration of SPD-MTS computation.

In order to improve the feature representation and enhance classification performance, a full local Gaussian descriptor can be used. In this model, the local mean vector μ is jointly exploited with the $V \times V$ covariance matrix $\mathbf{M}$ which gives the augmented SPD matrix of dimension $(V+1) \times (V+1)$ as proposed in [Lovric *et al.* 2000]:

$$\mathbf{M}_{augmented} = |\mathbf{M}|^{-\frac{1}{V+1}} \begin{bmatrix} \mathbf{M} + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix}. \tag{3.14}$$

3.3.2.3 Computation of TSRVF representation

In order to deal with SPD matrices trajectories, it is required to take into account the geometry of the SPD matrix space and exploit the mathematical framework to manipulate this type of data, particularly, the mapping operations to project data lying on the Riemannian manifold to a tangent space, and the parallel transport to translate data for comparison purposes. As detailed in chapter 1, two options can be considered where two different frameworks were established for handling the specific geometry of covariance matrices. The first one is based on a log-Euclidean metric where it involves projecting the matrices on a unique tangent plane by exploiting the logarithm mapping operator, where the set of covariances is transformed into a vector representation and thus calculations are simplified. For multivariate time series, it comes down to project its corresponding second-order trajectory (SPD-MTS) on a tangent plane, and perform the remaining operations for computing the SRVF representations in the tangent vector space. The second option focuses on a proposition of a more adapted framework where multiple tangent planes are considered. It aims at remaining the closest to the manifold. This allows to better consider the specific geometry of the SPD space and improve the modeling by limiting the distortion when projecting the set of covariance matrices in the tangent space.

As seen in the second chapter, comparisons have been done between a model with a unique tangent plane and a model with multiple tangent planes. This latter has yield to an increasing complexity while classification results remain stable for the two models. As such, the model with a unique tangent space was the best trade-off to limit model complexity while preserving good classification performance. Here, the context is totally different where complexity is much lower; it neither considers an ensemble learning nor apply a GMM modeling. In addition, there is no iterative process such as the EM or Karcher mean algorithms. Thus, one can afford to use the second option of taking into account as much as possible the geometry

of data by considering multiple tangent spaces to the manifold, that is, a scalar product is defined locally on each tangent space.

Let's consider α a trajectory on a Riemannian manifold $\mathcal{M}$, where $\mathcal{M}$ is endowed with a Riemannian metric $<,>$. $\mathbb{M} = \{\alpha : [0,1] \to \mathcal{M}\}$ denotes the set of all trajectories where for a covariance matrix time series $\mathbf{M} = \mathbf{M}(1), \ldots, \mathbf{M}(T)$, α is defined such as $\alpha(0) = \mathbf{M}(1)$ and $\alpha(1) = \mathbf{M}(T)$. By definition, the transported square-root vector field (TSRVF) of any smooth trajectory α is a parallel transport, defined in (1.87) of chapter 1, of a scaled velocity vector field of α to a reference point $c \in \mathcal{M}$ such that (3.9) can be rewritten for data lying in a Riemannian manifold [Su *et al.* 2014a]:

$$\mathbf{h}_\alpha(t) = \frac{\dot{\alpha}(t)_{\alpha(t)\to c}}{\sqrt{|\dot{\alpha}(t)|}} \in \mathcal{T}_c\mathcal{M}, \tag{3.15}$$

where $|\cdot|$ is defined by the Riemannian metric on $\mathcal{M}$ and $\dot{\alpha}(t)$ comes down to project $\alpha(t+1)$ on the tangent plane defined at $\alpha(t)$, using the logarithm mapping operator. Indeed, as seen in Table 1.1 of chapter 1, the derivative $\dot{\alpha}(t)$ is a tangent vector in the tangent space $\mathcal{T}_{\alpha(t)}\mathcal{M}$, it is computed using the logarithm mapping given by:

$$\dot{\alpha}(t) = \mathrm{Log}_{\alpha(t)}\Big(\alpha(t+1)\Big). \tag{3.16}$$

Then, since the velocities $\dot{\alpha}(t)$ are elements of different tangent spaces at different times, by applying the parallel transport from $\alpha(t)$ to c, trajectories are brought back together to the same vector space at c, denoted by $\mathcal{T}_c\mathcal{M}$ as shown in Figure 3.12. By doing that, comparisons become possible.

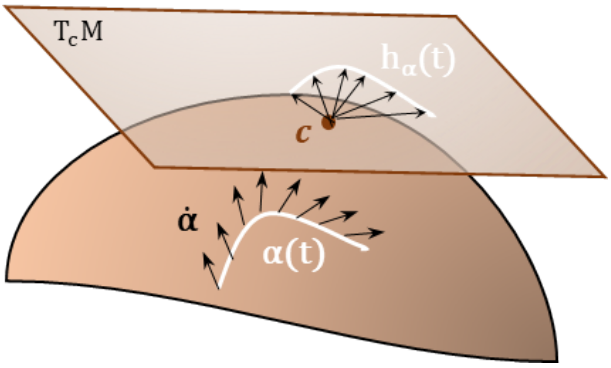


Figure 3.12: Illustration of the TSRVF representation.

As such, each trajectory of covariance matrices, SPD-MTS, is represented by its corresponding TSRVF. As explained in chapter 1, there are many options for the choice of the reference point c. Experimental results in [Su *et al.* 2014a, Su *et al.* 2014b] demonstrated that classification are quite stable with respect to this choice, and thus we will fix it equal to the identity matrix.

- **Distance between TSRVFs:**

Under the TSRVF representation, the distance between two trajectories α_1 and α_2 is given by the standard $\mathcal{L}_2$ norm, between the corresponding TSRVFs $\mathbf{h}_{\alpha_1}$ and $\mathbf{h}_{\alpha_2}$, such that:

$$\mathrm{d}_{TSRVF}(\alpha_1, \alpha_2) = ||\mathbf{h}_{\alpha_1} - \mathbf{h}_{\alpha_2}||_2 = \mathrm{d}(\mathbf{h}_{\alpha_1}, \mathbf{h}_{\alpha_2}). \tag{3.17}$$

Since TSRVF is a path in the tangent space $\mathcal{T}_c\mathcal{M}$, the $\mathcal{L}_2$ norm can be used to compare trajectories. Moreover, the optimal path to align $\mathbf{h}_{\alpha_1}$ and $\mathbf{h}_{\alpha_2}$ is found by:

$$\mathrm{d}_w(\mathbf{h}_{\alpha_1}, \mathbf{h}_{\alpha_2}) = \inf_{\gamma}(||(\mathbf{h}_{\alpha_1} - \mathbf{h}_{\alpha_2 \circ \gamma}||_2) \tag{3.18}$$

Similarly to (3.13), this minimization is solved using the dynamic time warping (DTW) presented in Algorithm 6.

3.3.2.4 Classification performance on Libras dataset

Since the time series have been defined using the TSRVF representation and aligned by the DTW technique to overcome the temporal distortions, a classification algorithm can be applied. Here, a k-NN classifier is performed.

Figure 3.13 compares classification performance involving three second-order descriptors, second-order moment (in blue), covariance matrix (in red), and full local Gaussian descriptor (augmented SPD matrix in green) as a function of considered time segment Δt. The used dataset is the Libras hand movement dataset.

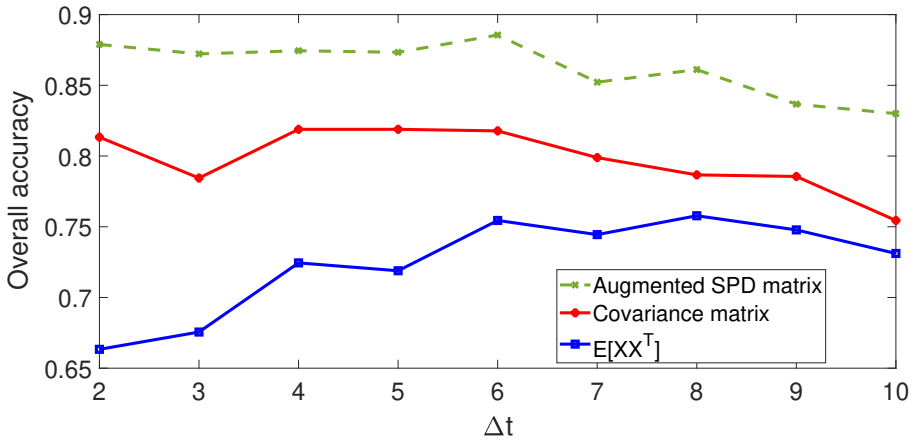


Figure 3.13: Classification comparison between three considered second-order statistics for SPD-MTS computation ($V = 2$).

Since the use of full Gaussian descriptor gives the most successful results, this descriptor is retained for the following experiments. Regarding the choice of Δt, it is selected in such a way that it preserves a good temporal localisation, and at the same time sufficiently large for estimation purpose.

The following experiment, showed in Table 3.4, assesses a classification comparison between different combinations where it provides a threefold interest. First, we aim at demonstrating the benefit of warping the time series to improve similarity measurement using the DTW algorithm detailed in section 3.3.1. For that, the Euclidean distance between time series, introduced in equation (3.2), for a point-to-point measurement is compared with the version where series are first aligned using the DTW before distance measurement. Second, for the purpose of satisfying all desired properties, in particular distance re-parameterization invariance, we look at the classification performance while representing time series with their corresponding SRVF representation introduced in equation (3.9). The two versions, whether aligning series or not

are experimented. Finally, we seek to highlight the potential of using second-order statistics through the TSRVF representation in equation (3.15) which involves multiple tangent planes whether the DTW alignment is applied or not. The experimented data is the Libras dataset and the applied classifier is a nearest neighbor classifier.

Methods	**OA** (%)
Euclidean distance	79.2 ± 1.1
Warping + Euclidean distance	85.6 ± 3.8
SRVF + Euclidean distance	82.8 ± 2.3
SRVF + Warping + Euclidean distance	82.8 ± 2.3
TSRVF + Euclidean distance	80.1 ± 3.0
TSRVF + Warping + Euclidean distance	**88.5 ± 2.7**

Table 3.4: Classification performance on the Libras dataset for different combinations involving the use of Euclidean distance, the dynamic time warping and the first and second-order representations SRVF and TSRVF, respectively.

As shown, exploiting second-order statistics, by using SPD-MTS trajectories, improves classification performance. In addition, as almost all methods that exploit warping technique outperform those which do not use it, it proves the need of warping trajectories to match corresponding time series points.

Furthermore, as the benefit of exploiting ensemble learning strategies has been demonstrated in the previous chapter, where it enhances the classification robustness, the focus in the following is mainly on an ensemble learning based architecture, called the time series cluster (TCK) strategy proposed in [Mikalsen *et al.* 2018], and our contribution to extend it to the use of second-order statistical features.

3.4 Time series cluster Kernel for second-order statistical features (SO-TCK)

3.4.1 Time series cluster kernel (TCK)

TCK has recently been introduced in [Mikalsen *et al.* 2018] for the classification of multivariate time series. It exploits the power of kernel methods, codebook based representations and ensemble learning strategies. The global principle is explained in Figure 3.14. The main idea behind this method is to compute a positive semi-definite similarity measure (*i.e.* a kernel) between two multivariate times series (MTS). For that, a GMM model is first trained on a sample extracted from the training set, which is next used to encode each multivariate time series. Moreover, to ensure robustness, an ensemble learning strategy is considered. The next subsection presents the main steps during training and testing.

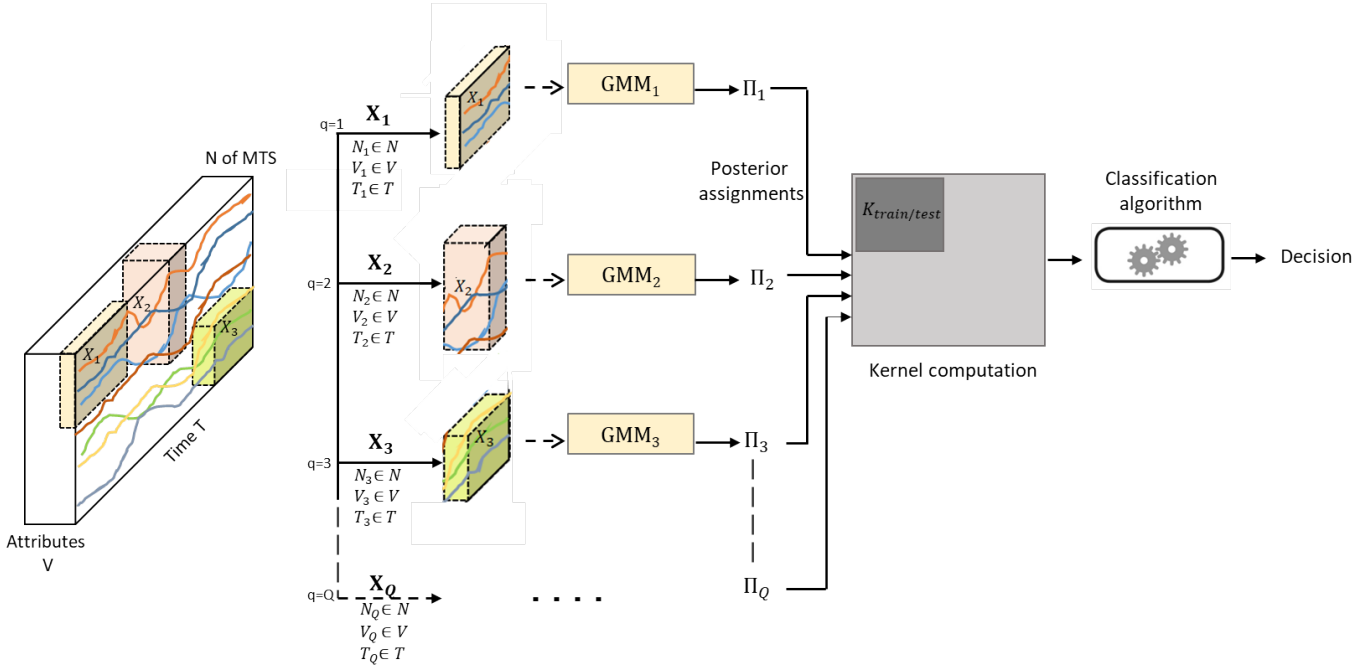


Figure 3.14: General principle of the time series cluster kernel (TCK).

3.4.1.1 Training phase

A multivariate time series (MTS) $\mathbf{X}$ is represented as a matrix of dimension $V \times T$ where V is the number of attributes and T is the time length. It is a finite sequence of V univariate time series, *i.e.* $\mathbf{X} = \left\{\mathbf{x}_v \in \mathbb{R}^T\right\}$ for $v = 1, \ldots, V$. During this stage, N MTS are considered for training, and $\mathbf{X}^{(n)}$ represents the n^{th} MTS sample. To cope with missing data, a second MTS is considered. $\mathbf{R}^{(n)}$ is a binary MTS defined by $\mathbf{r}_v^{(n)}(t) = 0$ if the value $\mathbf{x}_v^{(n)}(t)$ is missing and $\mathbf{r}_v^{(n)}(t) = 1$ otherwise.

As explained before and shown in Figure 3.14, TCK is based on an ensemble learning approach. The considered ensemble learning is based on using the same model, for instance the GMM modeling, with different parameters and initializations. Practically, Q subsets are considered, where each subset is a subsample of N_q data, V_q attributes and T_q consecutive time instances extracted from the training set. For example, as illustrated in Figure 3.15, the subset $\mathbf{X}_1$ is constituted of N_1 time series (orange, light and dark blue) of length T_1 and described by V_1 attributes. Then, for each subset, a codebook is created by learning a GMM model to estimate weights ω_g, means μ_g and variances σ_g^2. Here, for the illustration, the considered number of GMM components is $G = 4$. Those GMM parameters are then used to encode each MTS and produce posterior assignments Π_1.

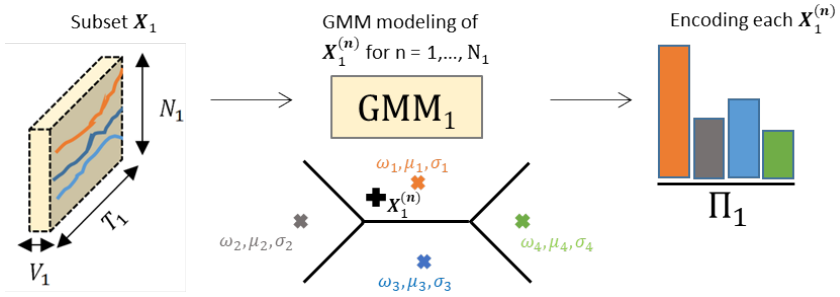


Figure 3.15: Diagram of TCK steps for a single subset ($q = 1$).

The illustrated steps are fully detailed in the following.

- GMM modeling for codebook generation

For the q^{th} subset, the training set is composed of N_q MTS $\mathbf{X}_q^{(n)}$ of dimension $V_q \times T_q$ with their associated binary MTS $\mathbf{R}_q^{(n)}$. A GMM model with G[1] components is considered to learn a codebook, where its probability density function for the incompletely observed MTS $(\mathbf{X}_q^{(n)}, \mathbf{R}_q^{(n)})$ is given by:

$$p(\mathbf{X}_q^{(n)}|\mathbf{R}_q^{(n)}, \Theta) = \sum_{g=1}^{G} \omega_g \prod_{v=1}^{V_q} \prod_{t=1}^{T_q} \mathcal{N}(\mathbf{x}_v^{(n)}(t)|\mu_{gv}(t), \sigma_{gv})^{\mathbf{r}_v^{(n)}(t)}, \tag{3.19}$$

where $\Theta = \{\omega_g, \mu_{gv}, \sigma_{gv}\}$ for $g = 1, \ldots, G$. MTS are assumed to have time-dependent means, *i.e.* $\{\mu_{gv} \in \mathbb{R}^{T_q}\}$ for each attribute v. To enforce regularity, a kernel-based Gaussian prior is defined for the mean. In addition, the covariance matrix for each Gaussian components is assumed to be diagonal and time-independent, that is $\Sigma_g = \text{diag}\{\sigma_{g1}^2, \ldots, \sigma_{gV_q}^2\}$ and σ_{gv}^2 is the variance of attribute v for data belonging to the g^{th} cluster. To estimate the GMM parameters Θ, a MAP-EM algorithm has been proposed in [Mikalsen *et al.* 2018]. The algorithm is detailed in Algorithm 7. The set composed by the estimated GMM parameters represents the codebook. Here, by referring to the different families of ensemble classifiers, the used ensemble strategy is based on training a same model, for instance the GMM model, by using different parameters and initializations. As such, to ensure even more robustness, the number of GMM components G and the initialization of the MAP-EM algorithm are randomly selected for each subset.

As such, to estimate the GMM parameters Θ, a MAP-EM algorithm is considered. For that, some priors are defined for the mean μ_{gv} and the deviation σ_{gv}, where:

- $P(\mu_{gv}) = \mathcal{N}(\mu_{gv}|m_v, S_v)$, with m_v are the empirical means and S_v the prior covariance matrices defined as $S_v = s_v\mathcal{K}$. s_v are the empirical standard deviations and $\mathcal{K}$ a kernel matrix such as: $\mathcal{K}_{tt'} = b_0 \exp(-a_0(t-t')^2)$, $t, t' = 1, \ldots, T$. a_0 and b_0 are user-defined hyper-parameters.
- $P(\sigma_{gv}) \propto \sigma_{gv}^{-N_0} \exp(-\frac{N_0 s_v}{2\sigma_{gv}^2})$, with N_0 a user-defined hyper-parameter.

The set of hyper-parameters is denoted $\Omega = \{a_0, b_0, N_0\}$ and the estimates θ are found using the MAP-EM algorithm as detailed in Algorithm 7.

- Coding method

The encoding of the n^{th} MTS $\mathbf{X}^{(n)}$ for subset q is carried out by computing the posterior assignment $\Pi_q^{(n)}$, obtained by:

$$\Pi_q^{(n)}(g) = \frac{\omega_g \prod_{v=1}^{V_q} \prod_{t=1}^{T_q} \mathcal{N}(\mathbf{x}_v^{(n)}(t)|\mu_{gv}(t), \sigma_{gv})^{\mathbf{r}_v^{(n)}(t)}}{\sum_{k=1}^{G} \omega_k \prod_{v=1}^{V_q} \prod_{t=1}^{T_q} \mathcal{N}(\mathbf{x}_v^{(n)}(t)|\mu_{kv}(t), \sigma_{kv})^{\mathbf{r}_v^{(n)}(t)}}. \tag{3.20}$$

[1] To avoid notation confusion, in this chapter G refers to the GMM components whereas K refers to the kernel matrix.

Algorithm 7 MAP-EM for GMM parameters estimation

Input: Time series dataset $\mathbf{X}^{(n)}_{\{n=1,\dots,N\}}$ of length T, hyper-parameters Ω and G number of GMM components.

Initialize: the parameters Θ.

1: *E-step*: For each $\mathbf{X}^{(n)}$, compute the posterior probabilities $\Pi^{(n)}(g)$ using current parameters Θ.

2: *M-Step*: Update parameters using the current posteriors until convergence.

$\theta_g = \frac{1}{N}\sum_{n=1}^{N}\Pi^{(n)}(g)$

$\sigma^2_{gv} = \left(N_0 + \sum_{n=1}^{N}\sum_{t=1}^{T}\mathbf{r}_v^{(n)}(t)\Pi^{(n)}(g)\right)^{-1}\left(N_0 s_v^2 + \sum_{n=1}^{N}\sum_{t=1}^{T}\mathbf{r}_v^{(n)}(t)\Pi^{(n)}(g)(\mathbf{x}_v^{(n)}(t) - \mu_{gv}(t))^2\right)$

$\mu_{gv} = \left(S_v^{-1} + \sigma_{gv}^{-2}\sum_{n=1}^{N}\Pi^{(n)}(g)\mathrm{diag}(\mathbf{r}_v^{(n)})\right)^{-1}\left(S_v^{-1}m_v + \sigma_{gv}^{-2}\sum_{n=1}^{N}\Pi^{(n)}(g)\mathrm{diag}(r_v^{(n)})\mathbf{x}_n^{(n)}\right)$

Output: Mixture parameters Θ.

Then, the vectors of each component g are concatenated to obtain the final posterior assignment $\Pi_q^{(n)}$ of length G. Figure 3.16 shows visually an example of encoding an observation $\mathbf{X}^{(n)}$ where $G = 4$ clusters are considered for the GMM modeling. As seen, since $\mathbf{X}^{(n)}$ is located closer to the orange cluster than the others (gray, blue, green), the posterior assignment of $\mathbf{X}^{(n)}$ to belong to that cluster is the highest, which corresponds to the largest bar in the diagram.

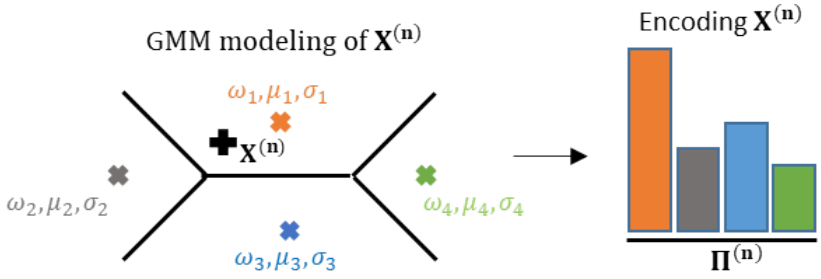


Figure 3.16: Illustration of encoding a sample $\mathbf{X}^{(n)}$ on the space of GMM parameters.

In the following, vector $\Pi_q^{(n)}$ containing the G posterior probabilities $\Pi_q^{(n)}(g)$ is considered to define the feature map used in the kernel.

As different number of components G is considered for each subset q, the model has the ability to capture different levels of granularity in the data. As such, considering a low number of components allows a global comparison of the considered time series segments, whereas a high number of components reflects a more local comparison to extract specific details. To visually explain that, let's take the example of two samples $\mathbf{X}_1^{(n)}$ and $\mathbf{X}_1^{(m)}$ from the subset $\mathbf{X}_1$. Figure 3.17 gives two examples of GMM modeling where in (a), three components ($G = 3$) are considered, while in (b), a larger number of components ($G = 7$) is taken and their corresponding encoding $\Pi_1^{(n)}$ and $\Pi_1^{(m)}$. As shown, both samples, $\mathbf{X}_1^{(n)}$ and $\mathbf{X}_1^{(m)}$ are assigned to the same cluster in (a) while they are separated on two different clusters in (b).

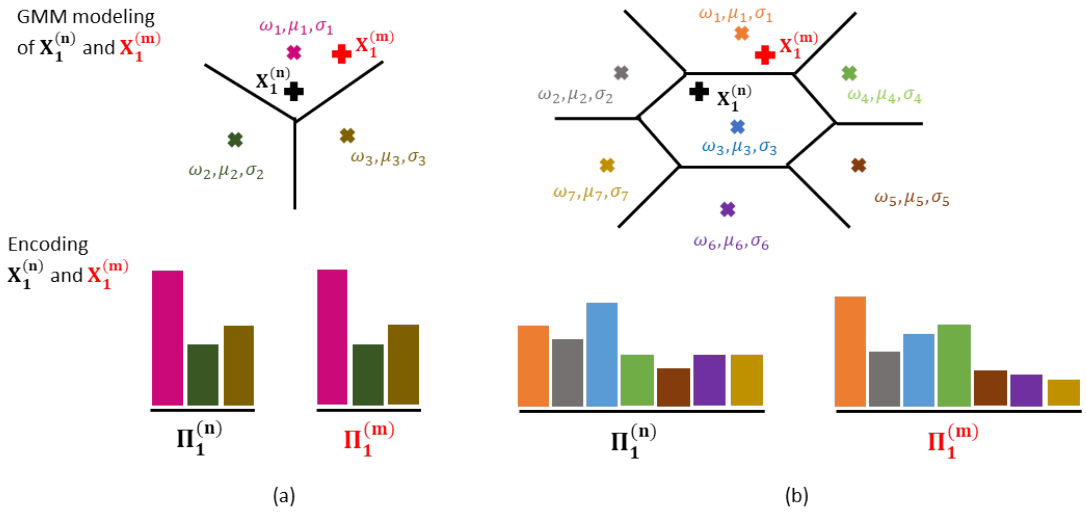


Figure 3.17: Example of effect on the encoding of two different GMM modeling settings. Left: using three GMM components ($G = 3$). Right: using seven GMM components ($G = 7$).

As shown, for model (a), the encoding of the two samples is similar while for (b) the produced vectors $\Pi_1^{(n)}$ and $\Pi_1^{(m)}$ are different. In fact, considering a high number of components will allow to focus on underlying details on the sequence. In a complementary manner, a low number of component permits a comparison at a most general level. This supports the benefit of varying the parameterization to improve the discriminating power of the model.

Then, to build the TCK kernel matrix, the different parameters are set as follows:

- Each GMM model uses a different number of components from the interval $[2, C]$, where for each component, Q represents different random initialization conditions (q_1) and number of components (G) such as: $Q = \{(q_1, G) \mid q_1 = 1, \dots Q,\ G = 2, \dots, C\}$.
- As each GMM model is trained on a random subset of MTS, a random subset of attributes is denoted $V_q < V$, over a randomly chosen time segment $T_q < T$.
- Then, the inner product of posterior probabilities from each mixture component are then added up to build the TCK kernel matrix.

To summarize, the ensemble learning strategy to build the TCK kernel matrix is described in Algorithm 8.

Algorithm 8 Training phase

Input: Training set $\mathbf{X}^{(n)}_{\{n=1,\dots,N\}}$ of length T, Q initializations, C maximal number of mixture components.

1: **for** $q \in Q$ **do**
2: Define subsets with randomly selecting:
i. hyper-parameters $\Omega(q)$
ii. a time segment T_q
iii. a subset of V_q
iv. a subset N_q of MTS
v. initialization of the mixture parameters $\Theta(q)$
3: Estimate GMM parameters by applying Algorithm 7 with G clusters.
4: Compute posteriors $\Pi_q^{(n)} = \left(\pi_q^{(n)}(1), \dots, \pi_q^{(n)}(G)\right)^T$, $n = 1, \dots, N$ with (3.20).
5: **end for**

Output: Time segments T_q, subsets of attributes V_q, subsets of MTS N_q, GMM parameters $\Theta(q)$ and posteriors $\Pi_q^{(n)}$.

3.4.1.2 Testing step

To evaluate the TCK over time series from the test set, the sub-sampling parameters need to be stored in order to estimate corresponding posterior assignments. Using the estimated mixture parameters in the training phase, the computation of posterior distributions of the testing time series dataset remains similar.

To compute a similarity measure between a training MTS $\mathbf{X}^{(n)}$ and a testing MTS $\mathbf{X}^{(m)}$, a kernel based approach is considered. It is built on the basis of an inner product between two posterior distributions to form a linear kernel in the space of posterior distributions as:

$$K\left(\mathbf{X}^{(n)}, \mathbf{X}^{(m)}\right) = K_{nm} = \sum_{q=1}^{Q} \Pi_q^{(n)T} \Pi_q^{(m)}, \tag{3.21}$$

where $\Pi_q^{(n)}$ and $\Pi_q^{(m)}$ are respectively the vector of posterior probabilities for the training and testing MTS obtained with (3.20). To summarize, the Algorithm 9 describes the testing phase steps for building the testing kernel matrix.

Algorithm 9 Testing phase

Input: Testing set $\mathbf{X}^{(m)}_{\{m=1,\dots,M\}}$, time segments T_q, subset of attributes V_q, subsets of MTS N_q, GMM parameters $\Theta(q)$ and posteriors $\Pi_q^{(n)}$.
Initialize: the kernel matrix $K = \mathbf{0}_{N \times M}$.

1: **for** $q \in Q$ **do**
2: Compute posteriors $\Pi_q^{(m)}$, $m = 1, \dots, M$ by applying (3.20) with mixture parameters $\Theta(q)$
3: Update kernel matrix $K_{nm} = K_{nm} + \Pi_q^{(n)T} \Pi_q^{(m)}$, $n, m = 1, \dots, N$
4: **end for**

Output: K TCK test kernel matrix.

In the end, a nearest neighbor classifier is used with the induced distance d between $\mathbf{X}^{(n)}$ and $\mathbf{X}^{(m)}$ given by:

$$d^2\left(\mathbf{X}^{(n)},\mathbf{X}^{(m)}\right) = K\left(\mathbf{X}^{(n)},\mathbf{X}^{(n)}\right) - 2K\left(\mathbf{X}^{(n)},\mathbf{X}^{(m)}\right) + K\left(\mathbf{X}^{(m)},\mathbf{X}^{(m)}\right). \tag{3.22}$$

Again, as observed in (3.21) and (3.22), two MTS will be similar if their posterior probability vectors $\Pi_q^{(n)}$ and $\Pi_q^{(m)}$ are similar for each subset q.

At the end, this approach employs the existing time series distances within k-NN classifiers. In particular, the 1-NN classifier which has mostly been used in time series classification due to its simplicity. Given a distance measure and a time series, the 1-NN classifier predicts the class of this series as the class of the object closest to it from the training set.

3.4.2 TCK for second-order statistical features

As seen in the previous chapter, second-order features, in particular covariance matrices features, have proved to play an important role in different tasks related to visual recognition process. Compared to first-order feature based classification algorithms, many authors have shown the interest of exploiting second-order statistics such as covariance matrix attributes [Faraki *et al.* 2015a, Barachant *et al.* 2013, Said *et al.* 2015a]. This kind of data has a particular structure, they are symmetric positive-definite (SPD) matrices. One of the major contributions of the thesis is a novel representation for time series. The proposed representation is based on feature covariance matrices. This section focuses on introducing the log-Euclidean representation of SPD-MTS in order to exploit them in TCK, yielding to the so called second-order time series cluster kernel (SO-TCK) method.

3.4.2.1 Log-Euclidean representation of SPD-MTS

In order to adapt TCK to work with SPD-MTS, the geometry of the space $\mathcal{P}_d$ of $d \times d$ symmetric and positive definite (SPD) matrices should be considered. In case of using covariance matrices, $d = V_q$, while for the augmented SPD matrix: $d = V_q + 1$. As observed in Figure 3.11, SPD matrices lie in a convex cone which is a Riemannian manifold. Tools developed in the context of Euclidean geometry are hence not adapted to manipulate these data points. A Riemannian metric is better suited such as the log-Euclidean (LE) one [Arsigny *et al.* 2006] or the affine-invariant (AI) metric. As detailed in chapter 1 and 2, considering the LE metric is as efficient in practice as the AI metric. In addition, it avoids adding more complexity to the model and yield to high computational costs. It consists in projecting the set of SPD matrices on a tangent space defined at a reference point, classically considered at the identity matrix. After being projected on the tangent space, tools of the Euclidean geometry can be used such as the MAP-EM algorithm defined in Section 3.4.1.1 to estimate GMM parameters and the Kernel method presented in Section 3.4.1.2. Practically, each SPD matrix $\mathbf{M}$ is mapped on the tangent space by applying the following operations:

$$\mathbf{m} = \mathrm{Vec}(\mathrm{logm}(\mathbf{M})), \tag{3.23}$$

where logm() is the matrix logarithm and Vec() the vectorization operator:

$$\mathbf{x} = \text{Vec}(\mathbf{X}) = \left[X_{11}, \sqrt{2}X_{12}, \ldots, \sqrt{2}X_{1d}, X_{22}, \sqrt{2}X_{23}, \ldots, X_{dd}\right]. \tag{3.24}$$

To summarize, these operations lead to a transformation from a SPD matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ to a vector $\mathbf{m} \in \mathbb{R}^{\frac{d(d+1)}{2}}$. To illustrate the SO-TCK framework, Figure 3.18 shows the extension of TCK approach, illustrated in Figure 3.15 to the use of second-order statistics. The subset $\mathbf{X}_1$ is constituted of N_1 time series (orange, blue and black) of length T_1 and described by V_1 attributes. Then, for each multivariate time series of the subset, a covariance matrix trajectory of size $V_q \times V_q \times (T_q - \Delta t + 1)$ is obtained following the explanations given in section 3.3.2.2. As the set of SPD-MTS $\mathbf{M}_1^{(n)}$ lies on a Riemannian manifold, they are projected on the tangent plane at the identity matrix $\mathbf{I_d}$ and vectorized according to (3.23) to obtain the log-Euclidean representation set of time series $\mathbf{m}_1^{(n)}$ of size $N_q \times \frac{V_q(V_q+1)}{2} \times (T_q - \Delta t + 1)$. Finally, similarly to the TCK approach, a codebook is created by learning a GMM model to estimate weights ω_g, means μ_g and variances σ_g for each multivariate time series $\mathbf{m}_1^{(n)}$ of the subset. Those GMM parameters are then used to encode each time series and compute posterior assignments Π_1.

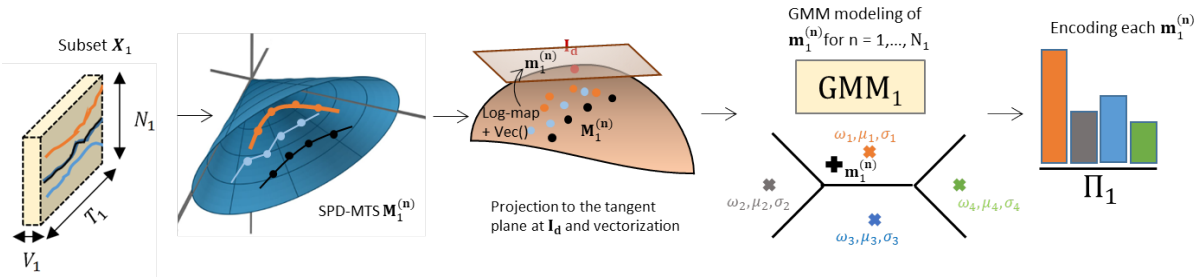


Figure 3.18: Diagram of SO-TCK steps for a single subset ($q = 1$).

3.4.2.2 Parameters selection

Figure 3.19 draws the evolution of the overall accuracy on the Libras dataset as a function of the temporal support Δt for the proposed SO-TCK approach. Three kind of second-order descriptors are considered:

- the covariance matrix (in red),
- the second-order moment ($\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$, in blue),
- and the full local Gaussian descriptor defined in (3.14) (Augmented SPD matrix in green).

As observed, the best results are obtained for this latter. In the following, SO-TCK will refer to the classification performance obtained with this full local Gaussian descriptor. Note also that the choice of Δt reflects a trade-off. It should be small enough to preserve a good temporal localisation, and at the same time sufficiently large for estimation purpose.

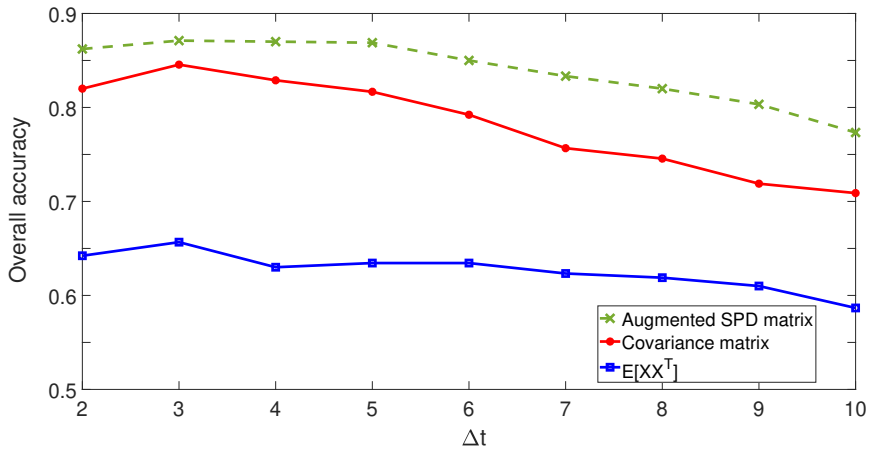


Figure 3.19: Classification comparison between three considered second-order statistics for SPD-MTS computation ($V_q = 2$).

Furthermore, Table 3.5 compares the classification performance between the original TCK and the proposed SO-TCK for the Libras dataset. As shown, the benefit of exploiting second-order statistics is clearly demonstrated where a gain of 15% is obtained when using the SO-TCK strategy.

Methods	**OA** (%)
TCK	72.6 ± 2.9
SO-TCK	**87.1 ± 2.8**

Table 3.5: Classification comparison between TCK and SO-TCK on the Libras dataset.

3.5 Experiments

In this section, we illustrate the potential of the proposed approaches to multivariate time series classification on four publicly available well-known datasets from UCI/UCR machine learning repository [Dua & Graff 2017], which constitutes the largest repository of time series datasets, and one on a remote sensing application.

3.5.1 Datasets of experiment

- **Libras:**

The dataset, acronym of the Portuguese name "Lingua BRAsileira de Sinais, represents hand movement of Brazilian official language, it contains 15 classes of 24 instances each, where each class references to a hand movement type. The hand movement is represented as a bi-dimensional curve performed by the hand in a period of time recorded by a video.

- **Natops:**

The Naval Air Training and Operating Procedures Standardization (NATOPS) is a manual of aircraft handling signals: the gestures most often used in routine practice on the deck environment. The data is collected using sensors on the hands, elbows, wrists and thumbs. Coordinates x, y, z for each of the eight locations are stored resulting in 400 samples for each gesture class. Figure 3.20 illustrates gestures of the NATOPS dataset.

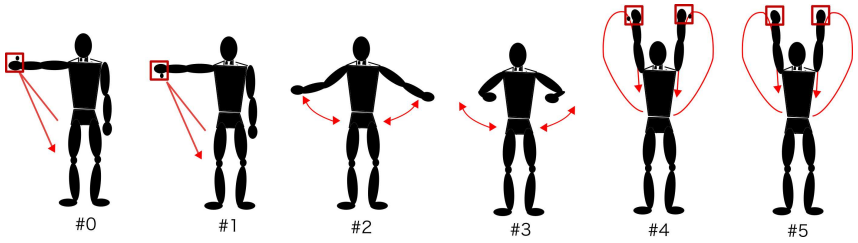


Figure 3.20: Gestures from the NATOPS dataset [Ribeiro *et al.* 2016].

- **Character trajectory:**

The dataset contain labelled samples of pen tip trajectories recorded whilst writing individual characters by the same writer [Williams *et al.* 2006]. Each character sample is a 3-dimensional pen tip velocity trajectory: x, y, and pen tip force as illustrated in Figure 3.21.

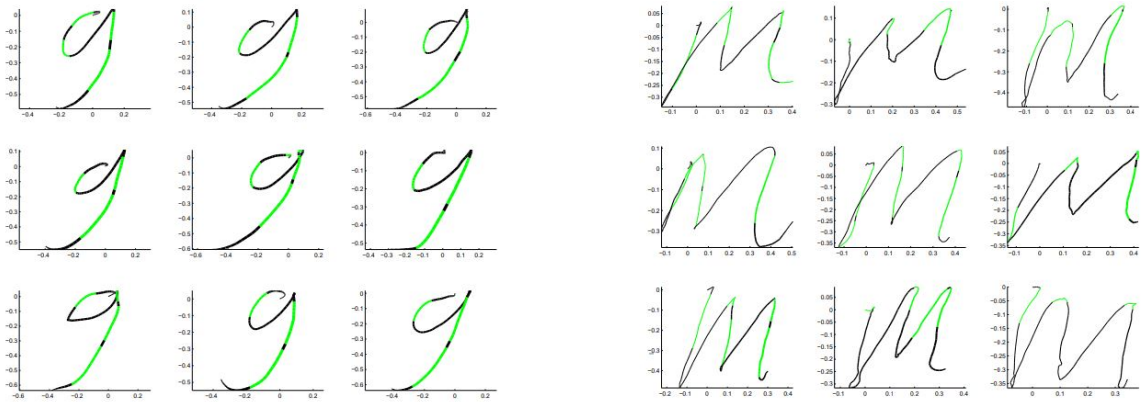


Figure 3.21: Examples of two classes from the character trajectories of 'g' and 'm' and the pen tip force highlighted with green color.

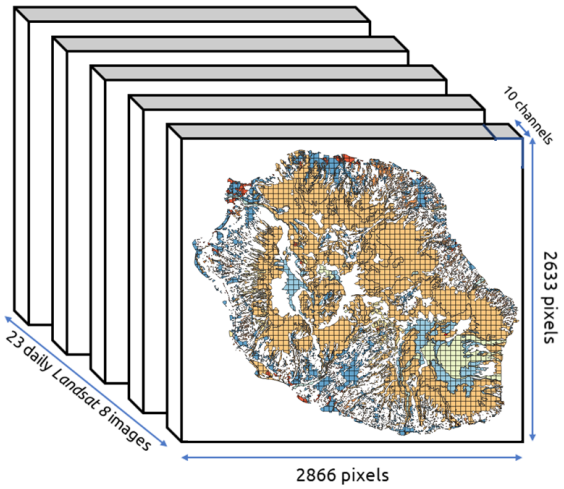
- **Racket sports:**

The dataset was created by university students playing badminton or squash while wearing a smart watch that sends x, y, z coordinates for both the gyroscope and accelerometer to an android phone. The considered four classes are either a forehand/backhand in squash or a clear/smash in badminton.

- **Tiselac:**

The Tiselac dataset, introduced in [Ienco 2017], has been generated from an annual time series of 23 Landsat 8 images acquired in 2014 above the Reunion Island. A total of 10 attributes (7 surface reflectances and 3 vegetation/water indices) are considered for each pixel at each timestamp. For this dataset, the goal is to predict 9 land cover classes, namely:

1. urban areas
2. other built-up surfaces
3. forests
4. sparse vegetation
5. rocks and bare soil
6. grassland
7. sugarcane crops
8. other crops
9. water



To ensure classes homogeneity and speed up calculations, the retained number of pixels of each classes is reduced to 200 pixels.

Table 3.6 gives the main characteristics of these datasets. In the following, performance are measured in terms of mean overall accuracy evaluated over 5 runs.

Datasets	Attributes V	Time length T	Samples for training	Samples for testing	Classes
Libras	2	23	180	180	15
Natops	3	51	180	180	6
Character traj.	3	23	300	2 558	20
Racket sports	6	30	151	152	4
Tiselac	10	23	900	900	9

Table 3.6: Characteristics of the experimented time series datasets.

3.5.2 Classification results

A classification comparison is conducted over five datasets. First, in the same spirit than experiments established in Table 3.4, where different configurations are tested to demonstrate the benefit of aligning series with DTW, using a re-parameterization invariant representation SRVF as well as exploiting second-order statistics through TSRVF. Also, the TCK and its extension to second-order statistics SO-TCK are compared. Results in Table 3.7 make the same comparisons for all datasets of interest.

[2]The reported results with the use of second-order statistics are given for the covariance matrix descriptor. In fact, the conducted experiments on the Tiselac dataset has demonstrated a benefit of using the covariance matrix instead of the full Gaussian descriptor.

Methods	Libras	Natops	Character trajectory	Racket sports	Tiselac[2]
Euclidean distance	79.2 ± 1.1	77.8 ± 3.4	**95.5 ± 0.4**	68.8 ± 2.6	60.4 ± 1.0
Warping + Euclidean distance	85.6 ± 3.8	71.7 ± 4.3	**95.6 ± 0.2**	81.3 ± 2.9	62.8 ± 2.9
SRVF + Euclidean distance	82.8 ± 2.3	**80.5 ± 2.4**	91.9 ± 2.9	63.7 ± 2.7	70.4 ± 1.3
SRVF + Warping + Euclidean distance	82.8 ± 2.3	**80.5 ± 2.4**	91.9 ± 0.2	63.8 ± 2.8	70.2 ± 1.2
TCK	72.6 ± 2.9	61.4 ± 3.5	91.7 ± 0.5	81.5 ± 3.2	63.7 ± 1.3
TSRVF + Euclidean distance	80.1 ± 3.0	76.2 ± 2.3	94.7 ± 0.4	91.1 ± 1.7	89.3 ± 0.5
TSRVF + Warping + Euclidean distance	**88.5 ± 2.7**	75.0 ± 2.7	93.2 ± 0.4	**94.7 ± 1.2**	**92.7 ± 0.4**
SO-TCK	87.1 ± 2.2	71.3 ± 3.7	93.9 ± 0.8	87.8 ± 2.8	74.5 ± 0.9

Table 3.7: Classification performance for different combinations involving the use of Euclidean distance, the dynamic time warping (DTW) and the first and second-order statistics through the SRVF and the TSRVF representations of SPD matrix trajectories, respectively.

As observed, second-order based methods perform better than first-order based strategies. For instance, the proposed SO-TCK is compared with the state-of-the-art approaches, including the original version TCK [Mikalsen *et al.* 2018]. We have adopted the same experimental setup as the one used in [Mikalsen *et al.* 2018]. For reproducibility purpose, TCK has been launched with the authors Matlab implementation. As observed, the best results are obtained for the proposed SO-TCK approach with a gain of 2 to 15 % compared to TCK, hence illustrating its potential for various application on time series classification.

In addition, the potential of warping series and the use TSRVF representation is clearly demonstrated where higher results are obtained in most cases compared to other strategies. Also, as far as the knowledge of authors are concerned, this is the first time, the use of TSRVF and SRVF representations are performed in the context of remote sensing monitoring (Tiselac) where, the combination of second-order statistics through the TSRVF representation and the warping technique, has shown a significant gain compared to other strategies.

3.6 Conclusion

This chapter assesses the problem of time series classification. The first part reviews different methods dedicated to this problem including machine learning and deep learning based methods. Since second-order statistics demonstrated a great interest in many applications, two methods were investigated and extended to second-order statistics. Since then, this chapter exploit the benefit of considering dependencies between multivariate time series attributes and their potential for enhancing classification performance.

In the context of similarity measurement between time series, the focus is on distance-

based methods to deal with the problems related to time series where the most commonly used distance is the Euclidean distance. Therefore, the measure does not provide satisfactory results when there is a time shifting between series of interest. For that, dynamic time warping (DTW) approach permits aligning series before comparing them. By doing that, corresponding points are matched regardless the differences of the speed profile between the series. However, despite ensuring speed invariance, the induced distance between series still have some limitations. In fact, the re-parameterization of the series does not preserve the distance between them. To overcome this, the square-root velocity function (SRVF) representation is introduced. For each time series, its corresponding SRVF representation is computed, then the whole set is aligned using the DTW algorithm to find the optimal warping path. Furthermore, to extend this framework to second-order statistics, the transported square-root velocity function (TSRVF) is required. It involves first the computation of SPD matrix time series (SPD-MTS) and then the representation of the trajectories on a manifold as a vector field by considering multiple tangent spaces at different reference points, which allows us to use existing algorithms efficiently, while also respecting the geometric and temporal constraints. The efficiency of the strategy is evaluated in different benchmark time series datasets and it has shown a significant gain compared to other state-of-the art strategies.

Future works in this context may aim at investigating the potential of different DTW variants, such as the constrained DTW [Zhang *et al.* 2017], the derivative DTW (DDTW) [Keogh & Pazzani 2001] or the weighted DTW [Jeong *et al.* 2011], to outperform the original DTW algorithm. In fact, DTWs variants have been intensively evaluated to demonstrate their interest. As such, they can be used to replace the original version in aligning second-order trajectories of TSRFV representations.

In addition, inspired by the time series cluster kernel (TCK) and the potential of second-order statistical descriptors for many classification tasks, this work has extended the formalism of TCK to second-order features. For that, the log-Euclidean metric has been considered to represent a SPD-MTS as a multivariate time series where the principle of TCK can be employed. Experimental results on benchmark datasets and land cover classification with remote sensing data have shown, most of the time, the potential of the proposed method compared to state-of-the-art times series classification algorithm.

To go further, the two proposed strategies can be combined in order to align time series before applying the proposed SO-TCK approach. This requires the use of time series matching techniques extended to multiple alignment problem. For doing that, the given trajectories can be used to define a template trajectory and then align each given trajectory to this template in a pairwise fashion. One way of defining this template is to use the mean of given trajectories under an appropriately chosen metric. For example, based on the original version of TSRVF, the mean trajectory can be computed by using a Karcher mean algorithm [Su *et al.* 2014b].

Moreover, many works focuses on neural network strategies to deal with time series classification problems. For example, CNNs have been popular for time series classification through their ability to capture spatial and temporal patterns using trainable filters. In the same spirit, one can exploit the power of CNN models to learn multiple discriminative features where a convolution can be seen as applying a filter over the time series. A perspective of

interest can be the development of a hybrid model, in the same spirit than those developed in chapter 2. For example, the CNN convolutional layer outputs, which represents a novel time series representation, would feed the proposed architectures such as the SO-TCK strategy.

Chapter 4

Forest health monitoring using Sentinel-1 and Sentinel-2 time series

Contents

4.1 Introduction

Forests are an integral part of natural ecosystems where they cover a large part of the continental surface, approximately 30% of the worlds total land mass. It is a source of renewable materials and energy and provides numerous ecological, economic, social, and cultural services [Boyd *et al.* 2013]. Also, it carries out major actions on our environment such as the provision of ecosystem services (conservation of biodiversity, regulation of climate change through the carbon cycle, etc.). However, environmental changes and global trade have increased forest vulnerability to a range of disturbances, including diseases and insects. Natural forest declines include those related to the action of climatic and biotic factors (damage caused by storms, drought, heat waves, insect attack etc.). In fact, climatic changes, such as temperature, will alter tree health, tree species distributions, and tree resistance. Simultaneously, changes in moisture and precipitation regimes, becoming drier in certain areas and wetter in others, will also cause a range expansion of insects and diseases, allowing organisms to occupy areas of forests cover. Some of these sylvosanitary problems have been known for a long time and reappear recurrently. Other problems are caused by exotic species and are serious threats due to the lack of co-evolution between these pathogens or pests and native forest species. Also, human, at the origin of many fires and deforestation, is not left behind.

Finally, the interactions between all those different factors are numerous and result in an increased forest vulnerability. For example, bark beetles cause significant damages after storms or fires and lead to mortality and die-off. In France, because of those different factors, damages are observed in the South-West (Dordogne) and also in other French regions (Centre, Île-de-France, Occitanie, etc.).

At the national level, the health of forests and grasslands is an issue of high concern. For that, different systems to monitor the vitality and health of forests exist. Specifically, the french ministry of agriculture and food provides forest monitoring implemented by the department of forest health (DSF)[1]. These missions are carried out by mobilizing correspondents-observers (CO), agents of the national forest office (ONF)[2], of the national forest ownership centre (CNPF)[3], and other organisms responsible for providing advice and guidance to private forest owners, and decentralized administration services.

In terms of monitoring the evolution of forest stands by remote sensing techniques, it remains difficult to detect less radical events such as silvicultural interventions (thinning, clearing, soil maintenance, afforestation) or damage caused by biotic or abiotic vagaries. Even with high spatial resolution, using only one or a few satellite observations is a barrier to fine detection and accurate dating of these events. As Kennedy et al. [Kennedy *et al.* 2014] pointed out, the Landsat time series can capture only a small portion of the ecological and functional processes involved in the forest cover dynamics. In order to be able to identify the forest health problems on the images, the first condition is to be able to characterize the seasonal leaf cycle for abnormalities identification, that is to say the differences in the seasonal cycle of healthy stands and infected stands.

[1] DSF: Département Santé des Forêts
[2] ONF: Office National des Forets
[3] CNPF: Centre National de la Propriété Forestière

Within this framework, the TEMPOSS project, which stands for "Modelling of the temporal trajectory of Sentinel-1 & 2 observations for monitoring forest health", focuses on the fine characterization of the structure of the forest and the monitoring of its changes, with a view to identify, map and follow sylvosanitary problems. It is conducted on several French sites, representing various types of forest cover (different species, pure or mixed, various environmental conditions, fragmentation or continuity of cover classes, etc.) and sylvosanitary contexts.

The project aims to model and classify the temporal trajectories of the signals measured by radar (Sentinel-1) and optical (Sentinel-2) sensors with a revisit frequency of several days. The results are expected at two levels: (1) the development of original algorithms to identify homogeneous settlement entities; (2) an initial assessment of their potential to derive indicators of forest health and its evolution.

This work is realized in close collaboration with the national focal point "Data, remote sensing and epidemiology" of the forest health department (DSF) who plays an important role as a forestry expert where he helps introducing the forest issues and supplies the ground truth data related to the studied problems. In addition, the initial results obtained by DSF using a random forest classification provide a starting point. Therefore, providing both a remote sensing monitoring problem and ground truth data, the TEMPOSS project defines an appropriate application framework for this thesis. So we worked on evaluating the potential of Sentinel-1 and Sentinel-2 for monitoring the forest health with a focus on a particular forest disease, namely the chestnut ink disease, affecting the Montmorency forest. For this end, this chapter proposes original algorithms exploiting second-order descriptors and it focuses on the following main points:

- An overview of diseases and insects attack that causes forest damages, especially in France. In collaboration with the department of forest health (DSF), more attention will be directed to chestnut disease caused by the pathogens named phytophtoras.
- As the objective is to exploit remote sensing data, this chapter focuses on evaluating the potential of the Sentinel-1 (C-band radar) and Sentinel-2 (optical) time series as well as a fusion scheme of optical and radar imagery for monitoring the symptoms of the phytophtoras attack on chestnut Montmorency forest (in France).

From a methodological point of view, the main developments concern the satellite data to be used, the selection of variables and indices extracted from Sentinel-1 and Sentinel-2 images, the choice of classification and regression algorithms and their parameterization. To summarize, the main contributions are:

- Identification of discriminating **attributes from Sentinel-1 and Sentinel-2**. It consists of selecting the remote sensing variables of interest by **extracting various effective vegetation indices related to forest health issues**.
- Since second-order statistics manifested a great potential on different classification applications, they are exploited in the context of forest health monitoring to improve accuracy of discriminating healthy from damaged classes. For that, **a covariance pooling framework is proposed to exploit dependencies between different attributes, such as spectral bands and vegetation indices for optical data, and backscattering coefficients and forest degradation indices for radar data**.

- Furthermore, in accordance to previous chapters, considering ensemble strategies may be useful for enhancing algorithm robustness. In particular, in such challenging applications related to forest health monitoring. For that, we proposed to **extend the covariance pooling strategy to an ensemble approach where several features of attributes were be considered to fed each ensemble branch.** At the end, a majority vote is applied to elect the most relevant decision.

- For the sake of improvement, as the disease evolves continuously from healthy stands to completely destroyed trees, **a forest health indicator is defined and a regression model based on covariance pooling is proposed.** As such, the regression model permits to predict a quantitative variable related to the forest degradation status.

The chapter is organized as follows. Section 4.2 assesses a brief review of some selected forest diseases caused by insect attacks and the role of remote sensing data, whether optical or radar, in quantifying and monitoring the forest health related issues. Section 4.3 introduces Sentinel optical and radar imagery and gives an overview of their properties. Also, a brief review of their use in different land cover and land use applications is exposed. Then, section 4.4 addresses the targeted application, that is the chestnut ink disease, including explanations about the studied area and the dataset of interest. After that, section 4.5 investigates the potential of using second-order features on an ensemble strategy while using Sentinel-1 and Sentinel-2 images. Finally, Section 4.6 gives the main conclusions and perspectives for this work.

4.2 Remote sensing for forest health monitoring

In general, forest diseases are caused by pathogens that are infectious and transmissible, such as bacteria, fungi and viruses. Besides, insects attack different parts of the tree, with defoliators feeding on leaves or needles, and bark borers.

Most of the time, the only alternative to limit the spread of an epidemic is the rapid removal of wood. For that, exceptional operations of cutting are carried out and supervised by the National Forest Office (ONF) in public forests. For the most impacted stands, clear cuts can be considered, while ensuring the preservation of soils, memorial sites and natural areas.

4.2.1 Forest diseases

Over the past few decades, the frequency and intensity of disease and forest disturbances due to insect attacks have dramatically increased, leading to extensive tree mortality worldwide. Examples include the oak death epidemic in western United States, outbreaks of mountain pine beetle in Canada's boreal forest, bronze bug damage in forest plantations in South Africa, and the spread of bark beetles in central Europe and Scandinavia [Kelly & Meentemeyer 2002, Fassnacht *et al.* 2014, Oumar & Mutanga 2014]. In the following, some of the well-known forest disturbances in France caused by insects or diseases are described.

4.2.1.1 Bark beetles attack

Large forest losses occurred in the past due to bark beetle attacks. Bark beetles are small wood-eating insects, ranging in length from 2 to 7 mm, black or brown in colour, and belonging to the order of Coleoptera. The larvae of this beetle hatch under the bark, they dig galleries as

illustrated in the left side image of Figure 4.1. The larval stage lasts 8 to 10 weeks, young adults feed under the bark. In fact, they are beneficial insects for forest regeneration, because they usually feed on dead wood, allowing to accelerate their decomposition. However, under certain conditions, they attack healthy trees; either because the populations of bark beetles are too large to be satisfied only with dead wood, or because the trees are weakened (water stress due to drought, mutilations following a storm, nutritional deficiency linked to depleted soil, etc.). Also, bark beetles normally reproduce twice a year but recently, they were able to reproduce three times due to the climate changes (hot and dry weather).

Figure 4.1: Bark beetles attack. Left: larval galleries dug by oak bark beetles. Right: Trees turned red because of the attack[4].

Bark beetle attacks result in the disturbance of tree physiological function, eventually leading to loss of moisture from the canopy foliage and a change in color from green to red, as shown in the right-sided image of Figure 4.1, then comes the gray stage before completely dying. In France, bark beetle attacks are particularly feared in forestry where they cause serious damages in pine forests. It initially started in the "Grand Est" region. Now, the beetle epidemic covered almost all spruce forests in the northern half of France.

4.2.1.2 Forest chafer crisis

This beetle is a plague for the forests with sandy soils. In France, these species are naturally found in the forests of the Oise region. The evolution of a generation takes place over 5 years. The beetle lays its larvae in the soil and for four years they feed by nibbling the roots of plants and trees until causing their mortality as shown in Figure 4.2. When the insect takes off, it devours the leaves of trees, mainly oak. According to foresters, beyond a density of 10 larvae per square meter, the situation becomes irreversible.

These are mainly the root consumption of the larvae of these widespread insects that cause damages, leading to plants mortality and seedlings of all species.

[4] http://ephytia.inra.fr/fr/C/21221/Forets-Scolyte-du-chene-ou-scolyte-intrique
[5] http://ephytia.inra.fr/fr/C/20318/Forets-Hanneton-forestier

Figure 4.2: Mortality of young conifers that have suffered root attacks from forest chafer larvae[5].

In France, the combination of the chafer beetle crisis and climate change is of great impact for forests [Nageleisen *et al.* 2015]. As the species live in soil, the beetles are not visible and observed symptoms associated with root consumption are more tenuous, they consist of dwarf shoots, leaf wilts and stem mortalities. These symptoms may be confused with a drought problems.

4.2.1.3 Chesnut ink disease

Chestnut is the third broad-leaved species in France, after oak and beech. Although difficult to estimate because it is often present in a mixture with other species, chestnut trees covers more than one million hectares. Over the past few years, a great deal of progress has been made in developing dynamic chestnut forestry to meet the growing demand of wood. However, in many regions, the health status of chestnut forests has deteriorated over the past decade [Goudet 2016]. In 2005, 60% of chestnut groves were dying, and 40% of mortality was attributed to this decline. Forest wasting is often the result of several predisposing, triggering and aggravating factors. For the chestnut trees, these factors can be multiple: the disease of the canker, repeated droughts, human silvicultural interventions, etc. However, root infections with primary pathogens can also cause loss of vitality and mortality on their own or predispose chestnut trees to decay. Due to the increased intercontinental trade and climate change, pathogen invasion are increasing exponentially in Europe [Santini *et al.* 2013]. These pathogens, such as phytophtoras, cause emerging diseases that pose a significant threat to the health of all ecosystems.

Phytophthora [Zentmyer 1988] are eukaryotic microorganisms and are root necrosis pathogens of a very large number of plants. Most species live and multiply in the soil and infect the roots. Phytophthora, belonging to the Oomycete group, are native to Asia and are one of the hundred most invasive species recorded worldwide. They are characterized by their sensitivity to temperature. On chestnut, it causes the disease of the ink, whose reports are in very strong increase over the last ten years [Saintonge 2003]. Two species of Phytophthora have been found to be responsible for ink disease in Europe, namely P. cambivora and P. cinnamomi [Vannini *et al.* 2001]. The disease is thus present in western France, in regions under oceanic influence. Further east, it is limited by harsh winters and it is absent in the chestnut groves of Alsace and the Alps.

The two species of Phytophthora produce the same symptomas: the destruction of all or part of the root system leading to the degradation of its crown, or even to the death of the tree. As illustrated in the left side image of Figure 4.3, the disease progresses by staining from the first infected trees from near to near. Long-range dissemination is possible through the transport of infected soil and plants.

Figure 4.3: Phytophtora attack in oak tree. Left: Flame necrosis surrounded by a black border. Right: Chestnut mortality in Montmorency forest[6].

In the field, diagnosis is always difficult. In oaks, flame necrosis are quite characteristic as shown in left side of Figure 4.3. But this phenomenon is less common in the chestnut tree since their symptoms are not enough characteristic and discriminating which results on underestimated presence in France of phytophthoras. In fact, for a more accurate diagnosis, it is necessary to carry out a sampling for a laboratory examination.

As seen on these examples, the increasing problems related to forest health necessitate new tools for quantifying, measuring and analyzing the forest areas to bring a comprehensive monitoring system of forest ecosystems, their state and changes regarding different spatial, temporal and meteorological modifications. The use of satellite data for Earth observation is a good candidate for forest monitoring as shown by the increasing focus on remote sensing imagery for forest applications.

4.2.2 Remote sensing techniques for forest health monitoring

The two main types of satellite data used in remote sensing are optical and synthetic aperture radar (SAR) images. Optical satellite imagery provides a rich spectral information. However, since optical sensors measure reflected sun light, they can't penetrate through clouds. In contrast, radar sensors can acquire images at both day and night and in almost all weather conditions. Moreover, radar images can reveal details of the land cover that are not visible to the human eye, such as soil moisture or inundated vegetation, marine pollution, or forest biomass.

As illustrated in Figure 4.4, one can distinguish between "passive" and "active" remote sensing systems because the processing and analysis for the study of forests are completely different. The optical sensors, are dependent on sunlight and on the atmosphere. The

[6] http://ephytia.inra.fr/fr/C/20253/Forets-Encre-du-chataignier

atmosphere conditions therefore compromises temporal stability and spatial coherence of optical measurements. Active instruments emit their own signal and are able to measure the reflected part of the target area. In addition, they use wavelength longer than the one of the optical sensors, allowing measuring different characteristics of the imaged area.

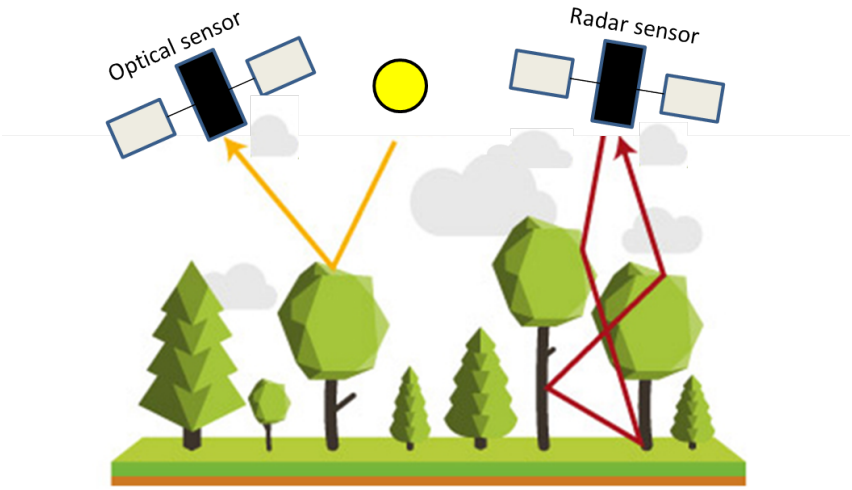


Figure 4.4: Illustration of the sensitivity of optical and radar signals to different tree elements.

For radar sensors, depending on the frequency and therefore the wavelength used (about 2 to 30 cm), the sensor will be sensitive to certain elements on the Earths surface. Long wavelengths (P and L bands) can penetrate the vegetation, while shorter wavelengths (X band) are more sensitive to the roughness of the soil surface and are reflected from the tops of trees. Also, SAR images provide information on geometric and dielectric properties of the surface or studied volume, depending essentially on the surface roughness, the type of material and its moisture content. Specifically, a rough surface will have a higher feedback signal than a flat and smooth surface (which will return almost no signal), a high volume of vegetation will return a stronger signal than a low volume, and wet vegetation will give more signal than the same dry vegetation. For forests, the backscatter signal is therefore not directly related to the trees from a physical point of view, but it will be influenced by the structure of the canopy and the spatial variability. The signal penetrates the canopy according to its wavelength. Shorter wavelengths (X and C bands) are sensitive to small elements such as leaves and small branches, while the larger wavelengths (P and L bands) are sensitive to large branches and trunks.

Optical imaging, on the other hand, is sensitive to the pigments of vegetation linked to photosynthesis, in the Red, Green and Blue bands, and to the cellular structure of the leaves in the near-infrared band. Thus optical and radar remote sensing provide different but complementary information to estimate general forest status such as crown height, density and shape, tree basal area, timber volume and biomass.

Sustainable forest management is essential to alleviate the destructive impacts of diseases or insects on forest ecosystems. Recently, remote sensing images, either optical data or SAR imagery, has generated much interest for forest applications. To use remote sensing images, damaged trees needs to show distinct symptoms which can be observed remotely. As such, and depending on damage type or stage, the symptoms may indicate the decline in

chlorophyll/water quantity in foliage, leaf discoloration, defoliation, or tree-fall gaps. Sensors are then expected to capture the differences between damaged and healthy trees. Moreover, the acquired data may also offer the ability of monitoring different damage stages from the healthy stage to the destroyed tree stage.

4.2.2.1 Remote sensing data characteristics

Remote sensing data proved their potential for monitoring forest disturbances through their qualifications based on:

- **Spectral characteristics:**

Multispectral imagery has proven its potential to asses the status of damaged trees [Meddens *et al.* 2011]. In fact, remote sensors have the ability to record the visible portion of the electromagnetic spectrum (wavelengths from approximately 400 to 700 nm) and are capable of detecting disease symptoms where the spectral values in a forest images can be linked to the forest health. Because of the disease or insects, the tree ability to photosynthesize is affected, which results on visual change in foliage color, also called discoloration. For example, as described previously, pine trees turn red in the red-attack stage by bark beetles and oak trees visually appear brown as a result of the sudden oak death. However, since the disease or the insect attack is a gradual process, some symptoms cannot be easily observed. In contrast, sensors with the capacity of recording the near-infra-red spectrum (wavelengths from approximately 700 to 1300 nm) could be more sensitive to such delicate changes. To further improve performance of remote detection, researchers tried to combine different spectral bands to produce a variety of spectral indices.

- **Spatial characteristics:**

Recent development in remote sensing allows us to perceive spatial details on the Earth's surface at varying scales. For example, the MODIS sensors allow to acquire images with a spatial resolution of 1 km, 500 m and 250 m where SPOT-7 data shows a resolution of 6 m and 1.5 m and Landsat a resolution of 30 m and 15 m. The increase of damaged trees within small and discrete patches arises the challenges and the need of high spatial resolution images. In fact, considering a high-resolution pixel will allow covering a portion of a tree, and the corresponding pixel value may contain a high spectral variation as a result of the complex forest structure.

- **Temporal characteristics:**

Since the progression of a disease or insect damages have a duration in time, the temporal characteristics allow to consider trajectories of disease and insect propagation over a long time. As such, temporal imagery offers a characterization of forest change which attracts the interest of many researchers. Increasing application of diverse data archives for long-term forest health are widely used, in particular, the Landsat and Sentinel time series, which offers minimized temporal gaps, global coverage and provide images for free.

4.2.2.2 Forest health monitoring using remote sensing data

Many advanced classification methods were already developed. These methods are based on both pixel and objects. The main objective of the following is to provide a brief survey of remote sensing methods to detect forest damages by diseases and insects.

- **Classification**

First, classification using satellite imagery was developed to differentiate between land-cover types. The measurement of forest damage is more challenging where the focus would be on disease symptoms. As seen previously, compared to healthy trees, damaged trees have distinct symptoms, such as reduced moisture, discolored foliage, and defoliated canopy. To make the parallel with classical algorithms for land cover classification, the distinct symptoms of infected forests are considered as land-cover types.

The classic maximum-likelihood classifier demonstrated a great success on detecting forest disease. It was applied to Landsat images in [Walter & Platt 2013] for differentiating mountain pine beetle red attacks from non-red attacks. As a complement to classic classifiers, machine learning methods have been introduced to the domain of remote sensing classification since the 1990s. For example, to monitor the changes in chlorophyll concentration using the red edge, which is the inflection point that occurs in the rapid transition between red and near infra-red reflectance, support vector machines (SVMs) have proven to be successful. [Adelabu *et al.* 2014] used red edge bands of RapidEye images for detecting three levels of insect defoliation ranging from healthy to defoliated plants under severe defoliation in an African savanna. In addition, [Adelabu *et al.* 2014] have studied the use of random forest classification for insect defoliation levels which gives comparable results with SVMs. In fact, one of the key points of random forest algorithms is their capacity to rank variables according to their importance which allows identifying the most discriminating spectral bands or indices involved in disease mapping.

Despite being useful for the purpose of detecting changes in forest health, especially using near-infra-red channels, optical data are sensitive to weather conditions, which results on complications to acquire data in cloudy regions. To overcome this limitation, many researchers focus on combining optical with radar data. In fact, the SAR systems are not only capable of providing images regardless the weather conditions, but also have the ability to penetrate the forest canopy to some degree. As such, microwave and optical data offer complementary information that can improve the classification accuracy. Since then, this point become an important focus of remote sensing research. For example, [Ortiz *et al.* 2013] combined images from both sensors, optical from RapidEye and radar from TerraSAR-X sensors, for monitoring different levels of discoloration of bark beetle infestation.

- **Regression**

Regression analysis allows practitioners to estimate continuous defoliation or tree mortality levels, from healthy to damaged. For example, a regression model was applied to estimate an outbreak of black-headed budworm in Western Newfoundland, in Canada [Luther *et al.* 1997]. In addition, in order to understand continuous tree damage levels, some researches [de Beurs & Townsend 2008] exploit multiple linear regression to link attributes (such as spectral bands and spectral indices) with measured damage indicators, such as defoliation intensity. A large range of remote sensing data types (MODIS, Landsat, Lidar, etc.) were used. With the development of sensors ability to provide high spectral resolution, the dimensionality of data increases and causes many computation challenges. To overcome this issue, [Verbesselt *et al.* 2009] proposed a regression model able to select the best performing variables out of a large amount of variables.

4.3 Sentinel remote sensing data

Sentinel missions are jointly implemented by the EC (European Commission) and ESA (European Space Agency) for global land observation (data on vegetation, soil and water cover for land, inland waterways and coastal areas, and also provide atmospheric absorption and distortion data corrections) at high resolution with high revisit capability. It constitutes an enhanced continuity of data provided by SPOT-5 and Landsat-7. In addition, data are available for free which make them more attractive for many challenging applications.

4.3.1 Optical data: Sentinel-2

The Sentinel-2 mission is an operational mission from the European Space Agency (ESA), based on two satellites launched respectively in 2015 and 2017. Sentinel-2 acquires high resolution images (10 to 60 m depending on the spectral band). The orbital repeat cycle is 10 days and 2 satellites were placed on that orbit with a 180° angular distance: the two satellites can therefore achieve together a 5 days revisit period.

Compared to other satellite images, Sentinel-2 incorporates three new spectral bands in the red-edge field for the study of vegetation. They are important for the recovery and monitoring of important biophysical parameters such as indicators of vegetation health, the structural and functional parameters of the vegetation cover and the estimation of the biomass.

The Sentinel-2 products are available with three types of post processing, namely level-1C, level-2A and level-3A. The characteristics of each Sentinel-2 product types are summarized in Table 4.1. The products are obtained via the MAJA processor [Hagolle *et al.* 2017] which is a joint algorithm developed by CNES, CESBIO and DLR for cloud detection and atmospheric correction. Pre-processing of Sentinel-2 images is performed using Sen2Cor. It is a prototype processor for tasks of atmospheric, terrain and Level-1C data is surface reflectance measured at the top of the atmosphere. Level-1C data processed with Sen2Cor algorithm allows to obtain Level-2A products, the bottom-of-atmosphere reflectance. Level-2A data is the most ideal for research activities as it allows further analysis without applying additional atmospheric corrections. Furthermore, level-3A provides cloud-free images based on data acquired over a longer period of time. In fact, the Weighted Average Synthesis Processor (WASP) developed by Theia produces monthly summaries of Sentinel-2. For each pixel, and each spectral band, WASP averages the surface reflectances observed in clear skies over a 45-day period. For example, the July 15 synthesis will average cloud-free observations collected between June 26 and August 5. And this is repeated every month.

Level	Description	Tile size	Resolution
1C	Top-of-atmosphere reflectance	100×100 km^2	10, 20, 60 m
2A	Bottom-of-atmosphere reflectance	100×100 km^2	10, 20, 60 m
3A	Monthly summary	100×100 km^2	10 m

Table 4.1: Sentinel-2 product types.

The images, also called tiles, are 100×100 km^2 ortho-images in UTM/WGS84 projection as illustrated in Figure 4.5.

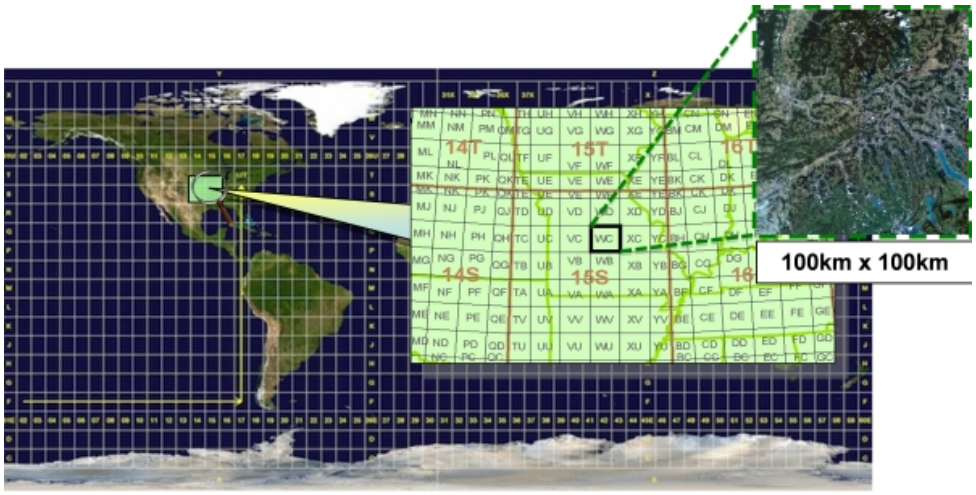


Figure 4.5: Level-1C product tiling[7].

With two satellites, Sentinel-2A and Sentinel-2B, areas are theoretically revisited every five days.

4.3.1.1 Spectral bands and product types

The Sentinel-2 thirteen spectral bands range from visible (VIS) to the short wave infra-red (SWIR) and are represented in Figure 4.6.

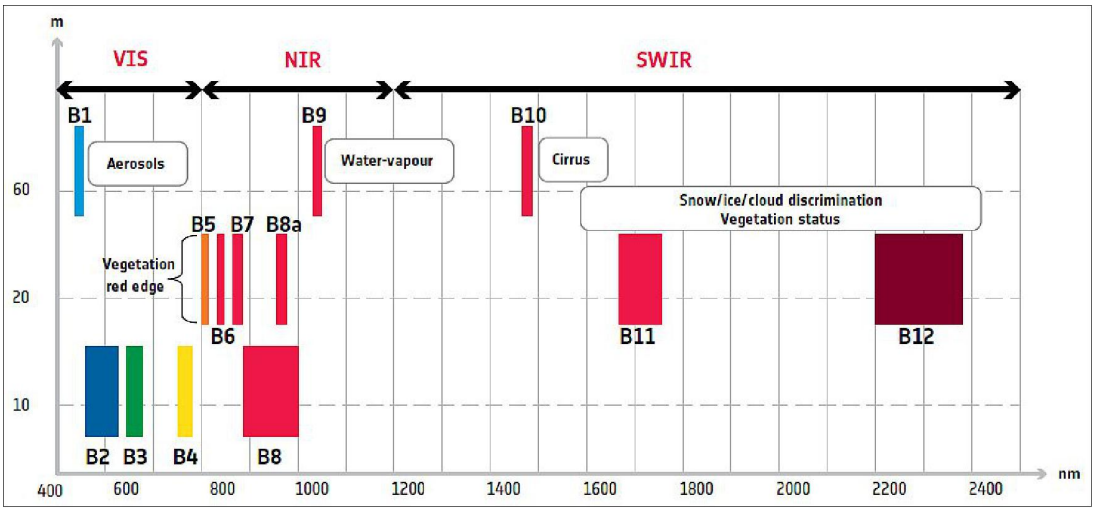


Figure 4.6: Sentinel-2 spectral bands [Bertini *et al.* 2012].

In summary, Sentinel-2 data are acquired on 13 spectral bands in the visual near-infra-red (VNIR) and short wave infra-red (SWIR):

- four bands at 10 m: 490 nm (B2), 560 nm (B3), 665 nm (B4), 842 nm (B8);
- six bands at 20 m: 705 nm (B5), 740 nm (B6), 783 nm (B7), 865 nm (B8a), 1 610 nm (B11), 2 190 nm (B12);
- three bands at 60 m: 443 nm (B1), 945 nm (B9) and 1 375 nm (B10).

[7] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/product-types

4.3.1.2 Derived vegetation indices

The absorption and reflectance of different light wavelengths by vegetation is a consequence of the cellular structure of the leaf. Healthy leaves absorb $70-90\%$ of incident visible radiation, particularly in the blue and red wavelengths (centered on 450 nm and 670 nm respectively), and reflect most of the green light (centered on 533 nm) which is the reason why leaves appear green to the human eye. A typical spectral reflectance curve is given in Figure 4.7.

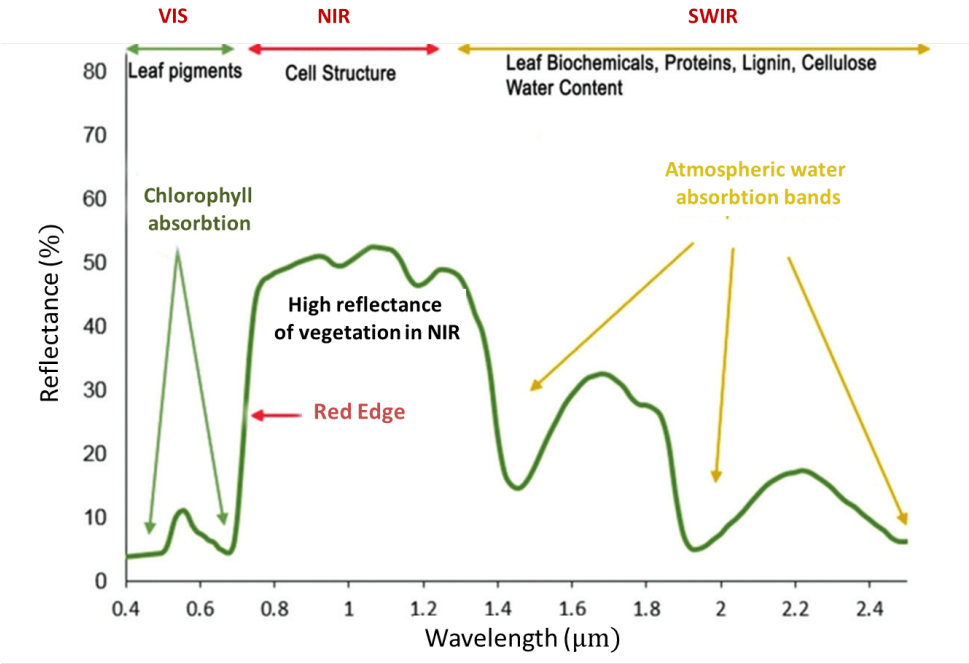
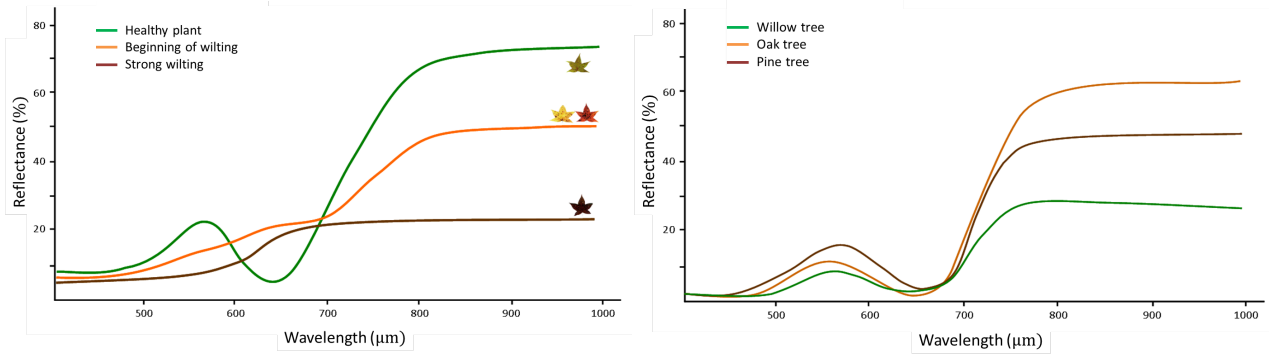


Figure 4.7: The spectral reflectance curve of vegetation [Roman & Ursu 2016].

The physiological changes that comes with the growth of a plant, from the maturation phase to its senescence, whether natural (phenological stages) or stress-related, strongly influence the vegetation spectral signature. Regarding natural changes, when autumn comes for example, plants decrease their photosynthetic activity, chlorophyll pigments disappear leaving the other foliar pigments to express their colors until the deconstruction of the cell layers. Consequently, there is a noticeable increase in reflectance in the longest wavelengths of the visible (yellow - red) and at the same time, a decrease in reflectance in the near-infra-red.

Regarding plant stress, it also considerably changes the spectral properties of vegetation, as the chlorophyll pigment rapidly decays and loses its absorption properties. Also, it results in leaf chlorosis: a yellowing discoloration due to chlorophyll losing dominance. Stressed plants have, therefore, a different spectral signature which can be observed in visible light and have a lower reflectance in the near-infra-red (NIR) region. In addition, as shown in Figure 4.8, reflectance varies not only according to state of health of a plant (a), but also according to the types of plants (b). The spectral signature of the vegetation in the visible does not vary according to the plant type, whereas in the near-infra-red, softwood trees, the pine tree for example, have a lower reflectance than broad leaves trees, such as oak trees.

[8] https://e-cours.univ-paris1.fr/modules/uved/envcal/html/vegetation/caracteristique-vegetation/proprietes.html.



(a) (b)

Figure 4.8: Spectral signature variations based on: (a) physiological vegetation state, (b) vegetation type[8].

Therefore, images captured in the visible spectrum can be visually interpreted, while the ones involving bands from the invisible wavelengths or their combinations with visible light require a mathematical transformations to make them interpretable.

In order to evaluate vegetation cover, vigor, and growth dynamics, many vegetation indices (VIs) have been proposed in the literature within remote sensing applications using different satellite sensors. They are quantitative measures and are based on the principle of combining spectral bands to highlight specific characteristics of vegetation regarding growth and health. The different extracted VIs rely on the spectral characteristics of vegetation and the different instruments and platforms to determine which solution is best to get a particular issue.

- The normalized difference vegetation index (NDVI):

With the use of high resolution spectral sensors, the number of bands obtained by remote sensing is increasing which allows the proposition of a large range of vegetation indices for many remote sensing applications. One of the most commonly used and implemented index is the Normalized Difference Vegetation Index (NDVI) [Kriegler *et al.* 1969]. It consists of a normalized ratio between the red and near-infra-red bands such as:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}}, \tag{4.1}$$

where NIR is the near-infra-red reflectance band and R is red reflectance band . It is based on the difference between the maximum absorption of radiation in red as a result of chlorophyll pigments and the maximum reflectance in NIR as a result of leaf cellular structure. Hence, NDVI characterizes canopy growth or vigor and allows the distinction of vegetation from the soil background. For vegetation, the range of NDVI values is between 0 and 1 with a sensitive response to green vegetation but remains also sensitive to the effects of soil brightness, soil color, atmosphere, cloud and cloud shadow. It also takes negative values between -1 and 0 in case of clouds, water and snow.

- The ratio vegetation index (RVI):

Jordan [Jordan 1969] proposed in 1969 the ratio vegetation index (RVI), which is based on the principle that leaves absorb relatively more red than infra-red light. The RVI is given by:

$$\mathrm{RVI} = \frac{\mathrm{NIR}}{\mathrm{R}} \tag{4.2}$$

Dense vegetation will generate high values of RVI, soil will lead to values close to 1, whereas clouds, water and snow, which have high reflectance in the visible than the near-infrared, will give values lower than 1.

- The brightness index (BI):

The brightness index permits to estimate whether the observed surface element is light or dark. This index is therefore sensitive to the gloss of the soil, related to its moisture and the presence of salts on the surface. It is also computed from the red R and near-infra-red NIR channels as follows:

$$\mathrm{BI} = \sqrt{\mathrm{R}^2 + \mathrm{NIR}^2} \tag{4.3}$$

- Normalized Difference Vegetation Indices involving red-edge bands (NDVIre2, NDII and NBR)

The potential of the red-edge bands in detecting vegetation stress, forest disturbance and moisture content changes has promoted the development of many red-edge related modified vegetation indices [Cheret *et al.* 2018], such as the NDVIre2, which has proven its sensitivity to detecting changes in the canopy [Potter *et al.* 2012], the Normalized Difference Infrared Index NDII [Ji *et al.* 2011] and the Noise Burn Ratio NBR [Keeley 2009]. They are computed using the near infrared NIR and middle infra-red MIR bands. According to Figure 4.6, it gives:

$$\mathrm{NDVIre2} = \frac{\mathrm{B8a} - \mathrm{B6}}{\mathrm{B8a} + \mathrm{B6}} \tag{4.4}$$

$$\mathrm{NDVII} = \frac{\mathrm{B8} - \mathrm{B11}}{\mathrm{B8} + \mathrm{B11}} \tag{4.5}$$

$$\mathrm{NBR} = \frac{\mathrm{B8} - \mathrm{B12}}{\mathrm{B8} + \mathrm{B12}} \tag{4.6}$$

- Continuum removal-Short waved infra-red (CR-SWIR)

This index was recently introduced by Raphael Dutrieux *et. al* [Dutrieux *et al.* 2021b], in the context of monitoring forest health, especially for bark beetle infestation detection. It uses near and medium infra-red bands and is sensitive to the water content of vegetation. When the vegetation is decaying, the index is high. Conversely, when the vegetation is healthy, the index will be low. For more information, the interested reader is referred to [Dutrieux *et al.* 2021b].

In addition to optical data, Synthetic Aperture Radar (SAR) images are generally preferred when meteorological conditions are not suitable to acquire cloud-free optical images.

4.3.2 Radar data: Sentinel-1

SAR images has attracted a lot of attention and has been investigated in several studies proving their effectiveness in many land cover monitoring applications.

In fact, in contrast to optical sensors, a radar is an active system. It permits to illuminate the Earth surface using microwave energy and measuring the reflected signal. Then, the

elapsed time and energy of the return pulse are recorderd by the antenna. Therefore, images can be acquired day or night, completely independent of the solar lighting. In addition, the radar signal penetrates the clouds without difficulty, allowing to acquire images regardless of the weather conditions. Once the radar has emitted a microwave signal, the power with which an object reflects the signal is measured. This is called backscatter. The amplitude and phase of the backscattered signal depends on the physical (i.e., geometry, roughness) and electrical properties (i.e., permittivity) of the image object. Amplitude is the strength of the radar response and phase is the fraction of one complete sine wave cycle (a single SAR wavelength). The phase of the SAR image is determined primarily by the distance between the satellite antenna and the ground targets.

4.3.2.1 Acquisition modes and polarizations

The European Space Agency (ESA) launched one of the constellation of two radar satellite Sentinel-1A in April 2014. It provides C-band images, that is to say with wavelengths of 5.6 cm, in both singular and dual polarization within a cycle of 12 days. Since 2016, the acquisition is made every 6 days with the combination of the two satellites Sentinel-1A and Sentinel-1B. Sentinel-1 images are available with two polarizations: in co-polarization VV and crossed VH. Polarizations is a property of the electromagnetic wave that describes its orientation. Thus in simple VV polarization, the waves are sent and picked up vertically (V). In VH cross polarization, waves are emitted vertically (V) and are received horizontally (H).

Sentinel-1 data are available in the following acquisition modes:

- **Interferometric wide swath (IW):** This mode allows to take measurements on a swath of 250 km with a 5 m $\times$ 20m resolution;
- **Wave mode (WV):** This mode is used to know the direction and height of ocean waves. This mode acquires a series of 20 km $\times$ 20 km;
- **Strip map (SM):** This mode provides coverage with 5 $m \times 5$ m resolution on a strip of 80 km;
- **Extra-width swath (EW):** This mode, similar to the IW mode, is used for maritime or polar areas. It provides resolutions of 20 m to 40 m on a swath of 400 km.

In this work, Interferometric Wide Swath (IW) is the pre-defined mode over land and has a swath width of 250 km. It provides dual polarization images in VV and VH. In addition, Sentinel-1 images are available in two different formats: Ground range detected (GRD) format, comprising perceived intensity and amplitudes and single look complex (SLC) format, which contain phase information, useful for interferometric applications. However at C-band, for a 12 day interval, the interferometric coherence is lost on forested area. To illustrate this point, Figure 4.9 (left) shows this coherence over the Montmorency forest, located near Paris. The black color indicates a low coherence while the white color corresponds to a high coherence value. In this figure, the forest is located at the center of the image (as seen in the right-sided image). As observed, the interferometric coherence is low on the forested area.

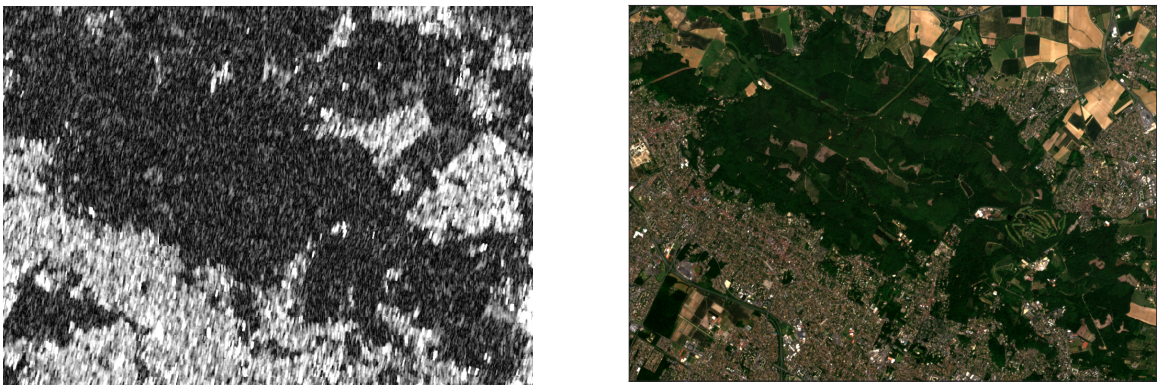


Figure 4.9: Left: Illustration of the interferometric coherence for an interval of 12 days between two Sentinel-1 images of Montmorency forest. Right: optical image of the Montmorency.

For this reason, in the following, the interferometric phase will not be considered, only radiometric attributes will be used to monitor forest health.

4.3.2.2 Derived vegetation indices

Regarding radar images, several approaches are used for monitoring crop growth by exploiting the notion of polarization. Thus vegetation indices are calculated from the polarization ratios. The use of indices is important as they are developed from a combination of radar measurements, which can improve the sensitivity for estimating or monitoring a surface characteristics while reducing other impacts (forest type and structural form, environmental conditions, or radar imaging geometry including incidence angle and topography). Our analyses are based upon the three following indices.

- Radar vegetation index (RVI):

The Radar Vegetation Index (RVI) was proposed in [Kim & van Zyl 2009] and is effective for assessing plant biomass [Kim *et al.* 2012]. It is expressed as follows:

$$\mathrm{RVI} = \frac{8\ \mathrm{VH}}{\mathrm{HH} + \mathrm{VV} + 2\ \mathrm{VH}}. \tag{4.7}$$

It imposes three polarizations HH, VV and VH and can be considered as an alternative to the NDVI index used for optical data [Sahadevan *et al.* 2013] where it is near zero for a smooth bare surface and increases with vegetation growth. It has an enhanced sensitivity to vegetation cover and biomass. In 2005, Trudel *et al.* introduced in [Trudel *et al.* 2012] the radar vegetation index adapted to two polarizations (IVRD), which stands fo dual polarization radar vegetation index. It permits the evaluation of plant biomass using only two polarisations, whether VH and HH or VH and VV. They are computed such as:

$$\mathrm{IVRD}_{HH} = \frac{\mathrm{VH}}{\mathrm{HH} + \mathrm{VH}}, \tag{4.8}$$

$$\mathrm{IVRD}_{VV} = \frac{\mathrm{VH}}{\mathrm{VV} + \mathrm{VH}}. \tag{4.9}$$

Also, Charbonneau *et al.* assumed in [Charbonneau *et al.* 2005] that HH = VV. This assumption is valid when the interaction between the soil and vegetation is negligible. Thus equation for RVI (4.7) reduces to:

$$\mathrm{RVI} = \frac{4\ \mathrm{VH}}{\mathrm{VV} + \mathrm{VH}}. \tag{4.10}$$

This index is useful in case of Sentinel-1 images where the provided data are given in only two polarizations. It will be retained for the following.

- Radar forest degradation index (RFDI):

RFDI [Joshi *et al.* 2015] is an useful index when it comes to monitor changes in forest cover due to deforestation and degradation. Its values range from less than 0.3 for dense forests, between 0.4 and 0.6 for degraded forests, and greater than 0.6 for deforested landscapes.

$$\mathrm{RFDI} = \frac{\mathrm{VV} - \mathrm{VH}}{\mathrm{VV} + \mathrm{VH}}. \tag{4.11}$$

As shown in (4.11), RFDI can be used with dual-polarization imagery such as Sentinel-1.

- Volume scattering index (VSI):

VSI is an indicator of canopy thickness or density. It uses the average cross-polarized magnitude $CS = \frac{\mathrm{VH}+\mathrm{HV}}{2}$ and the average like-polarized magnitude $LK = \frac{\mathrm{VV}+\mathrm{HH}}{2}$ such that:

$$\mathrm{VSI} = \frac{CS}{CS + LK}. \tag{4.12}$$

By using the previous assumption that HH = VV [Charbonneau *et al.* 2005], (4.12) reduces to:

$$\mathrm{VSI} = \frac{\mathrm{VH}}{\mathrm{VV} + \mathrm{VH}} = 4\mathrm{RVI}. \tag{4.13}$$

Since there is a linear relationship between the indices VSI and RVI, only one of them will be used for the following.

4.3.2.3 Pre-processing methods

Pre-processing of SAR imagery was conducted using the ESA SNAP toolbox. SAR image processing chain consists of 3 main steps: (1) Radiometric calibration; (2) Terrain correction; (3) speckle filtering.

- Radiometric calibration

Before processing the SAR images, the data are radiometrically calibrated. Radiometric correction involves removing the misleading influence of topography on backscatter values. For example, the correction eliminates bright backscatter caused by radar reflection from steep slopes, leaving only the backscatter that reveals surface characteristics such as vegetation and soil moisture.

- Terrain correction

SAR images are likely affected by geometric and brightness distortions over elevated and sloping terrain due to the nature of the SAR range mapping and reflectance functions. In fact, it is due

to side-looking rather than straight-down looking imaging and compounded by rugged terrain. Therefore, before using SAR images, these distortions are removed through a process called terrain correction. It consists on moving image pixels into the proper spatial relationship with each other.

- Speckle filtering

In coherent imaging systems, speckle is a strong noise which visually degrades the appearance of images. In fact, it is a physical phenomena generated during the process of creating the SAR image and is caused by coherent radiation. As illustrated in Figure 4.10, within every pixel of the resolution cell, many objects contribute to backscattering. The result is the coherent sum of all contributions (vector addition of all the contributions in the complex plane). Resultant amplitudes interfere, either constructively (yellow vector) or destructively (red vector), depending on the phase of the contributions. As such, the resulting images exhibit bright and dark pixels, even for homogeneous regions. This phenomenon is called speckle noise and it often reduces the quality of images and complicates image interpretation.

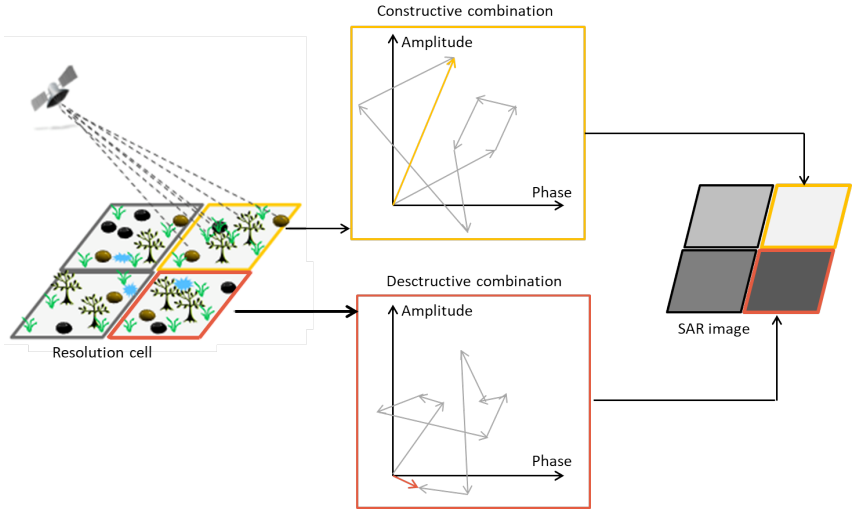


Figure 4.10: Every grey vector corresponds to a scatterer in the resolution cell. Resultant amplitude of the pixel (red and yellow vector) is the coherent sum of all individual contributions[9].

To reduce the speckle noise while preserving the informative structure of the underlying image, several strategies have been proposed, such as the Lee's multiplicative filter and its variants [Lee 1980, Lee *et al.* 2009, Yommy *et al.* 2015].

Regarding the increased frequency of forest insect attacks and the attendant increase of ecological and economic impacts, there is a strong demand for remote sensing-based monitoring approaches for such applications. Many publications oriented their researches on remote sensing data for detecting forest diseases and insect attacks. The following section gives some examples of land cover and land use monitoring application involving Sentinel-1 and Sentinel-2 images.

4.3.3 Land cover and land use monitoring using Sentinel imagery

The increasing availability of global satellite coverage acquisitions, providing high spatial resolution and temporal repetitiveness, such as those of the Sentinel satellites, allows the emergence of methods based on the exploitation of dense time series. Many advanced methods,

[9] https://www.earthstartsbeating.com/2017/05/26/detroit-reducing-the-noise-speckle/

dedicated for both pixel and object-based supervised classification, were applied using Sentinel images. With the advancement in computing capabilities, focus was directed towards machine learning approaches, including random forests (RF), k-nearest neighbors (k-NN), support vector machine (SVM) [Inglada *et al.* 2015, Denize *et al.* 2018, Thanh Noi & Kappas 2018], and deep learning techniques such as convolutional neural network (CNN) and recurrent neural networks [Segal-Rozenhaimer *et al.* 2020, Pelletier *et al.* 2019, Ndikumana *et al.* 2018, Courteille *et al.* 2021].

4.3.3.1 Sentinel for agricultural monitoring

As an example, Sentinel-2 is an important tool for monitoring agricultural activities. Various studies focused on developing methods to support agricultural activities. Many projects aim at producing global agricultural maps using Sentinel-2 products with machine-learning strategies. Other works also used Sentinel-2 images for managing agriculture, crop production monitoring, crop type mapping, irrigation agriculture monitoring, and assessment of crop health [Lambert *et al.* 2018, Hiestermann & Ferreira 2017], etc.

Furthermore, with the increasing potential of SAR images, as they provide spatial information on agricultural crops, Sentinel-1 sensor is investigated for land cover mapping. For example, [Abdikan *et al.* 2016] uses GRD products over the city of Istanbul in Turkey during the year 2016. A composite images using VV, VH and (VV-VH) data are exploited. For the classification, a supervised SVM method is implemented to map land cover types, that is water, urban, forest, agriculture and bareland.

In addition, for the same purpose of mapping land cover areas, deep neural networks (DNNs) are getting increasing attention to deal with land cover classification. For example, [Segal-Rozenhaimer *et al.* 2020] proposed a convolutional neural network (CNN) algorithm for the detection of cloud and cloud shadow fields in multi-channel satellite imagery, with the use of Sentinel-2 and World-View-2 sensors.

4.3.3.2 Sentinel for forest monitoring

In the forestry sector, Sentinel-2 products have been powerful in many applications including mapping of forest area, discrimination of forest types and setting their boundaries [Nzimande *et al.* 2021, Wang *et al.* 2018]. Also, it knows a successful interest for applications related to health monitoring such as leaf area index (LAI) analysis [Sibanda *et al.* 2019] and invasive plant species monitoring [Kattenborn *et al.* 2019, Ng *et al.* 2017]. In [Kattenborn *et al.* 2019], random forest models are trained with multi-temporal Sentinel-1 and Sentinel-2 data to map three invasive species in Chile, where [Ng *et al.* 2017] aims at comparing the use of Sentinel-2 and Pléiades images, combined with random forest classifier, to produce a highly accurate vegetation map that would differentiate an invasive tree species from native forest trees and mixed vegetation classes in Kenya. It has been concluded that despite of the high spatial resolution, Pléiades images are expensive and the free of charge Sentinel-2 data provide a comparable alternative as its increased spectral resolution compensates for the lack of spatial resolution.

In the context of monitoring forest health at the national level, the french ministry of agriculture and food has established an agreement with the joint research unit for territories and environment through remote sensing and spatial information (UMR TETIS) of the national research institute for agriculture, food and the environment (INRAE) of Montpellier in order to develop a processing chain, called ForDead [Dutrieux *et al.* 2021a], allowing to detect and map forest changes. The package is based on the analysis of Sentinel-2 image time series and is already used in the context of the forest health crisis due to bark beetles in the North-East of France. The main objective is to make it available and easily manipulated for a routine use by all services and forest managers such as the ONF and the CNPF for all problems related to forest health.

ForDead uses Sentinel-2 time series collected to define for each pixel a seasonality of the index CR-SWIR, modelled using a harmonic function. The definition of a specific seasonality to each pixel allows to take into account different data variability factors corresponding to the mass of spruce: topography, exposure, tree density, nature of the undergrowth, possible presence of other species. The CR-SWIR calculated for images acquired from the year of interest are compared to the CR-SWIR expected by the periodic model, and values that deviate too much are considered as anomalies. As the anomaly can be explained by multiple factors other than bark beetles attack (imperfect atmospheric correction, presence of undetected clouds, dry period, etc.), the pixel is considered to belong to the disease class when it is detected as anomaly three successive times.

After assessing a general review of existing methods for monitoring forest diseases and insect attacks using remote sensing data. The main contribution of this work is to exploit Sentinel-1 and Sentinel-2 time series to monitor chestnut disease causes by phytophtoras. This work is part of the TEMPOSS project in close collaboration with the national focal point "Data, remote sensing and epidemiology" of the forest health department (DSF).

4.4 Chestnut ink disease

4.4.1 Context

The dieback of chestnut groves in Ile-de-France are due in particular to ink disease. This is particularly the case in the forest of Montmorency (Val d'Oise) which represents the main study area. Chestnut trees, which represent 70% of the Montmorency forest, are affected by the ink disease. This disease is caused by a pathogen, namely the *Phytophthora cambivora* and *Phytophthora cinnamomi*, that destroys the root system of chestnut trees. The two species of Phytophthora produce the same symtpomas: the destruction of all or part of the root system leading to the degradation of its canopy, or even the death of the tree. As illustrated in Figure 4.11, chestnut trees affected by phytophtora are characterised by discoloration of the foliage, where leaves become small and yellow, then branches start wilting until the death of the tree.

[10] https://www.verneuil78.fr/wp-content/uploads/2021/04/article-etat-sanitaire-chataignier-IdF_Oise.pdf

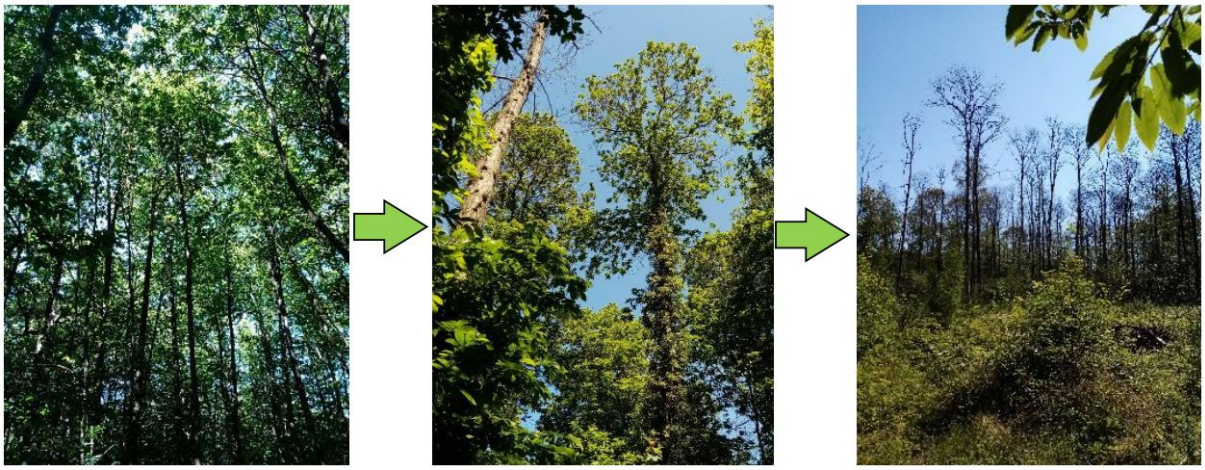


Figure 4.11: Increasing stand levels of chestnut disease due to phytophtoras[10].

In order to better characterise the extent of the phenomenon, which knows a rapid deterioration since 2016, several public (ONF) and private (CNPF) forestry organisms have joined forces with the department of forest health (DSF) in a project to map the health status of chestnut groves in Ile-de-France and in the Oise region based on the analysis of satellite images. It is in this context that the TEMPOSS project was born, aiming at monitoring the forest disease by exploiting Sentinel-1 and Sentinel-2 images.

4.4.2 Ground truth data

The construction of ground truth map is made by foresters who evaluate the infested areas, the different observed stages and the expansion of the infestation. Despite the care taken in data collection, observation are subject to several errors related to the following factors:

- Positioning error: First, there may be an uncertainty related to the geographic positioning of the infestations. Indeed, it is left to the appreciation of forest workers who perform a rough location on a map. It is likely to some error, which may change the quality of the analysis results. In fact, this is the whole point of establishing a detailed observation protocol for data acquisition. The more accurate the protocol is and is correctly followed during field observation companions, the less positioning errors occur.

- Temporal error: A second potential problem is that the expansion of the disease is not constant in time, but can evolve over a long period of time. Foresters delineate the infested area at the time of its discovery. This area will vary depending on the expansion of the disease. It is therefore possible that the area of damaged trees, delineated during surveys of the evaluation period, does not correspond to the original surface marked out by the forest agents.

- Enumeration error: A last possible difficulty remains in the criteria defining the limits of the infested trees. The criteria used for comprehensive censuses are visual. They considered an infested stand as belonging to a damaged class as several dominant trees are infested in a small area. But there is still a proportion of healthy trees, or ruined trees due to other factors. Since then, enumeration errors can easily happen when learning models on inaccurate pixels.

In order to produce harmonized ground truth data, the collection of data in our case is made by following a rigorous observation protocol. This latter has been developed corresponding

to the different stages of disease or pest development (healthy stands and stands affected at different degrees). The main objective is to extract terrain information, analyze and aggregate information from various sources (different organisms interpretations) and produce a final ground truth map.

4.4.2.1 Observation protocol

As several organisations are involved, and in order to produce an effective and homogeneous mapping of the forest health status in a geographic information system (GIS), the DSF has established a simplified assessment method called DEPERIS which can be used by all actors who wish to carry out an assessment of the state of forest health. The method is based on two criteria that describe the appearance of the tree crowns: branch mortality and lack of branching (or lack of needles for softwoods). Those two points are complementary to determine the level of the disease, represented by a score from 0 to 5. A tree is in the best apparent health when its DEPERIS score is at the lowest or when it is ranked with a low letter from A to F in the following abacus (Table 4.2).

		Lack of branching (hardwoods) Lack of needles (softwoods)					
		0	1	2	3	4	5
Branch mortality	0	A	B	C	D	E	F
	1	B	B	C	D	E	F
	2	C	C	C	D	E	F
	3	D	D	D	E	F	F
	4	E	E	E	F	F	F
	5	F	F	F	F	F	F

Table 4.2: Abacus describing the overall condition of the tree canopy can be combined to define a synthetic DEPERIS decline score for each tree[11].

Based on this protocol, the first step for mapping the chestnut groves in the Ile-de-France is to choose a pure chestnut stand (more than 90 % chestnut trees), as homogeneous as possible in terms of cover and health, then its center is defined and positioned. It should be located at a distance of at least 60 m from another type of stand, a path, a track, a road, a clearing or any other open space. Following the DEPERIS protocol, it gives rise to the classes described in the diagram of Figure 4.12.

[11] https://agriculture.gouv.fr/la-methode-deperis-pour-quantifier-letat-de-sante-de-la-foret

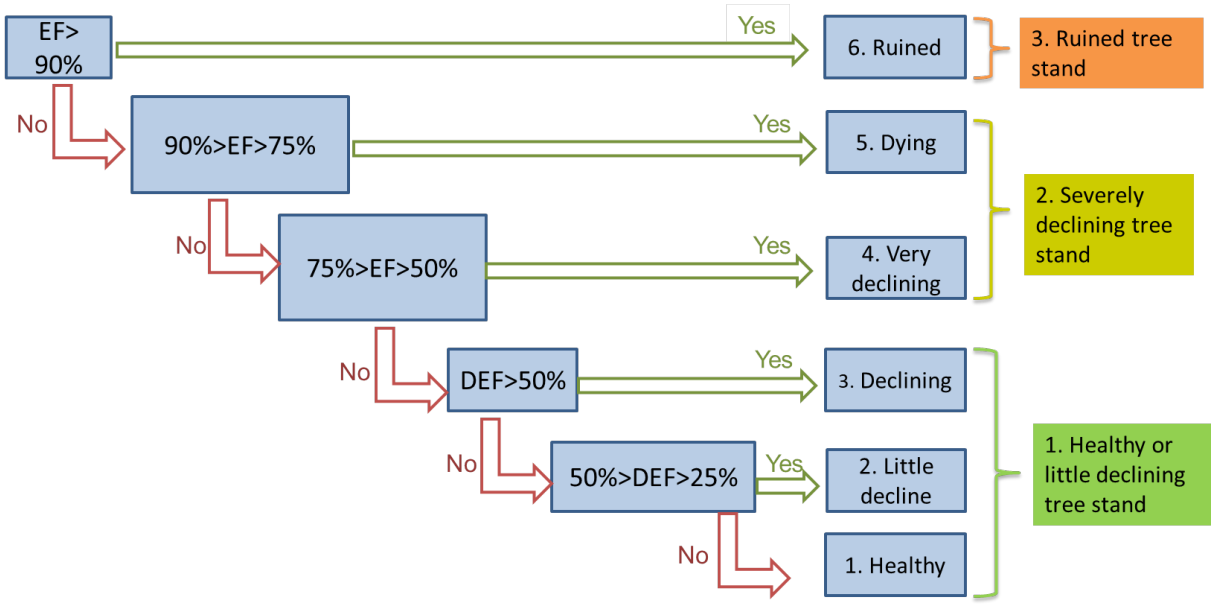


Figure 4.12: Considered (detailed and grouped) classes of the chestnut disease based on the DEPERIS protocol.

For classes of healthy, declining and severely declining stands, a descriptive small square of 20 dominant trees representing the characteristics of the class is geo-referenced using a GNSS receptor. For the other classes, ruined stands and clear-cuts of up to one year, there is no need to set a descriptive plot. In fact, the area of the ruined stand is measured with at least a minimum radius of 20 m from the center. Initially, the dataset is constituted of six classes of intermediate stages from healthy to ruined stand. As illustrated in Figure 4.12, close classes can be grouped to simplify the exercise. In summary, two observation missions of the health status of chestnut groves in the Montmorency state forest, particularly affected by the phenomenon, were carried out in 2019 and 2020.

4.4.2.2 Data management

The data are centralized in the DSF department which make them available to all participating organisations. Using the coordinates of the center of the plot and the radius, the plots were vectorized as discs. They cover a study area and have been integrated into a geographical information system (GIS). They include:

- A vector layer of polygons of the area of interest delineating disease outbreaks.
- Characteristics of the area (density, stand type, disease degree, number and/or volume of trees attacked, etc.)

The chestnut forests map is illustrated in Figure 4.13 where the blue highlighted areas corresponds to chetsnut covers in Ile-de-France region.

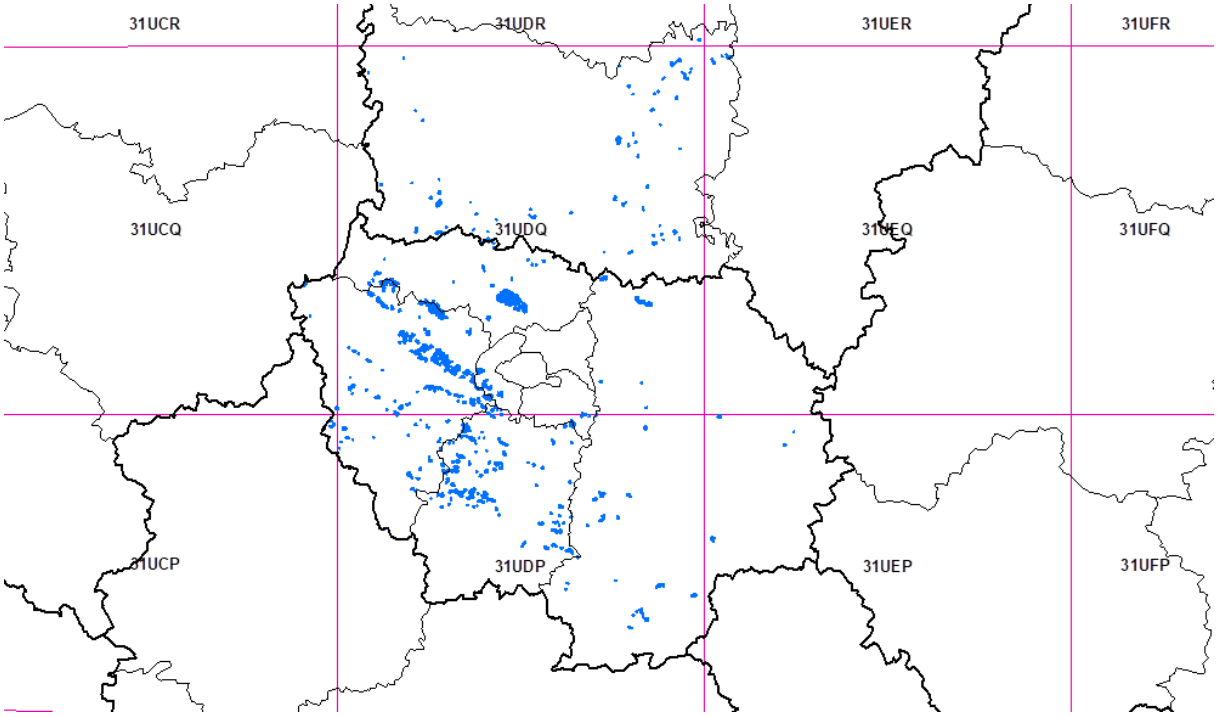


Figure 4.13: Sentinel-2 tiles covering the Ile-de-France region. In blue, chestnut groves in Ile-de-France and Oise (sources: National institute for geographic and forest information (IGN) and the national forests office (ONF))

Then, for our study, the satellite data used concern data acquired from the Sentinel-1 and Sentinel-2 satellites. The studied period is from January 2018 to December 2020. As shown in Figure 4.13, the area corresponding to the stands of interest are spread out over several Sentinel-2 tiles. For our study, the focus is only on tile 31UDQ where most of acquisition campaigns are carried out.

4.4.3 Dataset of experiment

For our study, we have constructed a dataset of experiment based on the provided ground truth map. This dataset is used to build all of classification and regression algorithms proposed on this chapter.

First, as shown in Figure 4.13, the area of study is very large, covering several Sentinel tiles of 100 km^2 each. We have restricted our study at analyzing two forests of the central tile 31UDQ, namely the Montmorency and Marly forests, where the infield observation protocol were carried out. Table 4.3 summarizes the main characteristics of the considered dataset.

	Image size	**Number of polygons**
Montmorency forest	694 × 1009	104
Marly forest	1952 × 1477	61

Table 4.3: Characteristics of images of interest.

To provide a visual example of the study area as well as an overview of the ground truth polygons, Figure 4.14 illustrates one of the considered areas of this study, which covers the Montmorency forest.

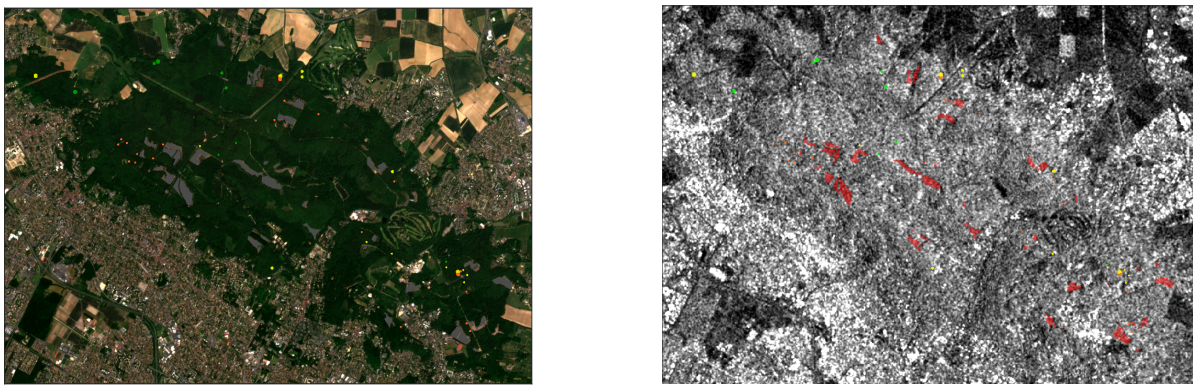


Figure 4.14: Example of Sentinel-1 and Sentinel-2 images of the Montmorency forest. Polygons of colors: green, yellow, orange and gray are the ground truth data for (1) healthy or little decline, (2) declining, (3) ruined and (4) clear-cut classes, respectively. To distinguish colours in the radar image, the clear cut polygons are shown in red.

To analyze the disease behaviour, the study is carried out over a duration of several years from January 2018 to December 2020. In this chapter, reported results will concern only the year 2020 involving the ground truth map produced in 2020. For that, Sentinel-2 image were downloaded as Level-3A products, which were already atmospherically corrected. They are monthly summaries provided every 15^{th} of each month from January to December 2020. For optical data, 12 different dates were used. Regarding the radar imagery, more images are available since the electromagnetic waves penetrate through cloud cover. A total number of 26 Sentinel-1 images are considered. Table 4.4 gives an overview of the used dataset characteristics by summarizing the number of ground truth polygons and the total number of pixels within each class.

	Class 1	Class 2	Class 3	Class 4
Number of polygons	50	27	26	62
Number of pixels	1360	543	492	7117

Table 4.4: Overview of ground truth data for each class.

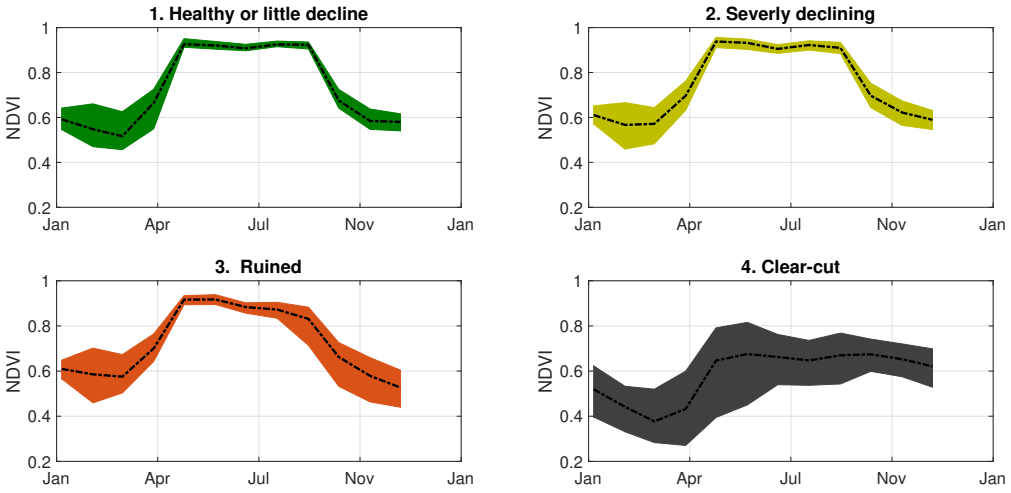
To summarize, the study of this work and the results of all following experiments are reported for the three cases:

- **Optical data** derived from Sentinel-2 images involving 10 spectral bands: 4 bands at resolution of 10 m and 6 bands at the resolution of 20 m. To simplify the processing workflow, the 20 m resolution images are re-sampled to the smallest resolution (i.e, 10 m).
- **Radar data** involving the provided polarizations VV and VH and different combinations of those latter, for instance the ratio VV/VH, the dual difference intensity $VV - VH$, the dual multiple intensity $VV \times VH$ and the vegetation indices RFDI and RVI detailed in section 4.3.2.2.
- **Combination of optical and radar images** to operate simultaneously with information provided by the two sensors. SAR imagery offers roughness information and is not sensible to weather conditions, while multi-spectral images could provide color information of the

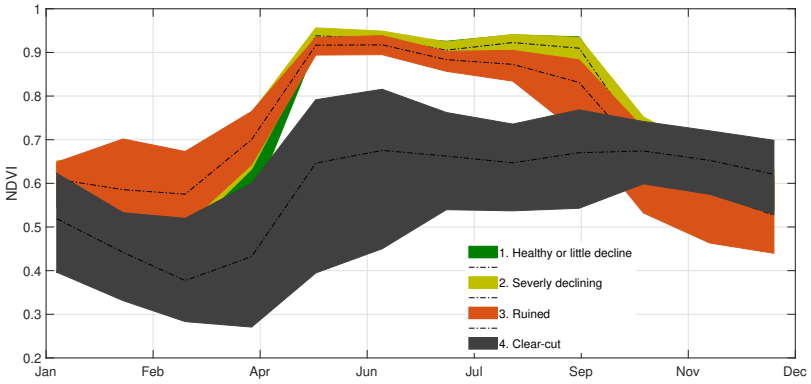
study area. To evaluate their complementarity, we propose a fusion scheme of both radar and optical images.

4.4.3.1 Temporal behaviour

The relationship of different Sentinel-2 indices over the area of interest were investigated. Reflectances change with vegetation phenology over time. What is meant by phenology is the study of cyclic and seasonal natural phenomena. Thus, the information provided by the Sentinel-2 images will not be the same according to the dates of acquisition. Temporal profiles of the indices can therefore give an overview about the development of the disease over time, and the pertinence of the considered indices. Figure 4.15 shows the NDVI index behaviour derived from Sentinel-2 images for the 4 considered classes during the year 2020.



(a)



(b)

Figure 4.15: The median and interquartile range of the NDVI variation according to each class during the year 2020. (a) Temporal profiles for each class, (b) representation of these profiles on the same plot.

The evolution of vegetation phenology is well observed, with an increase of NDVI in spring-summer and a decrease in the autumn for the classes (1), (2) and (3). Also, the clear cut class can be easily separated from the three other classes. It has a lower NDVI index during the vegetation period. However, for the 3 other classes, it is difficult to make a distinction between different health status since the temporal behaviour is almost similar for the three classes (1),

(2) and (3) as shown in Figure 4.15 (b). This point constitutes the main challenge of this dataset.

For further, as the vegetative period is well marked, the focus is within that time-frame. Retained Sentinel images, whether optical or radar, cover the period between May and September 2020. Table 4.5 gives a summary of the used dataset.

	Attributes	**Number of images**
Radar data (Sentinel-1)	VV, VH, VV/VH, VV+VH, VV×VH, RVI and RFDI	May to September 26 images
Optical data (Sentinel-2)	B2, B3, B4, B8, B5, B6, B7, B8a, B11, B12 NDVI, BI, RVI, NDVIre2, NDII and NBR	May to September 6 monthly summaries

Table 4.5: Summary of the used dataset.

4.5 Experiments

Preliminary studies were conducted by our project partner, Thierry Belouard from DSF. Based on optical Sentinel-2 images, he has applied a random forest algorithm to classify different disease levels [Carteron 2019]. From this point, the objective of this study is to present results obtained with the random forest classifier and to compare its performance with the use of second-order descriptors. In this section, a brief overview of the random forest algorithm is assessed. Then, to evaluate the potential of second-order descriptors using Sentinel-1 and Sentinel-2 data, an ensemble based strategy is proposed.

4.5.1 Random forest algorithm

A random forest [Breiman 2001] is an ensemble learning strategy that is used to solve regression and classification problems. It is based on a combination of many classifiers, namely decision trees, to provide solutions to complex problems. Training sets for multiple decision trees in random forest are made using the concept of bootstrapping, which is basically random sampling of the initial training set with replacement.

4.5.1.1 Classification

In classification, the prediction of the random forest is based on the collective predictions of the trees that make up the forest. The resulting random forest classifier with T decision trees is noted as:

$$\mathbf{H}(x) = \{h_t(x)\}_{t=1,\dots,T}, \tag{4.14}$$

where $h_t(x)$ is a random tree which produces a unique decision. The final decision is obtained by electing the most dominant class among predictions by individual trees such as:

$$v_c = \sum_{t=1}^{T} \mathcal{I}(\hat{y}_t == c), \tag{4.15}$$

where $\hat{y}_t$ is the prediction of the $t-$th tree h_t and the indicator function $\mathcal{I}(\hat{y}_t == c)$ takes the value 1 if the condition is met, as such the vote is counted. Given those votes, the final prediction $\hat{y}$ of the random forest is the class with the highest number of votes:

$$\hat{y} = \underset{c \in \{1,\ldots,C\}}{\arg\max}\ v_c \tag{4.16}$$

4.5.1.2 Training the random forest

The training stage of a random forest algorithm is based on the idea of bootstrap aggregating, which works by repeating the following steps for each of the T trees that form the random forest.

1. randomly sample, with replacement, L training examples from the training set;
2. train a tree-based model on the selected sample of first step.

The goal behind constructing the T trees is to divide data into small and homogeneous groups, where each node is constituted of data coming mostly from the same class. Several parameters of the algorithm can be adjusted but two are essential to optimize it, the number of trees that the model will calculate T and the number of predictors tested at each node L whose default value is, in the case of classification, equal to the square root of the number of predictors.

As a splitting criterion, the Gini impurity, introduced by Breiman *et al.* [Breiman 2001], is a commonly used measure of non-homogeneity. It is defined as:

$$G = \sum_c p_c(1 - p_c), \tag{4.17}$$

where p_c is the probability of class c and the interval of Gini index is $[0, 0.5]$. The Gini index is the smallest when the sample set is pure. In addition, the number of considered trees, T, is a hyper-parameter that can be tuned depending on the training set size. For that, cross-validation or out-of-bag error can be employed. The out-of-bag (OOB) error is a method of measuring the prediction error and allows validating the model. In fact, while splitting the samples to feed each node, data points were chosen randomly and with replacement, and the data points which are not a part of that particular sample are known as the out-of-the-bag (OOB) sample. The OOB error is the average error, for each observation in the training set, calculated using predictions from the trees that do not contain it in their respective bootstrap sample. Globally, the larger is the number of trees T, the less is the OOB error. It is because a large number of trees contributes to the proper generalization of the model. There is however no need to run the algorithm with too many trees T. From of a certain value, the OOB error is only slightly improved, where the calculation time increases very substantially.

4.5.2 Application to chestnut ink disease monitoring

Random forest algorithms became popular in applications related to mapping diverse range of vegetation attributes in which they are known to perform well, are fast to compute and easy to tune. Although it requires that the time series dataset is transformed where the temporal arrangement structure is discarded. As such, it implicitly supposes that the observations are independent. Given a sequence of numbers for a time series dataset, we can restructure the data by using time steps as input variables. For a series $\mathbf{X}$ of length T and attributes V, it can

be phrased as an input observation of $T \times V$ attributes.

To evaluate the classification performance, the accuracy of the methods are computed in terms of the overall accuracy (OA) and the Kappa coefficient (K). First of all, in the context of chestnut ink disease, preliminary experiments were carried by performing a simple random forest algorithm to evaluate the potential of considered attributes, which are the spectral bands and the vegetation indices. The obtained results regarding optical data imagery are reported in Tables 4.6 and 4.7. Table 4.6 shows the classification results using 10 Sentinel-2 spectral bands whereas Table 4.7 gathers both spectral bands and vegetation indices derived from Sentinel-2 images. Random forest default settings are retained and a total number of 32 polygons are considered for the training stage.

- Classification results with optical data using spectral bands: the four bands at a resolution of 10 m , and six bands at a resolution of 20 m:

	Class 1	Class 2	Class 3	Class 4
Class accuracy	49.5 ± 12.2	42.1 ± 12.1	61.6 ± 7.6	77.0 ± 7.5
Overall accuracy	59.8 ± 4.7			
Kappa accuracy	45.4 ± 6.0			

Table 4.6: Classification performance involving only spectral bands of Sentinel-2 images.

- Classification results with optical data combining the previous spectral bands and vegetation indices introduced in section 4.3.1.2, namely the NDVI, BI, RVI and indices exploiting red-edge bands: NDVIre2, NDII and NBR.

	Class 1	Class 2	Class 3	Class 4
Class accuracy	48.8 ± 13.8	47.3 ± 11.3	63.8 ± 9.6	84.3 ± 6.2
Overall accuracy	62.8 ± 6.4			
Kappa accuracy	50.0 ± 8.2			

Table 4.7: Classification performance involving spectral bands and vegetation indices derived from Sentinel-2 images.

As observed, a slight gain 3% is obtained when using vegetation indices.

4.5.3 Ensemble covariance pooling for chestnut ink disease classification

Since second-order statistics demonstrated a great success in the previous chapters. It is obvious to exploit their potential in applications related to forest health to enhance classification performance. For that, the proposed approach is based on a global covariance pooling where polygons are considered as individual observations. As the remote sensing data is sensitive to outliers, we also exploit the benefit of combining several classifiers to improve classification robustness. The sub-sampling is performed randomly, without replacement, where a predefined number of polygones are selected to fed each subset. Then an SPD matrix is computed. The general framework of a single subset is illustrated in Figure 4.16.

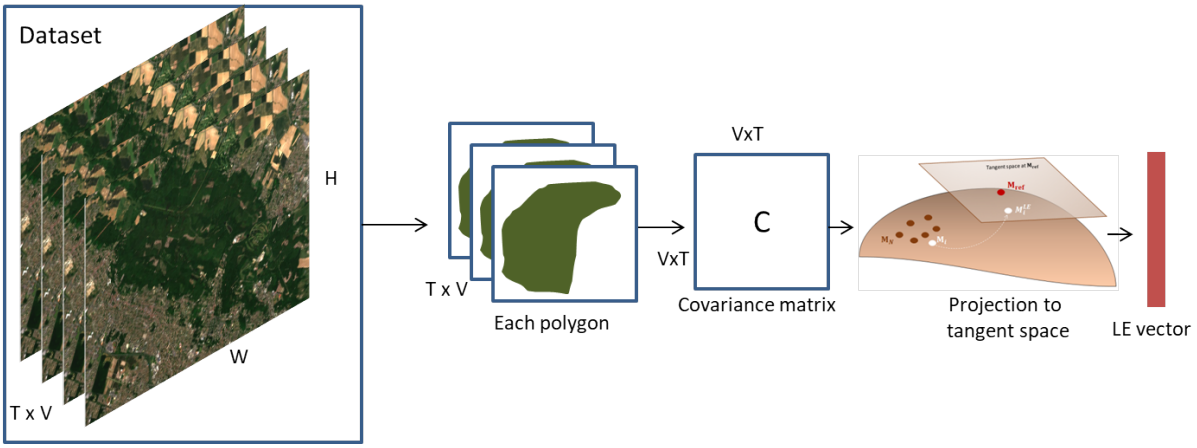


Figure 4.16: Global covariance pooling framework.

To deal with the geometrical properties of the SPD matrix space, the log-Euclidean framework is applied. Hence, second-order statistics of each subset are transformed to their log-Euclidean representation as detailed in equation (1.56) of chapter 1 and are then given as inputs to the random forest classifier. At the end, a majority vote is applied to produce the final prediction as shown in Figure 4.17.

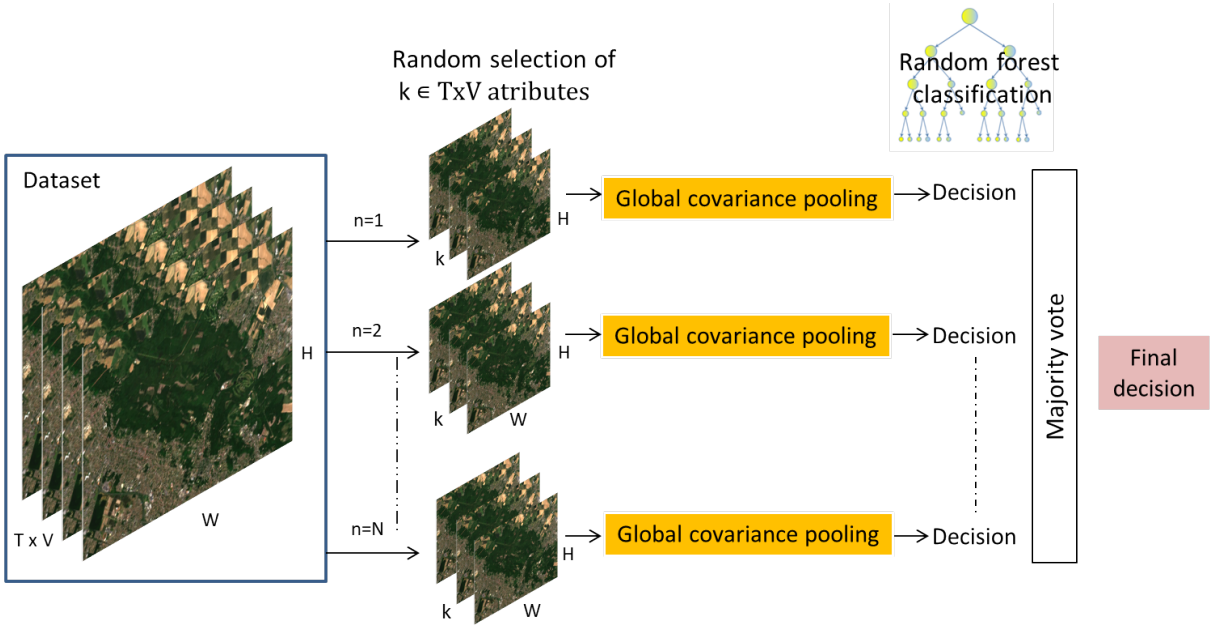


Figure 4.17: Ensemble global covariance pooling framework.

As seen, several ensemble parameters need to be tuned, namely, the number of subsets N and the subset size k which represents the number of considered features from the initial set of $V \times T$ features. To assess a fair comparison, four kind of descriptors are considered:

- the mean of polygons, which represents a first-order statistical feature. Let's consider a set of pixels $\mathbf{x}_{i\{i=1,\ldots,M\}}$ belonging to a single polygon, the mean is given by:

$$\mu = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i,$$

It "roughly" corresponds to the random forest classifier used in the previous section.

- the covariance matrix, where each polygon is represented by a covariance matrix, defined by:

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

- the combination of covariance, after applying the logarithm mapping and the vectorization operations to produce its log-Euclidean representation $\mathbf{C}_{LE}$ (see equation (1.56) of chapter 1), and mean to verify the benefit of fusing first and second-order statistical features through a simple concatenation, such as:

$$(\mathbf{C}_{LE}, \mu) = \text{Concat}\Big(\mathbf{C}_{LE}, \mu\Big),$$

- and the full local Gaussian descriptor (Augmented SPD matrix), which is:

$$\mathbf{C}_{augmented} = |\mathbf{C}|^{-\frac{1}{k+1}} \begin{bmatrix} \mathbf{C} + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix}.$$

In the following, the performance of Sentinel-1 and Sentinel-2 imagery are investigated for monitoring the chestnut disease on the Montmorency forest.

4.5.3.1 Optical data

Figure 4.18 assesses the comparison between different considered descriptors. Moreover, to evaluate the influence of the ensemble parameters, different values of number of subsets N and subset size k are experimented. As observed, the classification results for this method are

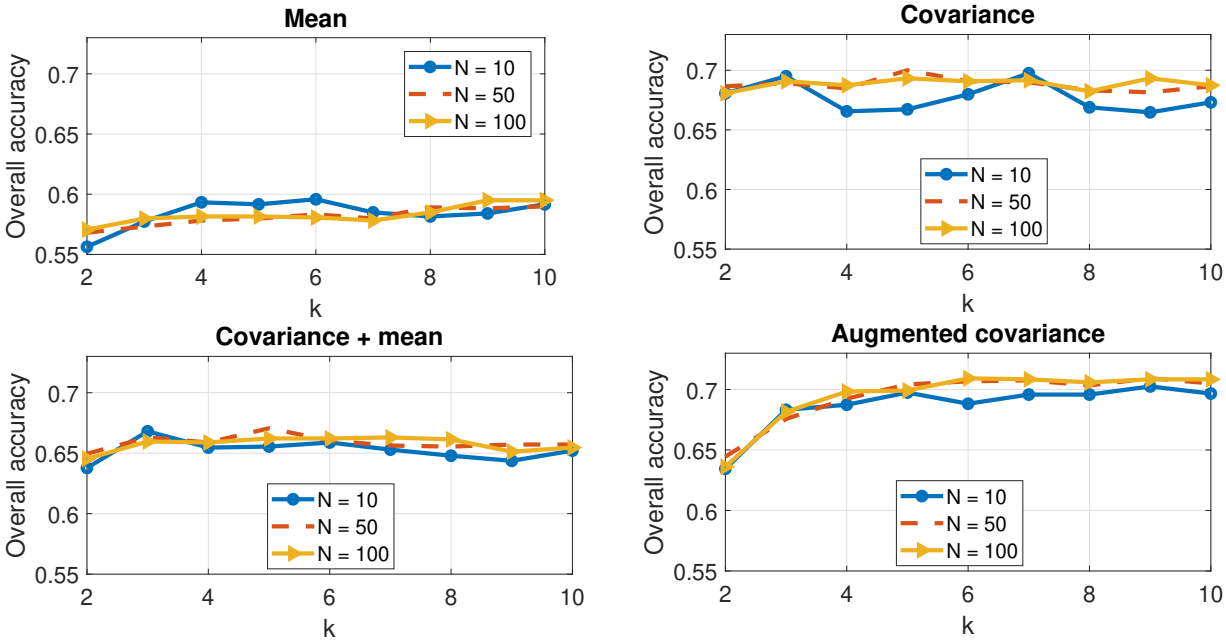


Figure 4.18: Optical imagery classification results: Comparison between first and second-order based models for various ensemble parameter setting.

relatively stable regarding the number of subsets N and the considered features k for each subset. Therefore, the best results are recorded when using the full Gaussian descriptor. It demonstrates the benefit of using second-order features compared to the first-order features represented by the mean. Under this configuration, and using the parameters $N = 50$ subsets, and $k = 9$ features

for each subset, a detailed summary of the obtained results are reported in Table 4.8 comparing the ensemble model whether using first or second-order statistical features.

	Class 1	Class 2	Class 3	Class 4	**OA** (%)	**Kappa** (%)
1^{st}-order	48.5 ± 12.1	**42.1 ± 11.3**	**61.6 ± 7.6**	76.2 ± 7.6	58.8 ± 4.7	44.6 ± 6.0
2^{nd}-order	**73.8 ± 10.2**	27.3 ± 10.1	**68.8 ± 5.9**	**89.5 ± 4.6**	**70.9 ± 2.8**	**59.6 ± 3.5**

Table 4.8: Classification performances with optical data (Sentinel-2) for comparison between first and second-order models: Use of the sample mean for the first order strategy, and the global covariance pooling approach involving the full Gaussian descriptor for the second-order strategy ($N = 50$ and $k = 9$).

Table 4.8 depicts the accuracy for each class. As shown, best results are recorded for the prediction of the clear cuts (class 4) while it totally fails classifying the damaged trees (class 2).

4.5.3.2 Radar data

First of all, in the same manner than with optical data based model, experiments carried out in Figure 4.19 will be used to verify the influence of ensemble parameters and the chosen descriptor on the classification performance using the radar imagery. One more time, the radar

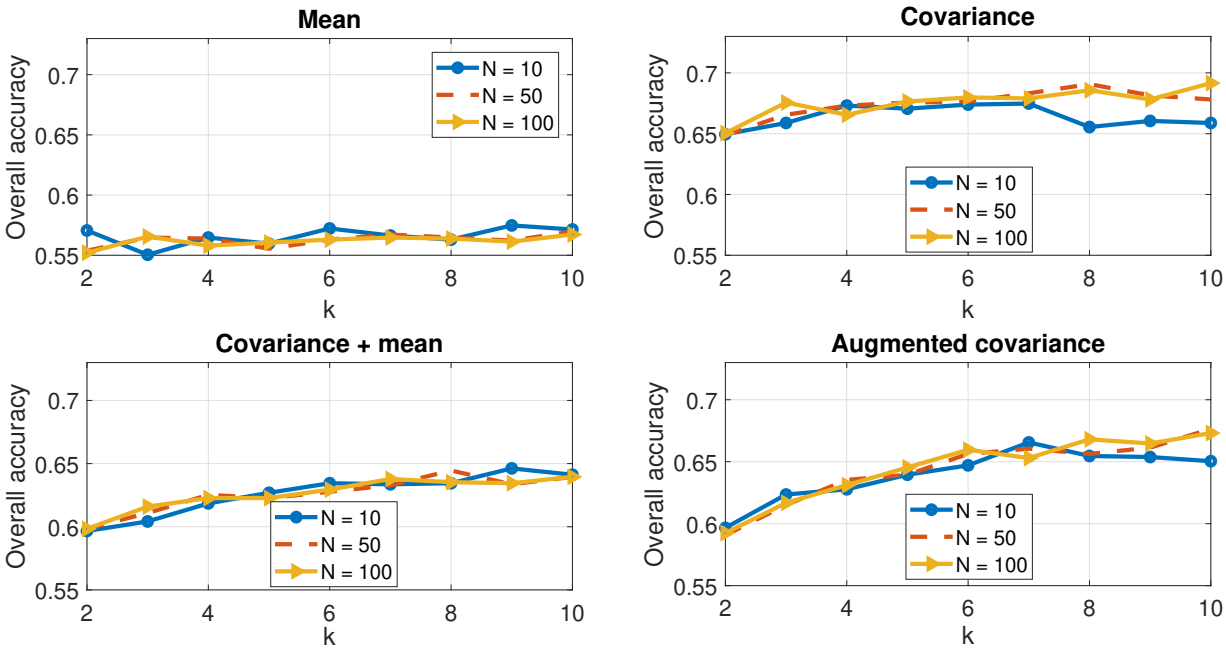


Figure 4.19: Radar imagery classification results: Comparison between first and second-order based models for various ensemble parameter setting.

imagery acts similarly than optical data where the augmented covariance shows the highest classification performance. To summarize, Table 4.9 assesses comparison between first-order (mean) and second-order (augmented covariance) based ensemble models. Same conclusions can be drawn. Accuracy assessment for the classification results highlights the potential of detecting the clear cuts areas.

	Class 1	Class 2	Class 3	Class 4	**OA** (%)	**Kappa** (%)
1^{st}-order	44.5 ± 12.9	**46.3 ± 9.2**	**65.0 ± 4.5**	69.2 ± 7.8	56.2 ± 6.2	41.9 ± 7.6
2^{nd}-order	**74.2 ± 10.9**	11.5 ± 10.4	**65.5 ± 7.3**	**83.7 ± 7.5**	**66.1 ± 3.1**	**52.5 ± 3.6**

Table 4.9: Classification performances with radar data (Sentinel-1) for comparison between first and second-order models: Use of the sample mean for the first order strategy, and the global covariance pooling approach involving the full Gaussian descriptor for the second-order strategy ($N = 50$ and $k = 9$).

4.5.3.3 Fusion of optical and radar imagery

Furthermore, since radar and optical data do not capture the same information, it can be of interest to combine these two kind of information. In the proposed framework, we simply propose to concatenate the optical and radar features as input of our ensemble global covariance pooling model.

	Optical imagery	**Radar imagery**	**Fusion**
OA (%)	**70.9 ± 2.8**	66.1 ± 3.1	**71.3 ± 3.1**
Kappa (%)	**59.6 ± 3.5**	52.5 ± 3.6	**59.7 ± 4.1**

Table 4.10: Ensemble covariance pooling classification results summary comparison between the use of optical, radar data and the fusion of both of them.

Finally, classification results reveal a slight benefit of exploiting second-order statistics by means of combining covariance matrix descriptor and the mean vector. To synthesize, based on tables 4.8 and 4.9, we observed that classification performs well to detect clear-cut areas. Nevertheless, it remains not efficient enough to distinguish the different stages of the disease. Moreover, the different type of remote sensing imagery have approximately comparable behaviours with a small gain of the use of optical and the fusion scheme compared to the radar data.

For the sake of improvement, as the disease evolves continuously from healthy stands to completely destroyed trees, we propose to reformulate the problem as predicting a quantitative variable corresponding to a forest degradation index. For that, a regression model is proposed.

4.5.4 Ensemble covariance pooling for chestnut ink disease regression

Regression trees are constructed by a recursive partitioning of the input space based on some criterion to estimate the regression function. In the regression setting, the prediction of the random forest is the average of the predictions made by the individual trees. If there are T trees in the forest, each providing a prediction $\hat{y}_t$, the final prediction $\hat{y}$ is obtained by:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t \tag{4.18}$$

In contrast with classification methods, the use of regression analysis allows to estimate not only the discrete stages of forest disturbances (e.g., damaged vs. healthy) but also continuous defoliation or tree mortality levels from none to 100%. In the present study, we derived a quantitative forest health indicator using the provided health status scores.

4.5.4.1 Forest health indicator

As explained in section 4.4.2.1, the disease levels are ranked following the DEPERIS protocol, where a tree is in the best health when its score is the lowest, *i.e.* when it is ranked using a score from 0 to 5 or letters from A to F.

A clear cut can appear once a stand is ruined. But it can also happen for an healthy stand which constitutes the final step of the forest cycle. As observed in the previous section, this class can be quite easily detected with whether optical or radar image. For this reason, clear cuts will not be taken into account in the following. Type A trees are the healthiest trees and type F trees are the most affected by ink disease. According to those latter score, we derived the forest health indicator to estimate continuous defoliation according to the percentage of healthy/decaying trees. It is given by:

$$I = \frac{5 \times (\%A) + 4 \times (\%B) + 3 \times (\%C) + 2 \times (\%D) + 1 \times (\%E)}{5}. \tag{4.19}$$

The forest health indicator represents a weighted average of the DEPERIS scores. It ranges from 0 to 1. The lowest score corresponds to a stand composed of 100% of type F trees while the maximum value of 1 is reached for healthy trees (100% of type A). Since this health indicator is a continuous variable, a regression model should be used to predict it from Sentinel-1 and Sentinel-2 observations. For that, the proposed ensemble global covariance pooling framework is re-adapted where the base regressor is a random forest.

The Mean Squared Error (MSE) is used as a default metric for evaluation of the regression performance of the following experiments. In that case, the lower the MSE, the better the performance are.

4.5.4.2 Global covariance pooling

In the same spirit of the experiment carried out for the classification case. Figures 4.20 and 4.21 assess comparison between different choices of first and second-order descriptors, as well as their combination regarding optical and radar data, respectively. Also, it allows to tune the ensemble parameters with N the number of considered subsets and k the features retained for each subset.

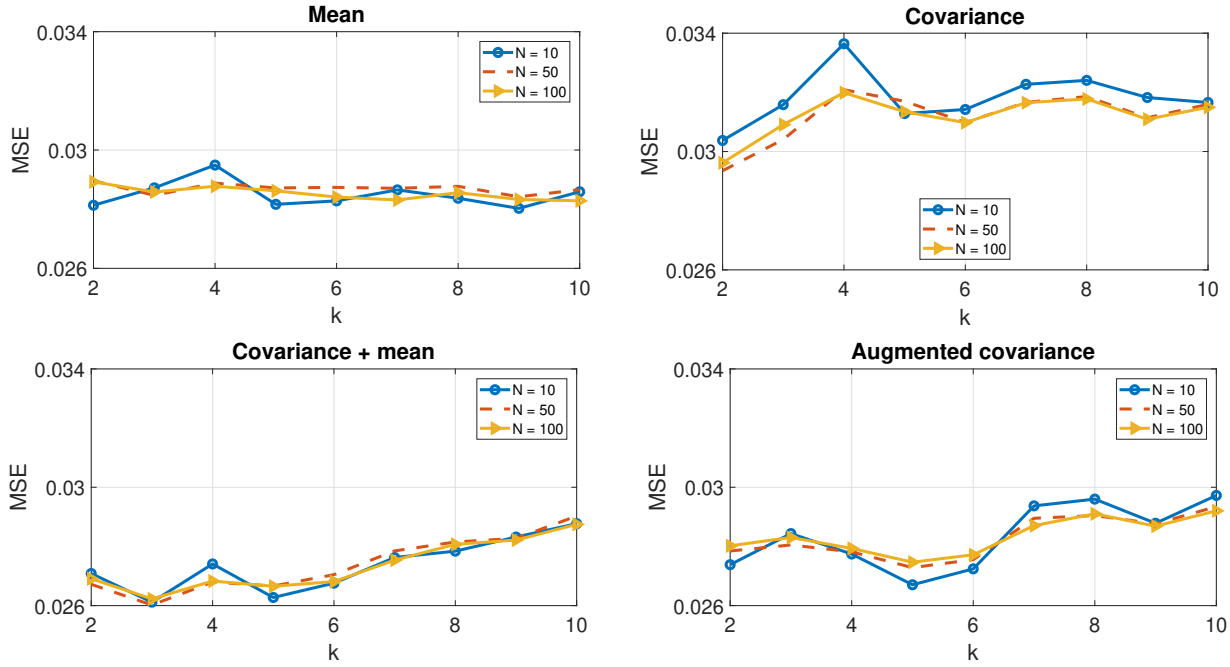


Figure 4.20: MSE comparison between different descriptors using optical data.

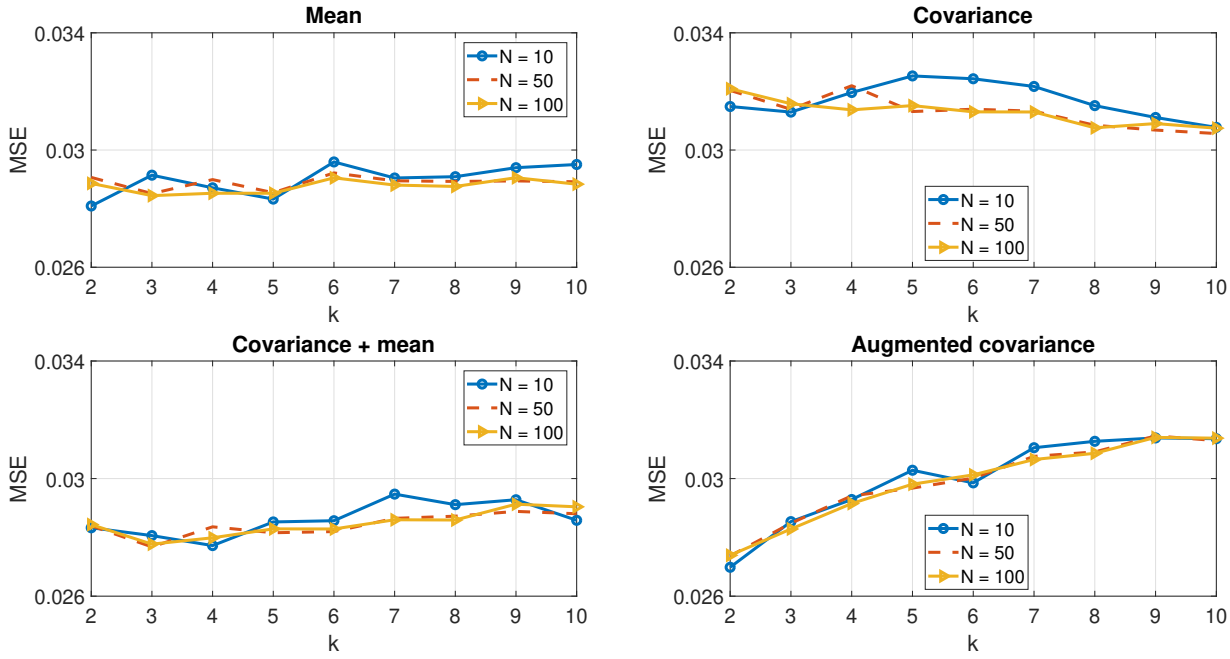
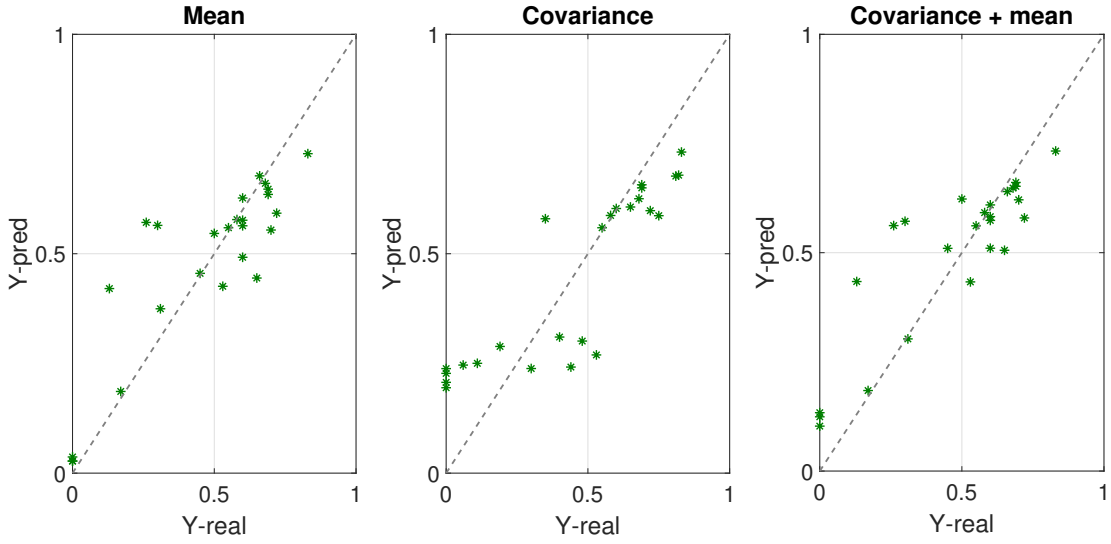
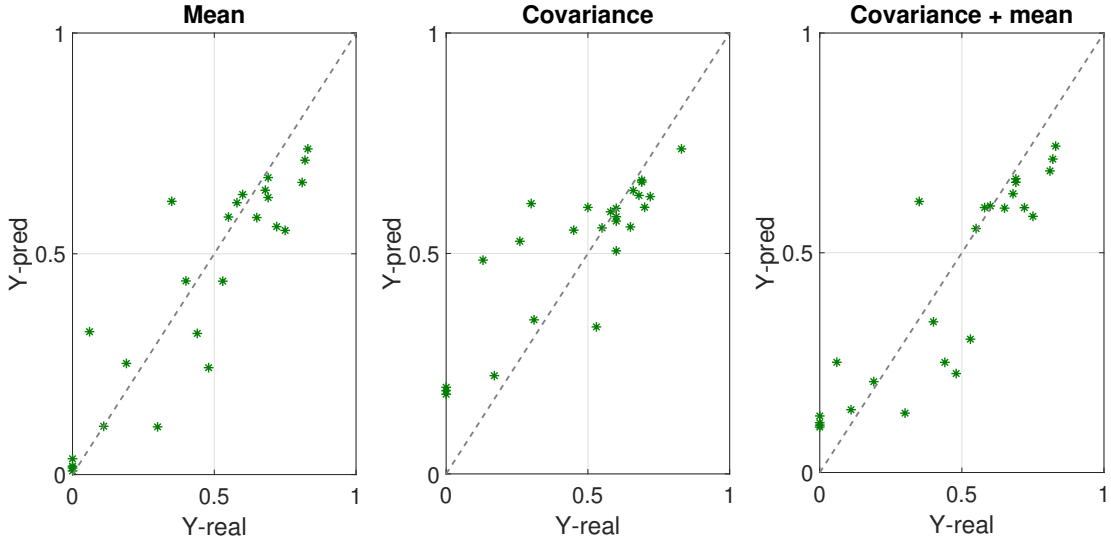


Figure 4.21: MSE comparison between different descriptors using radar data.

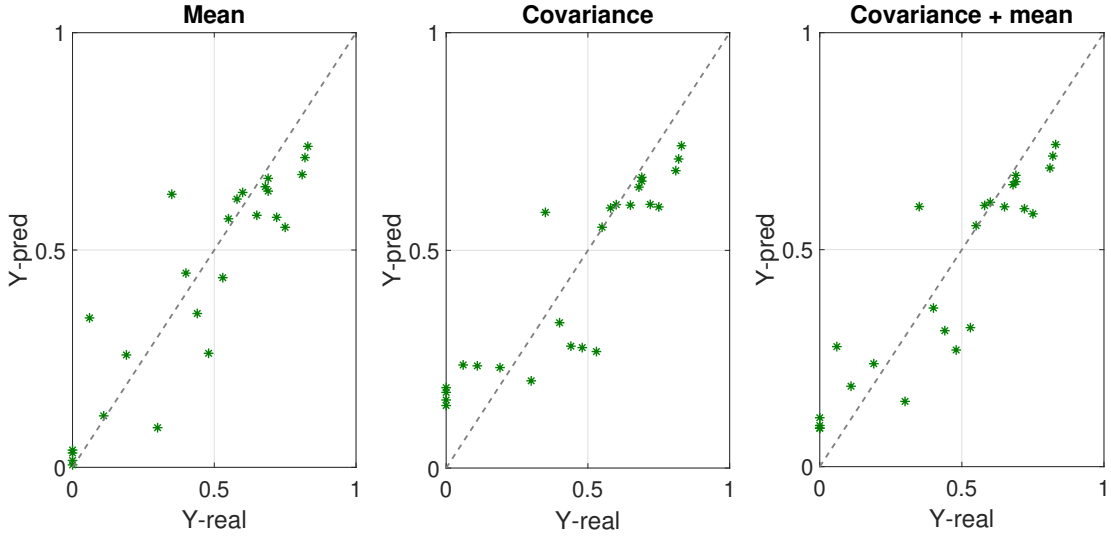
As shown, results are quite stable regarding the number of chosen subsets (N) and the size of each subset (k). However, best results are recorded when using second-order statistical features whether combining first and second-order descriptors (mean + covariance) or using the full Gaussian descriptor (augmented covariance). In the following, first and second-order strategies are compared to illustrate the potential of each. For that, mean feature is compared with the covariance and the combination of both of them using optical and radar imagery as well as the fusion of both of them. Ensemble parameters are set to $N = 50$ subsets with $k = 5$ features for each subset. The results are displayed in Figure 4.22.



(a) Optical data



(b) Radar data



(c) Fusion of optical and radar data

Figure 4.22: Regression model sample illustration. Comparison between the first order based model (mean), second-order (covariance) and combination (mean + covariance) while using optical, radar and fusion of both optical and radar data.

Through the obtained results, the following conclusions can be drawn when comparing the three statistics, *i.e.* the mean vector, the covariance matrix and the combination of the two.

- The use of mean (left of Figure 4.22) operates well when it comes to health indicator values lower than 0.5, this mainly concerns the advanced stages of the disease.
- Meanwhile, the use of covariance matrices (middle of Figure 4.22) allow a better detection of first classes related to healthy or little declining trees. That is for values higher than 0.5.
- The combination benefits from the predictive power of both, first and second features (right of Figure 4.22). As observed, the samples are slightly better predicted.

Quantitatively, the measured mean square error (MSE) for all experiments is represented on Table 4.11.

		Optical imagery	**Radar imagery**	**Fusion**
MSE	Mean	2.87×10^{-2}	2.96×10^{-2}	2.78×10^{-2}
	Covariance	3.17×10^{-2}	3.20×10^{-2}	2.83×10^{-2}
	Covariance + mean	2.67×10^{-2}	2.79×10^{-2}	$\mathbf{2.59\times10^{-2}}$

Table 4.11: Ensemble covariance pooling regression results comparison between the use of optical, radar data and the fusion of both of them.

Results show a minor improvement when combining first and second-order statistical features, for instance the mean vector and the covariance matrix descriptors. Furthermore, the use of optical and radar data in a fusion scheme brings a small enhancement compared to their use separately.

4.6 Conclusion

This chapter is aimed to figure out the potential of using Sentinel-1 and Sentinel-2 imagery for monitoring the chestnut disease of Montmorency forest. First of all, a review of different forest diseases is assessed as well as a brief overview of the remote sensing based methods dedicated to analyze, quantify and monitor forest health issues. Then, the focus is on Sentinel data imagery where the characteristics of two sensors are summarized and some recent studies involving those satellites for land cover and land use applications are investigated.

In this context, the TEMPOSS project objectives are toward the monitoring of forest changes due to sylvosanitary problems, in particular the chestnut ink disease and the bark beetle infestations. Regarding the outbreaks of bark beetles in spruce and pine forests localized in north-eastern France (Grand Est and Bourgogne-Franche-Comté regions), a preliminary study was conducted (but not presented in this chapter). Due to global warming, this insect can reproduce several times on a year which causes severe and rapid forest damages. To monitor effectively the changes that take place within a short period, it is required to use remote sensing data with high temporal resolution. Regarding optical data, Sentinel-2 of level 2-A are well suited. Therefore, provided images have an irregular time lag since orbits with different phases

are involved. Also, due to clouds and shadows, many images are discarded and the irregular sampling is more pronounced. In addition, the temporal signature is not sufficiently marked. Indeed, capabilities of optical satellite data to detect the first stage of a bark beetle infestation, known as the green attack, are limited because no discoloration of the needles occurs at this stage. For all those reasons, the first obtained classification results on the damage caused by bark beetle attacks in the forests of the Meuse, with the use of Sentinel-2 time series, were not satisfying.

In this work, attention is given to the chestnut ink disease that results on significant defoliation and tree mortality in France, especially in the forest of Montmorency (Val d'Oise).

From a methodological point of view, optical and radar sensors are analyzed separately and in combination to detect the different levels of the disease. The remote sensing data were acquired during the year 2020. In order to make the distinction between healthy stands and areas affected by the phytophtoras, two statistical approaches were compared. The first one is based on first-order statistical feature, employing a simple random forest while the second model exploit dependencies between different attributes by means of global second-order statistics. The highest classification accuracy was recorded for clear cut areas in comparison with other classes.

Since the disease continuously changes over time, the classification problem is reformulated as a regression analysis. We have introduced a forest health indicator and proposed to re-adapt the ensemble learning based on covariance pooling framework. When it comes to results, the combination of the first and second-order statistics show their complementarity regarding different disease levels. In addition, the fusion scheme of optical and radar data demonstrates a slight improvement. However, the use of the global covariance pooling approach requires the availability of homogeneous areas, in this case calculus are done directly on polygons. This makes it an object-based method. Therefore, this technique cannot be applied to any forest area without a first step of data collection in order to bring out the homogeneous areas of interest. To overcome that, a pixel-oriented method could be more appropriate where the area of interest would be treated without any prior information. For that, a pre-processing step can be applied to extract homogeneous areas using a segmentation algorithm before performing the proposed covariance pooling approach, with for example a superpixel approach (SLIC, etc.) [Achanta *et al.* 2012].

Future studies should investigate whether the findings from the current study can be validated on larger areas and different forest health applications such as the bark beetles attack. In addition, to improve the results, one can explore monitoring the long-term, including the historical impacts of diseases and insects on forests. Meanwhile, the use of other rich indices such as the CR-SWIR index provided from the developed ForDead package [Dutrieux *et al.* 2021a] could be of great interest for chestnut ink disease application.

Conclusions and perspectives

Conclusions

This main objective of this PhD thesis is to propose new ensemble learning methods on the space of covariance matrices. In this setting, we have conducted supervised classification on the basis of the log-Euclidean metric, where covariance matrices of CNN features or multispectral attributes were represented by their corresponding log-Euclidean vectors. We have evaluated the potential of these second-order features, with comparison with first-order based models, for various applications including remote sensing scene classification and time series classification. Additionally, a special interest has been given to assess the potential of radar (Sentinel-1) and optical (Sentinel-2) images to monitor forestry health problem, in particular regarding the chestnut ink disease in the Montmorency forest.

In the context of both applications, whether for time series classification or image classification, we were interested in a problem of classification on the space of covariance matrices. Chapter 1 introduced the space of symmetric positive definite matrices and the basic notion of information geometry that are necessary to handle this kind of data. For that, two complete Riemannian statistical frameworks, based on the log-Euclidean (LE) and affine-invariant (AI) Riemannian metrics, were presented. Gaussian models are considered on both metric spaces, as well as their Gaussian mixture model extensions. In practice, regarding the comparison of their corresponding GMM models, close conclusions were drawn where comparable classification results were obtained. Therefore, the use of AI metric leads to very complex calculations whereas considering the Log-Euclidean metric, the GMM modeling is limited to a single tangent plane defined at the identity matrix. However, projecting covariance matrices on a tangent plane can lead to projection distortions. To overcome this problem, we have proposed to consider a Gaussian mixture model (GMM) with multiple reference points, one for each component of the model as well as deriving an expectation-maximization algorithm to estimate the GMM parameters. It allows limiting the distortion during the projection as well as maintaining the computational complexity at its lowest.

Based on that general mathematical background for the log-Euclidean representation of a covariance matrix, and motivated by the success of deep neural networks and hybrid architectures, we have proposed in chapter 2 two hybrid transfer learning approaches based on covariance pooling of CNN features. The first approach, called the hybrid log-Euclidean Fisher vectors (Hybrid LE FV) and published in [Akodad *et al.* 2018b], relies on the log-Euclidean Fisher vector encoding of region covariance matrices of first and second CNN layers. The second architecture, uses high-level features issued from deeper layers that are pooled together by computing their covariance matrix, namely the ensemble learning covariance pooling (ELCP) [Akodad *et al.* 2019c]. In order to give more importance to small objects of interest in the scene, the visual saliency map are used, where largest weights are given to the most salient regions during covariance computation. Furthermore, to take full advantage of the local and global aspects, we have proposed to fuse both strategies on an ensemble learning architecture based on the most diverse ensembles. The proposed Ensemble LE FV - ELCP efficiently combines the provided decisions and allows to enhance the classification performance.

The resulting approaches have been applied for different remote sensing scene classification problems, including aerial and satellites based images, and they demonstrate a significant accuracy improvement compared to state-of-the-art methods. This is firstly thanks to the use of second-order statistics through the covariance pooling and secondly by means of ensemble learning techniques that helped enhancing the performances.

Time series classification is a general task that can be useful across many subject-matter domains and applications. The overall goal of chapter 3 focuses on extending some classification algorithms to second-order statistical features. For that, the study was carried out step by step. Starting from the point-to-point distance measurement between two time series, we have highlighted two limitations of the usual Euclidean distance, namely, its sensitivity to temporal transformations, such as speed changes between the two sequences and its variance toward re-parameterization. To overcome those drawbacks, we have investigated the Dynamic Time Warping (DTW) to limit the distortion in the time axis and the re-parameterization invariance is solved by the use of square-root velocity function (SRVF) representations of the considered time series. Furthermore, the second-order matrix trajectory (SPD-MTS) is introduced to take into account the dependencies between time series attributes. Since then, multivariate classification problem is rephrased as a second-order matrix trajectory classification problem. Given that, the SRVF framework is extended to the transport square-root velocity function (TSRVF) representation [Su *et al.* 2014a] as a representation that provides a way to deal with trajectories on Riemannian manifolds while preserving invariance properties. Moreover, in order to get benefit of the advantages of kernel methods, codebook based representations and ensemble learning strategies, we have investigated the potential of extending the time series cluster kernel (TCK) [Mikalsen *et al.* 2018] method to second-order matrix trajectories, namely the second-order TCK (SO-TCK) algorithm [Akodad *et al.* 2020a]. All these approaches were validated on various multivariate time series including applications on action recognition and crop classification.

Finally, in the context of forest health issues, chapter 4 focused on monitoring the chestnut ink disease in Montmorency forest. We evaluated the potential of both radar and optical images acquired respectively by Sentinel 1 and 2 sensors. We also investigate the interest of a multimodal approach by combining these two kinds of data. The main challenge in this application is the need to find patterns in the data that are different between classes in order to determine the class of the time series at hand. To this end, we first reviewed some state-of-the-art vegetation and degradation forest indices that can be extracted from satellite images to monitor forest health. Then, based on the covariance pooling of these indices, we have introduced an ensemble learning approach for the classification of the forest health status (healthy, declining, severely declining, and clear cut). Next, as the disease evolves continuously from healthy stands to completely destroyed trees, we proposed to reformulate the problem as predicting a quantitative variable corresponding to the forest degradation (or health status) index. We also have demonstrated how the proposed classification model can be adapted to this regression problem. On that basis, we have evaluated the potential of Sentinel 1 and 2 data for this application.

The main contributions and the obtained results in the course of this thesis work have been valued by several scientific publications: a journal article and four conference papers. In

addition, it is worth noting that the different projects involved have opened the opportunity to propose several internship subjects and thus to supervise five master students.

Perspectives

The work of this thesis have opened the way to many prospects and areas of improvement. At the end of each chapter, we have drawn some future works. In this section, we propose to develop some of these ideas.

Since the considered remote sensing datasets are of small dimension, we have chosen the use of pre-trained CNN models on the ImageNet dataset, to transfer the learning to our datasets of interest. However, ImageNet dataset is constituted of computer vision images, which have different characteristics than remote sensing images, usually made of multiple spectral bands. Pre-trained models on the ImageNet dataset are, therefore, not the most relevant for remote sensing classification problems. Recently, with the increase of freely available satellite images, covering large areas and providing different temporal and spectral characteristics, new large remote sensing databases have emerged. For example, BigEarthNet[12] is a benchmark archive, consisting of 590,326 pairs of Sentinel-1 and Sentinel-2 image patches. Pre-trained models on this dataset are hence more suitable to remote sensing application.

To go one step further, the proposed second-order based architectures could be extended to an end-to-end learning strategy which would permit developing forward and backward propagation regarding second-order pooling layers. In that context, several methods were proposed in the literature to integrate global covariance pooling into deep CNNs. For example, the global Matrix Power Normalized COVariance (MPN-COV) pooling architecture proposed in [Li *et al.* 2018] is based on capturing CNN feature maps correlation through a robust covariance matrix estimator and thus produces a normalized covariance matrix as a representation. Similar idea may be considered here to extend the proposed ensemble covariance pooling (ELCP) method to an end-to-end training. A successful first attempt was completed during this thesis by a master student, Maria-Camelia Puscasu, where she has achieved, along the same lines as the MPN-COV technique, an end-to-end training of multi-layer ensemble based method involving covariance pooling layers for remote sensing scene classification and texture classification.

Moreover, training such a complex model may include extremely complex problem statements with expensive computationally costs. One solution that attracted an increasing attention in recent years rely on the use of knowledge distillation [Hinton *et al.* 2015]. It is the process of transferring knowledge from a large model to a smaller one while preserving comparable performance. For example, in our case, the ensemble learning strategy knowledge may be transferred to a simple CNN model which would be less expensive.

Besides, saliency based algorithms attracted intense attention in recent years. It permits increasing the representation power of many models by focusing on important features and reducing the impact of unnecessary ones. In this work, we have demonstrated the potential of weighting covariance matrices by saliency. Therefore, since saliency maps are generated

[12] http://bigearth.net/

using a pre-trained model, the produced attention maps remain the same for each subset while feature maps vary from one subset to another. To overcome that issue, different architectures were proposed in the literature to design an attention module. For example, the Convolutional Block Attention Module (CBAM) [Woo *et al.* 2018] took inspiration from the CNN model to extract informative features by blending cross-channel and spatial information together. By incorporating this trainable module on the ELCP branches, the learning will be made using the corresponding subset feature maps in the channel and spatial axes respectively.

Furthermore, many works focuses on neural network strategies to deal with time series classification problems, one candidate is the convolutional neural network (CNN) which is the most popular, since it is able to successfully capture the spatial and temporal patterns through the use of trainable filters, assigning importance to specific patterns using trainable weights. Back to SO-TCK strategy, consideration might be given to integrate CNN networks on the proposed architecture. Following the idea of chapter 2, an hybrid architecture may be developed combining pre-trained CNN models and the proposed SO-TCK method. By doing so, convolutional layers perform a convolution of an input series with a set of filter matrices to obtain as output different series of feature maps, with the goal to extract richer statistics of high-level features. Those produced convolutional time series would be used further to compute second-order matrix trajectories.

In the context of forest health monitoring, as a perspective, the proposed strategy can be complemented by overcoming the need of homogeneous areas, for covariance pooling, with a segmentation algorithms which would be seen as a pre-processing step. For example a super-pixel approach (SLIC, etc.) can be used [Achanta *et al.* 2012]. In addition, the study may be extended to a longer time period to include the multi-year aspect as well as considering other weather information such as temperature and rainfall to evaluate their effect. Also, the proposed strategies could be tried on other applications related to forest health diseases.

Bibliography

[Abdikan *et al.* 2016] Saygin Abdikan, Fusun Balik Sanli, Mustafa Üstüner and F. Calò. *Land cover mapping using Sentinel-1 SAR Data.* volume XLI-B7, pages 757–761, 06 2016.

[Absil *et al.* 2008] P.-A. Absil, R. Mahony and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, Princeton, NJ, 2008.

[Achanta *et al.* 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua and Sabine Süsstrunk. *SLIC superpixels compared to state-of-the-art superpixel methods.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pages 2274–2282, 2012.

[Acharya *et al.* 2018] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel and Luc Van Gool. *Covariance pooling For facial expression recognition.* CoRR, vol. abs/1805.04855, 2018.

[Adelabu *et al.* 2014] Sam Adelabu, Onisimo Mutanga and Elhadi Adam. *Evaluating the impact of red-edge band from Rapideye image for classifying insect defoliation levels.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 95, page 3441, 09 2014.

[Akodad *et al.* 2018a] Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Encodage de matrices de covariance par les vecteurs de Fisher log-Euclidien : application à la classification supervisée d'images satellitaires.* In GDR-ISIS/CNES CCT-TSI, Paris, France, 2018.

[Akodad *et al.* 2018b] Sara Akodad, Lionel Bombrun, Charles Yaacoub, Yannick Berthoumieu and Christian Germain. *Image classification based on log-Euclidean Fisher Vectors for covariance matrix descriptors.* In International Conference on Image Processing Theory, Tools and Applications (IPTA), Xi-an, China, Nov 2018.

[Akodad *et al.* 2019a] Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Architectures hybrides de réseaux de neurones exploitants les statistiques d'ordre 2 sur les sorties des couches convolutives d'un CNN : application à la classification supervisée d'images satellitaires.* In GDR-MADICS, Rennes, France, 2019.

[Akodad *et al.* 2019b] Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Encodage de matrices de covariance par les vecteurs de Fisher log-Euclidien : application á la classification supervisée d'images satellitaires.* In GRETSI, Lille, France, 2019.

[Akodad *et al.* 2019c] Sara Akodad, Solène Vilfroy, Lionel Bombrun, Charles C Cavalcante, Christian Germain and Yannick Berthoumieu. *An ensemble learning approach for the classification of remote sensing scenes based on covariance pooling of CNN features.* In 27th European Signal Processing Conference, 27th European Signal Processing Conference, La Coruña, Spain, September 2019.

[Akodad *et al.* 2020a] Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Cluster kernel for learning similarities between symmetric positive definite matrix time series.* In 2020 IEEE International Conference on Image Processing (ICIP), pages 3304–3308, Abu Dhabi, United Arab Emirates, October 2020. IEEE.

[Akodad *et al.* 2020b] Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Méthode d'ensemble pour la classification de trajectoires temporelles de matrices symétriques définies positives.* In GDR-ISIS, France, 2020.

[Akodad *et al.* 2020c] Sara Akodad, Lionel Bombrun, Junshi Xia, Yannick Berthoumieu and Christian Germain. *Ensemble learning approaches based on covariance pooling of CNN features for high resolution remote sensing scene classification.* Remote Sensing, vol. 12, no. 20, 2020.

[Anirudh *et al.* 2017] Rushil Anirudh, Pavan Turaga, Jingyong Su and Anuj Srivastava. *Elastic functional coding of Riemannian trajectories.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 5, pages 922–936, 2017.

[Arandjelović & Zisserman 2013] R. Arandjelović and A. Zisserman. *All about VLAD.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 1578–1585, 2013.

[Arandjelovic *et al.* 2015] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla and J. Sivic. *NetVLAD: CNN architecture for weakly supervised place recognition.* CoRR, vol. abs/1511.07247, 2015.

[Armijo 1966] L. Armijo. *Minimization of functions having Lipschitz continuous first partial derivatives.* Pacific Journal of Mathematics, vol. 16, no. 1, pages 1–3, 1966.

[Arsigny *et al.* 2006] V. Arsigny, P. Fillard, X. Pennec and N. Ayache. *Log-Euclidean metrics for fast and simple calculus on diffusion tensors.* In Magnetic Resonance in Medicine, volume 56, pages 411–421, Aug 2006.

[Atto *et al.* 2016] A.M. Atto, Emmanuel Trouvé, Jean-Marie Nicolas and Thu Trang Lê. *Wavelet operators and multiplicative observation models -application to SAR image time series analysis.* IEEE Transactions on Geoscience and Remote Sensing, vol. 54, July 2016.

[Backes *et al.* 2012] Andre Backes, Dalcimar Casanova and Odemir Bruno. *Color texture analysis based on fractal descriptors.* Pattern Recognition, vol. 45, pages 1984–1992, May 2012.

[Bagnall *et al.* 2015] Anthony Bagnall, Jason Lines, Jon Hills and Aaron Bostrom. *Time-series classification with COTE: the collective of transformation-based ensembles.* IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 9, pages 2522–2535, 2015.

[Bagnall *et al.* 2016a] A. Bagnall, J. Lines, J. Hills and A. Bostrom. *Time-series classification with COTE: The collective of transformation-based ensembles.* In 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pages 1548–1549, May 2016.

[Bagnall *et al.* 2016b] Anthony Bagnall, Aaron Bostrom, James Large and Jason Lines. *The great time series classification bake off: an experimental evaluation of recently proposed algorithms. extended version*, 2016.

[Bagnall *et al.* 2017] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large and Eamonn Keogh. *The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances.* Data Mining and Knowledge Discovery, vol. 31, no. 3, pages 606–660, May 2017.

[Bahdanau *et al.* 2016] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. *Neural machine translation by jointly learning to align and translate*, 2016.

[Bailly *et al.* 2016] Adeline Bailly, Simon Malinowski, Romain Tavenard, Laetitia Chapel and Thomas Guyet. *Dense bag-of-temporal-SIFT-words for time series classification.* In Ahlame Douzal-Chouakria, José A. Vilar and Pierre-François Marteau, editors, Advanced Analysis and Learning on Temporal Data, pages 17–30, Cham, 2016. Springer International Publishing.

[Barachant *et al.* 2013] A. Barachant, S. Bonnet, M. Congedo and C. Jutten. *Classification of covariance matrices using a Riemannian-based kernel for BCI applications.* NeuroComputing, vol. 112, pages 172–178, 2013.

[Bay *et al.* 2006] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. *SURF: speeded up robust features.* In Aleš Leonardis, Horst Bischof and Axel Pinz, editors, Computer Vision – ECCV 2006, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Baydogan *et al.* 2013] Mustafa Baydogan, George Runger and Eugene Tuv. *A bag-of-features framework to classify time series.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pages 2796–802, Nov 2013.

[Berndt & Clifford 1994] Donald J. Berndt and James Clifford. *Using dynamic time warping to find patterns in time series.* In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS94, pages 359–370. AAAI Press, 1994.

[Bertini *et al.* 2012] F. Bertini, O. Brand, S. Carlier, Umberto Del Bello, Matthias Drusch, R. Duca, V. Fernandez, C. Ferrario, M.H. Ferreira, C. Isola, V. Kirschner, P. Laberinti, M. Lambert, G. Mandorlo, P. Marcos, Philippe Martimort, S. Moon, P. Oldeman, M. Palomba and J. Pineiro. *Sentinel-2 ESAs optical high-resolution mission for GMES operational services.* ESA bulletin. Bulletin ASE. European Space Agency, vol. SP-1322, Mar 2012.

[Box & Jenkins 1994] George Edward Pelham Box and Gwilym M. Jenkins. Time series analysis: forecasting and control. Prentice Hall PTR, USA, 1994.

[Boyd *et al.* 2013] I. L. Boyd, P. H. Freer-Smith, C. A. Gilligan and H. C. J. Godfray. *The consequence of tree pests and diseases for ecosystem services.* Science, vol. 342, no. 6160, page 1235773, 2013.

[Breiman 2001] L. Breiman. *Random forests.* Machine Learning, vol. 45, no. 1, pages 5–32, 2001.

[Breiman 2005] L. Breiman. *Bagging predictors.* Machine Learning, vol. 24, pages 123–140, 2005.

[Brodatz 1966] Phil. Brodatz. Textures; a photographic album for artists and designers. Dover Publications New York, 1966.

[Cai *et al.* 2007] Deng Cai, Xiaofei He and Jiawei Han. *Efficient Kernel Discriminant Analysis via Spectral Regression.* In Seventh IEEE International Conference on Data Mining (ICDM 2007), pages 427–432, 2007.

[Cai *et al.* 2017] Sijia Cai, Wangmeng Zuo and Lei Zhang. *Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization.* In 2017 IEEE International Conference on Computer Vision (ICCV), pages 511–520, 2017.

[Calinon & Jaquier 2019] S. Calinon and N. Jaquier. *Gaussians on Riemannian manifolds for robot learning and adaptive control.* ArXiv e-prints, Sep 2019.

[Calonder *et al.* 2012] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha and Pascal Fua. *BRIEF: computing a local binary descriptor very fast.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pages 1281–1298, 2012.

[Carteron 2019] Clotilde Carteron. *Création de données de référence pour l'application de la télédétection au suivi de la santé des forêts*, 2019. Master Research Thesis.

[Castaño-Moraga *et al.* 2007] C.A. Castaño-Moraga, C. Lenglet, R. Deriche and J. Ruiz-Alzola. *A Riemannian approach to anisotropic filtering of tensor fields.* Signal Processing, vol. 87, no. 2, pages 263–276, 2007. Tensor Signal Processing.

[Charbonneau *et al.* 2005] F Charbonneau, M. Trudel and R Fernandes. *Use of Dual-Polarization and Multi-Incidence SAR for soil permeability mapping.* Advanced Synthetic Aperture Radar (ASAR), 2005.

[Chatfield *et al.* 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi and Andrew Zisserman. *Return of the devil in the details: delving deep into convolutional nets.* CoRR, vol. abs/1405.3531, 2014.

[Cheng *et al.* 2020] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo and Gui-Song Xia. *Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pages 3735–3756, 2020.

[Cheret *et al.* 2018] Véronique Cheret, Yousra Hamraoui, Michel Goulard, Jean-Philippe Denux, Hervé Poilvé and Michel Chartier. *Mapping health status of chestnut forest stands using Sentinel-2 images.* ForestSAT 2018, October 2018. Poster.

[Cho *et al.* 2014a] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau and Yoshua Bengio. *On the properties of neural machine translation: encoder-decoder approaches*, 2014.

[Cho *et al.* 2014b] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio. *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* In Alessandro Moschitti, Bo Pang and Walter Daelemans, editors, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar., pages 1724–1734. ACL, 2014.

[Cimpoi *et al.* 2016] M. Cimpoi, S. Maji, I. Kokkinos and A. Vedaldi. *Deep filter banks for texture recognition, description, and segmentation.* International Journal of Computer Vision, vol. 118, no. 1, pages 65–94, May 2016.

[Courteille *et al.* 2021] Hermann Courteille, A. Benoît, N Méger, A Atto and D. Ienco. *Channel-based attention for LCC using Sentinel-2 time series*, 2021.

[Cruz *et al.* 2018] Rafael M.O. Cruz, Robert Sabourin and George D.C. Cavalcanti. *Dynamic classifier selection: recent advances and perspectives.* Information Fusion, vol. 41, pages 195 – 216, 2018.

[Csurka *et al.* 2004] G. Csurka, C. R. Dance, L. Fan, J. Willamowski and C. Bray. *Visual categorization with bags of keypoints.* In Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision, pages 1–22, 2004.

[Cuturi 2011] Marco Cuturi. *Fast global alignment kernels.* In Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML11, page 929936, Madison, WI, USA, 2011. Omnipress.

[Dau *et al.* 2019] H. Dau, A. Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, C. Ratanamahatana and Eamonn J. Keogh. *The UCR time series archive.* IEEE/CAA Journal of Automatica Sinica, vol. 6, pages 1293–1305, 2019.

[de Beurs & Townsend 2008] Kirsten de Beurs and Philip Townsend. *Estimating the effect of gypsy moth defoliation using MODIS.* Remote Sensing of Environment, vol. 112, Oct 2008.

[Dempster *et al.* 1977] A. P. Dempster, N. M. Laird and D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, no. 1, pages 1–38, 1977.

[Denize *et al.* 2018] Julien Denize, Laurence Hubert-Moy, Samuel Corgne, Julie Betbeder and E. Pottier. *Identification of winter land use in temperate agricultural landscapes based on Sentinel-1 and 2 times-series.* In 38th IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, July 2018. IEEE.

[Diba *et al.* 2017] A. Diba, A. M. Pazandeh and L. Van Gool. *Deep visual words: improved Fisher vector for image classification.* In 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pages 186–189, May 2017.

[Donoho 2000] David Donoho. *High-dimensional data analysis: the curses and blessings of dimensionality.* AMS Math Challenges Lecture, pages 1–32, Jan 2000.

[Douze *et al.* 2011] M. Douze, A. Ramisa and C. Schmid. *Combining attributes and Fisher vectors for efficient image retrieval.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 745–752, 2011.

[Dua & Graff 2017] Dheeru Dua and Casey Graff. *UCI machine learning repository*, 2017.

[Dunis & Huang 2002] Christian L. Dunis and Xuehuan Huang. *Forecasting and trading currency volatility: an application of recurrent neural regression and model combination.* Journal of Forecasting, vol. 21, no. 5, pages 317–354, 2002.

[D'Urso & Maharaj 2012] Pierpaolo D'Urso and Elizabeth Ann Maharaj. *Wavelets-based clustering of multivariate time series.* Fuzzy Sets and Systems, vol. 193, pages 33–61, 2012. Theme : Data Analysis.

[Dutrieux *et al.* 2021a] Raphael Dutrieux, Jean-Baptiste Feret, Kenji Ose and Florian De Boissieu. *Package Fordead*, 2021.

[Dutrieux *et al.* 2021b] Raphaël Dutrieux, Jean-Baptiste Féret and Ose Kenji. *Mise au point dune méthode reproductible pour le suivi généralisé des dégâts de scolytes par télédétection satellitaire.* Rendez-vous techniques de l'ONF - 69-70, 2021.

[Faloutsos *et al.* 2000] Christos Faloutsos, M. Ranganathan and Yannis Manolopoulos. *Fast subsequence matching in time-series databases.* ACM SIGMOD Record, vol. 23, June 2000.

[Faraki *et al.* 2014] M. Faraki, M. T. Harandi, A. Wiliem and B. C. Lovell. *Fisher tensors for classifying human epithelial cells.* Pattern Recognition, vol. 47, no. 7, pages 2348 – 2359, 2014.

[Faraki *et al.* 2015a] M. Faraki, M. T. Harandi and F. Porikli. *More about VLAD: A leap from Euclidean to Riemannian manifolds.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 4951–4960, 2015.

[Faraki *et al.* 2015b] M. Faraki, M. Palhang and C. Sanderson. *Log-Euclidean bag of words for human action recognition.* IET Computer Vision, vol. 9, no. 3, pages 331–339, 2015.

[Fassnacht *et al.* 2014] Fabian Fassnacht, Hooman Latifi, Aniruddha Ghosh, Paawan Joshi and Barbara Koch. *Assessing the potential of hyperspectral imagery to map bark beetle-induced tree mortality.* Remote Sensing of Environment, vol. 140, pages 533–548, Jan 2014.

[Fréchet 1948] M. Fréchet. *Les éléments aléatoires de nature quelconque dans un espace distancié.* 1948.

[Freund & Schapire 1996] Yoav Freund and Robert E. Schapire. *Experiments with a new boosting algorithm*, 1996.

[Fukushima 1988] Kunihiko Fukushima. *Neocognitron: A hierarchical neural network capable of visual pattern recognition.* Neural Networks, vol. 1, no. 2, pages 119–130, 1988.

[Gamboa 2017] John Cristian Borges Gamboa. *Deep learning for time-series analysis*, 2017.

[Gao & Vasconcelos 2004] Dashan Gao and Nuno Vasconcelos. *Discriminant saliency for visual recognition from cluttered scenes.* volume 17, Jan 2004.

[Gao *et al.* 2019] Z. Gao, J. Xie, Q. Wang and P. Li. *Global second-order pooling convolutional networks.* In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3019–3028, 2019.

[Garnot *et al.* 2020] V. Sainte Fare Garnot, L. Landrieu, S. Giordano and N. Chehata. *Satellite image time series classification with pixel-set encoders and temporal self-attention.* In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12322–12331, Los Alamitos, CA, USA, June 2020. IEEE Computer Society.

[Gini & Greco 2002] F. Gini and M. V. Greco. *Covariance matrix estimation for CFAR detection in correlated heavy tailed clutter.* Signal Processing, vol. 82, no. 12, pages 1847–1859, 2002.

[Gogolou *et al.* 2019] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas and Anastasia Bezerianos. *Comparing similarity perception in time series visualizations.* IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 1, pages 523–533, 2019.

[Goudet 2016] Morgane Goudet. *Réseau systématique de suivi des dommages forestiers*. 2016.

[Hagolle *et al.* 2017] Olivier Hagolle, Mireille Huc, Camille Desjardins, Stefan Auer and Rudolf Richter. *MAJA algorithm theoretical basis document*, December 2017.

[Halko *et al.* 2010] Nathan Halko, Per-Gunnar Martinsson and Joel A. Tropp. *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, 2010.

[He *et al.* 2018] N. He, L. Fang, S. Li, A. Plaza and J. Plaza. *Remote sensing scene classification using multilayer stacked covariance pooling*. IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 12, pages 6899–6910, Dec 2018.

[He *et al.* 2020] N. He, L. Fang, S. Li, J. Plaza and A. Plaza. *Skip-connected covariance network for remote sensing scene classification*. IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 5, pages 1461–1474, 2020.

[Hiestermann & Ferreira 2017] J. Hiestermann and S. L. Ferreira. *Cloud-based agricultural solution: a case study of near real-time regional agricultural crop growth information in South Africa*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-3/W2, pages 79–82, 2017.

[Hinton *et al.* 2015] Geoffrey Hinton, Oriol Vinyals and Jeff Dean. *Distilling the knowledge in a neural network*, 2015.

[Ho 1998] Tin Kam Ho. *The random subspace method for constructing decision forests*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pages 832–844, 1998.

[Hochreiter & Schmidhuber 1997] Sepp Hochreiter and Jürgen Schmidhuber. *Long short-term memory*. Neural Computation, vol. 9, no. 8, pages 1735–1780, Nov 1997.

[Hu *et al.* 2015] W. Hu, Y. Huang, L. Wei, F. Zhang and H. Li. *Deep convolutional neural networks for hyperspectral image classification*. Journal of Sensors, 2015.

[Huang & Gool 2017] Z. Huang and L. Van Gool. *A Riemannian network for SPD matrix learning*. In AAAI Conference on Artificial Intelligence, pages 2036–2042, 2017.

[Ienco *et al.* 2017] Dino Ienco, Raffaele Gaetano, Claire Dupaquier and Pierre Maurel. *Land cover classification via multitemporal spatial data by deep recurrent neural networks*. IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 10, pages 1685–1689, 2017.

[Ienco 2017] Dino Ienco. *TiSeLaC: time series land cover classification challenge*, 2017.

[Ilea *et al.* 2016] I. Ilea, L. Bombrun, C. Germain, R. Terebes, M. Borda and Y. Berthoumieu. *Texture image classification with Riemannian Fisher vectors*. In IEEE International Conference on Image Processing, pages 3543 – 3547, 2016.

[Ilea *et al.* 2018a] I. Ilea, L. Bombrun, S. Said and Y. Berthoumieu. *Covariance matrices encoding based on the log-Euclidean and affine invariant Riemannian metrics*. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 506–515, July 2018.

[Ilea *et al.* 2018b] I. Ilea, L. Bombrun, S. Said and Y. Berthoumieu. *Fisher vector coding for covariance matrix descriptors based on the log-Euclidean and affine invariant Riemannian metrics.* Journal of Imaging, vol. 4, no. 7, 2018.

[Inglada *et al.* 2015] J. Inglada, M. Arias, B. Tardy, D. Morin, S. Valero, O. Hagolle, G. Dedieu, G. Sepulcre, S. Bontemps and P. Defourny. *Benchmarking of algorithms for crop type land-cover maps using Sentinel-2 image time series.* In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pages 3993–3996, 2015.

[Ionescu *et al.* 2015] C. Ionescu, O. Vantzos and C. Sminchisescu. *Matrix backpropagation for deep networks with structured layers.* In IEEE International Conference on Computer Vision (ICCV), pages 2965–2973, 2015.

[Ismail Fawaz *et al.* 2019] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller. *Deep learning for time series classification: a review.* Data Min. Knowl. Discov., vol. 33, no. 4, page 917963, July 2019.

[Jaakkola & Haussler 1998] T. Jaakkola and D. Haussler. *Exploiting generative models in discriminative classifiers.* In In Advances in Neural Information Processing Systems 11, pages 487–493. MIT Press, 1998.

[James 1973] A. T. James. The variance information manifold and the functions on it, pages 157 – 169. Academic Press, 1973.

[Jayasumana *et al.* 2013] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong li and Mehrtash Harandi. *Kernel methods on the Riemannian manifold of symmetric positive definite matrices.* June 2013.

[Jégou *et al.* 2010] H. Jégou, M. Douze, C. Schmid and P. Pérez. *Aggregating local descriptors into a compact image representation.* In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3304–3311, June 2010.

[Jeong *et al.* 2011] Young-Seon Jeong, Myong K. Jeong and Olufemi A. Omitaomu. *Weighted dynamic time warping for time series classification.* Pattern Recognition, vol. 44, no. 9, pages 2231 – 2240, 2011. Computer Analysis of Images and Patterns.

[Ji *et al.* 2011] Lei Ji, Li Zhang, Bruce Wylie and Jennifer Rover. *On the terminology of the spectral vegetation index (NIRSWIR)/(NIR+SWIR).* International Journal of Remote Sensing, vol. 32, pages 6901–6909, Nov 2011.

[Joachims 1998] T. Joachims. *Text categorization with suport vector machines: learning with many relevant features.* In Proceedings of the 10th European Conference on Machine Learning, pages 137–142. Springer-Verlag, 1998.

[Jordan 1969] Carl F. Jordan. *Derivation of leaf-area index from quality of light on the forest floor.* Ecology, vol. 50, no. 4, pages 663–666, 1969.

[Joshi *et al.* 2007] Shantanu Joshi, Eric Klassen, Anuj Srivastava and Ian Jermyn. *Removing shape-preserving transformations in square-root elastic (SRE) framework for shape analysis of curves.* volume 4679, pages 387–398, Aug 2007.

[Joshi *et al.* 2015] Neha Joshi, Edward Mitchard, Natalia Woo, Jorge Torres, Julian Moll-Rocek, Andrea Ehammer, Murray Collins, Martin Jepsen and Rasmus Fensholt. *Mapping dynamics of deforestation and forest degradation in tropical forests using radar satellite data.* Environmental Research Letters, vol. 10, page 034014, Jan 2015.

[Karcher 1977] H. Karcher. *Riemannian center of mass and mollifier smoothing.* Communications on Pure and Applied Mathematics, vol. 30, no. 5, pages 509–541, 1977.

[Kattenborn *et al.* 2019] Teja Kattenborn, Javier Lopatin, Michael Förster, Andreas Christian Braun and Fabian Ewald Fassnacht. *UAV data as alternative to field sampling to map woody invasive species based on combined Sentinel-1 and Sentinel-2 data.* Remote Sensing of Environment, vol. 227, pages 61–73, 2019.

[Keeley 2009] Jon Keeley. *Fire intensity, fire severity and burn severity: A brief review and suggested usage.* International Journal of Wildland Fire, vol. 18, pages 116–126, 01 2009.

[Kelly & Meentemeyer 2002] Maggi Kelly and R.K. Meentemeyer. *Landscape dynamics of the spread of Sudden Oak Death.* Photogrammetric Engineering & Remote Sensing, vol. S68, pages 1001–1009, Oct 2002.

[Kennedy *et al.* 2014] Robert E Kennedy, Serge Andréfouët, Warren B Cohen, Cristina Gómez, Patrick Griffiths, Martin Hais, Sean P Healey, Eileen H Helmer, Patrick Hostert, Mitchell B Lyons, Garrett W Meigs, Dirk Pflugmacher, Stuart R Phinn, Scott L Powell, Peter Scarth, Susmita Sen, Todd A Schroeder, Annemarie Schneider, Ruth Sonnenschein, James E Vogelmann, Michael A Wulder and Zhe Zhu. *Bringing an ecological view of change to Landsat-based remote sensing.* Frontiers in Ecology and the Environment, vol. 12, no. 6, pages 339–346, 2014.

[Keogh & Pazzani 2001] Eamonn J. Keogh and Michael J. Pazzani. *Derivative dynamic time warping.* In In SIAM International Conference on Data Mining, 2001.

[Kim & van Zyl 2009] Yunjin Kim and Jakob J. van Zyl. *A time-series approach to estimate soil moisture using polarimetric radar data.* IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 8, pages 2519–2527, 2009.

[Kim *et al.* 2012] YiHyun Kim, Thomas J. Jackson, Rajat Bindlish, Hoonyol Lee and Sukyoung Hong. *Radar Vegetation Index for Estimating the Vegetation Water Content of Rice and Soybean.* IEEE Geoscience and Remote Sensing Letters, vol. 9, pages 564–568, 2012.

[Kong & Fowlkes 2016] Shu Kong and Charless C. Fowlkes. *Low-rank bilinear pooling for fine-grained classification.* CoRR, vol. abs/1611.05109, 2016.

[Kriegeskorte 2015] Nikolaus Kriegeskorte. *Deep neural networks: a new framework for modelling biological vision and brain information processing.* bioRxiv, 2015.

[Kriegler *et al.* 1969] F. J. Kriegler, W. A. Malila, R. F. Nalepka and W. Richardson. *Preprocessing transformations and their effects on multispectral recognition.* In Remote Sensing of Environment, VI, page 97, January 1969.

[Krizhevsky *et al.* 2012] A. Krizhevsky, I. Sutskever and G. E. Hinton. *ImageNet classification with deep convolutional neural networks.* In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[Kruskal & Liberman 1999] Joseph B. Kruskal and Mark Liberman. *The symmetric time-warping problem: from continuous to discrete.* In David Sankoff and Joseph B. Kruskal, editors, Time warps, string edits, and macromolecules - the theory and practice of sequence comparison, chapter 4. CSLI Publications, Stanford, CA 94305, 1999.

[Kuncheva & Whitaker 2003] Ludmila I. Kuncheva and Christopher J. Whitaker. *Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy.* Machine Learning, vol. 51, no. 2, pages 181–207, 2003.

[Kutner 2005] M.H. Kutner. Applied linear statistical models. McGraw-Hill/Irwin series operations and decision sciences. McGraw-Hill Irwin, 2005.

[Labsir 2020] Samy Labsir. *Méthodes statistiques fondées sur les groupes de Lie pour le suivi d'un amas de débris spatiaux.* Thesis, Université de Bordeaux, December 2020.

[Lambert *et al.* 2018] Marie-Julie Lambert, Pierre C. Sibiry Traoré, Xavier Blaes, Philippe Baret and Pierre Defourny. *Estimating smallholder crops production at village level from Sentinel-2 time series in Mali's cotton belt.* Remote Sensing of Environment, vol. 216, pages 647–657, 2018.

[Le Cun *et al.* 1990] Y. Le Cun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard and L. D. Jackel. *Handwritten digit recognition with a back-propagation network.* In D. S. Touretzky, editor, Advances in Neural Information Processing Systems 2, pages 396–404. Morgan-Kaufmann, 1990.

[Le Cun *et al.* 1998] Y. Le Cun, L. Bottou, Y. Bengio and P. Haffner. *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, vol. 86, no. 11, pages 2278–2324, 1998.

[Lee *et al.* 2009] Jong-Sen Lee, Jen-Hung Wen, T.L. Ainsworth, Kun-Shan Chen and A.J. Chen. *Improved sigma filter for speckle filtering of SAR imagery.* IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 1, pages 202–213, 2009.

[Lee 1980] Jong-Sen Lee. *Digital image enhancement and noise filtering by use of local statistics.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 2, pages 165–168, 1980.

[Lee 1997] J.M. Lee. Riemannian manifolds: An introduction to curvature. Graduate Texts in Mathematics. Springer New York, 1997.

[Lenglet *et al.* 2006] Christophe Lenglet, Mikaël Rousson, R. Deriche and Olivier Faugeras. *Statistics on the manifold of multivariate normal distributions: theory and application to diffusion tensor MRI processing.* Journal of Mathematical Imaging and Vision, vol. 25, pages 423–444, Oct 2006.

[Li *et al.* 2017] E. Li, J. Xia, P. Du, C. Lin and A. Samat. *Integrating multilayer features of convolutional neural networks for remote sensing scene classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 10, pages 5653–5665, Oct 2017.

[Li *et al.* 2018] Peihua Li, Jiangtao Xie, Qilong Wang and Wangmeng Zuo. *Is second-order information helpful for large-scale visual recognition?*, 2018.

[Lines *et al.* 2016] J. Lines, S. Taylor and A. Bagnall. *HIVE-COTE: the hierarchical vote collective of transformation-based ensembles for time series classification.* In 2016 IEEE 16th International Conference on Data Mining (ICDM), pages 1041–1046, Dec 2016.

[Lines *et al.* 2018] Jason Lines, Sarah Taylor and Anthony Bagnall. *Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles.* ACM Transactions on Knowledge Discovery from Data, vol. 12, pages 1–35, July 2018.

[Lovric *et al.* 2000] M. Lovric, M. Min-Oo and E. A Ruh. *Multivariate normal distributions parametrized as a Riemannian symmetric space.* Journal of Multivariate Analysis, vol. 74, no. 1, pages 36 – 48, 2000.

[Lowe 2004] D. G. Lowe. *Distinctive image features from scale-invariant keypoints.* 2004.

[Luther *et al.* 1997] J. E. Luther, S. Franklin, J. Hudák and J. P. Meades. *Forecasting the susceptibility and vulnerability of balsam fir stands to insect defoliation with Landsat Thematic Mapper data.* Remote Sensing of Environment, vol. 59, pages 77–91, 1997.

[López-Quiroz *et al.* 2009] Penélope López-Quiroz, Marie-Pierre Doin, Florence Tupin, Pierre Briole and Jean-Marie Nicolas. *Time series analysis of Mexico City subsidence constrained by radar interferometry.* Journal of Applied Geophysics, vol. 69, no. 1, pages 1–15, 2009. Advances in SAR Interferometry from the 2007 Fringe Workshop.

[MacQueen 1967] J. MacQueen. *Some methods for classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[Malhotra *et al.* 2017] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal and Gautam Shroff. *TimeNet: Pre-trained deep recurrent neural network for time series classification*, 2017.

[Maronna *et al.* 2006] R. Maronna, R. Martin and V. Yohai. *Robust statistics: theory and methods.* 2006.

[Martinez & Kak 2001] A.M. Martinez and A.C. Kak. *PCA versus LDA.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pages 228–233, 2001.

[Meddens *et al.* 2011] Arjan J.H. Meddens, Jeffrey A. Hicke and Lee A. Vierling. *Evaluating the potential of multispectral imagery to map multiple stages of tree mortality.* Remote Sensing of Environment, vol. 115, no. 7, pages 1632–1642, 2011.

[Mika *et al.* 1999] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K. R. Mullers. *Fisher discriminant analysis with kernels.* In Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468), pages 41–48, Aug 1999.

[Mikalsen *et al.* 2018] K. Ø. Mikalsen, Filippo Maria Bianchi, Cristina Soguero-Ruiz and Robert Jenssen. *Time series cluster kernel for learning similarities between multivariate time series with missing data.* Pattern Recognition, vol. 76, pages 569 – 581, 2018.

[Moosmann *et al.* 2006] Franck Moosmann, Diane Larlus and Frédéric Jurie. *Learning saliency maps for object categorization.* In International Workshop on The Representation and Use of Prior Knowledge in Vision (in ECCV '06), Graz, Austria, May 2006.

[Muirhead 1982] R. J. Muirhead. Aspects of multivariate statistical theory. Wiley Series in Probability and Statistics. Wiley, 1982.

[Nageleisen *et al.* 2015] Louis-Michel Nageleisen, Thierry Bélouard and Joseph Meyer. *Le hanneton forestier (Melolontha hippocastani Fabricius 1801) en phase épidémique dans le nord de lAlsace.* Revue Forestière Française, Jan 2015.

[Ndikumana *et al.* 2018] Emile Ndikumana, Dinh Ho Tong Minh, Nicolas Baghdadi, Dominique Courault and Laure Hossard. *Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France.* Remote Sensing, vol. 10, no. 8, 2018.

[Ng *et al.* 2015] J. Ng, F. Yang and L. S. Davis. *Exploiting local features from deep networks for image retrieval.* CoRR, vol. abs/1504.05133, 2015.

[Ng *et al.* 2017] Wai-Tim Ng, Purity Rima, Kathrin Einzmann, Markus Immitzer, Clement Atzberger and Sandra Eckert. *Assessing the potential of Sentinel-2 and Pléiades data for the detection of Prosopis and Vachellia spp. in Kenya.* Remote Sensing, vol. 9, no. 1, 2017.

[Nwe *et al.* 2017] Tin Lay Nwe, Tran Huy Dat and Bin Ma. *Convolutional neural network with multi-task learning scheme for acoustic scene classification.* In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 1347–1350, 2017.

[Nzimande *et al.* 2021] Ntombifuthi Nzimande, Onisimo Mutanga, Zolo Kiala and Mbulisi Sibanda. *Mapping the spatial distribution of the yellowwood tree (Podocarpus henkelii) in the Weza-Ngele forest using the newly launched Sentinel-2 multispectral imager data.* South African Geographical Journal, vol. 103, no. 2, pages 204–222, 2021.

[Ojala *et al.* 2002] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen and S. Huovinen. *Outex - new framework for empirical evaluation of texture analysis algorithms.* In 2002 International Conference on Pattern Recognition, volume 1, pages 701–706 vol.1, 2002.

[Ordóñez & Roggen 2016] Francisco Javier Ordóñez and Daniel Roggen. *Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition.* Sensors, vol. 16, no. 1, 2016.

[Ortiz *et al.* 2013] Sonia Ortiz, Johannes Breidenbach and Gerald Kändler. *Early detection of Bark beetle green attack using TerraSAR-X and RapidEye data.* Remote Sensing, vol. 5, issue 4, pp. 1912-1931, vol. 5, pages 1912–1931, Apr 2013.

[Oumar & Mutanga 2014] Zakariyyaa Oumar and Onisimo Mutanga. *Integrating environmental variables and WorldView-2 image data to improve the prediction and mapping of Thaumastocoris peregrinus (bronze bug) damage in plantation forests.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 87, page 3946, 01 2014.

[Pan *et al.* 2017] Junting Pan, Cristian Canton-Ferrer, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol and Xavier Giró-i-Nieto. *SalGAN: visual saliency prediction with generative adversarial networks.* CoRR, vol. abs/1701.01081, 2017.

[Paoletti *et al.* 2018] M.E. Paoletti, J.M. Haut, J. Plaza and A. Plaza. *A new deep convolutional neural network for fast hyperspectral image classification.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 145, pages 120 – 147, 2018. Deep Learning RS Data.

[Pascal *et al.* 2008] F. Pascal, Y. Chitour, J-P. Ovarlez, P. Forster and P. Larzabal. *Covariance structure maximum-likelihood estimates in compound Gaussian noise: existence and algorithm analysis.* Trans. Sig. Proc., vol. 56, no. 1, pages 34–48, January 2008.

[Pelletier *et al.* 2019] Charlotte Pelletier, Geoffrey I. Webb and François Petitjean. *Deep learning for the classification of Sentinel-2 image time series.* In IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pages 461–464, 2019.

[Pennec *et al.* 2006] X. Pennec, P. Fillard and N. Ayache. *A Riemannian framework for tensor computing.* International Journal of Computer Vision, vol. 66, no. 1, pages 41–66, 2006.

[Pennec 2004] Xavier Pennec. *Probabilities and statistics on Riemannian manifolds: a geometric approach.* Research Report INRIA, Jan 2004.

[Pennec 2006] X. Pennec. *Intrinsic statistics on Riemannian manifolds: basic tools for geometric measurements.* Journal of Mathematical Imaging and Vision, vol. 25, no. 1, pages 127–154, 2006.

[Perronnin & Dance 2007] F. Perronnin and C. Dance. *Fisher kernels on visual vocabularies for image categorization.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, 2007.

[Perronnin & Larlus 2015] F. Perronnin and D. Larlus. *Fisher vectors meet neural networks: a hybrid classification architecture.* In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3743–3752, June 2015.

[Perronnin *et al.* 2010a] F. Perronnin, Y. Liu, J. Sánchez and H. Poirier. *Large-scale image retrieval with compressed Fisher vectors.* In The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pages 3384–3391, 2010.

[Perronnin *et al.* 2010b] F. Perronnin, J. Sánchez and T. Mensink. Improving the Fisher kernel for large-scale image classification, volume 6314 of *Lecture Notes in Computer Science*, pages 143–156. Springer Berlin Heidelberg, 2010.

[Petersen & Pedersen 2008] K. B. Petersen and M. S. Pedersen. *The matrix cookbook*, October 2008. Version 20081110.

[Pham *et al.* 2016] M.-T. Pham, G. Mercier, O. Regniers and J. Michel. *Texture retrieval from VHR optical remote sensed images using the local extrema descriptor with application to vineyard parcel detection.* Remote Sensing, vol. 8, no. 5, page 368, 2016.

[Pham *et al.* 2017] Minh-Tan Pham, Grégoire Mercier and Lionel Bombrun. *Color texture image retrieval based on local extrema features and Riemannian distance.* Journal of Imaging, vol. 3, no. 4, page 43, Oct 2017.

[Picard *et al.* 2010] Rosalind Picard, C. Graczyk, Steve Mann, J. Wachman, L. Picard and L. Campbell. *VisTex vision texture database.* Jan 2010.

[Pires de Lima & Marfurt 2019] Rafael Pires de Lima and Kurt Marfurt. *Convolutional neural network for remote-sensing scene classification: transfer learning analysis.* Remote Sensing, vol. 12, page 86, Dec 2019.

[Popivanov & Miller 2002] I. Popivanov and R.J. Miller. *Similarity search over time-series data using wavelets.* In Proceedings 18th International Conference on Data Engineering, pages 212–221, 2002.

[Potter *et al.* 2012] Christopher Potter, Shuang Li, Shengli Huang and Robert L. Crabtree. *Analysis of sapling density regeneration in Yellowstone National Park with hyperspectral remote sensing data.* Remote Sensing of Environment, vol. 121, pages 61–68, 2012.

[Raileanu & Stoffel 2004] Laura Raileanu and Kilian Stoffel. *Theoretical Comparison between the Gini Index and Information Gain Criteria.* Annals of Mathematics and Artificial Intelligence, vol. 41, pages 77–93, May 2004.

[Rajkomar *et al.* 2018] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew Dai, Nissan Hajaj, Peter Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Gavin Duggan, Gerardo Flores, Michaela Hardt, Jamie Irvine, Quoc Le, Kurt Litsch, Jake Marcus, Alexander Mossin and Jeff Dean. *Scalable and accurate deep learning for electronic health records.* npj Digital Medicine, vol. 1, Jan 2018.

[Regniers *et al.* 2015] O. Regniers, L. Bombrun, D. Guyon, J.-C. Samalens and C. Germain. *Wavelet-based texture features for the classification of age classes in a maritime pine forest.* IEEE Geosc. and Rem. Sens. Lett., vol. 12, no. 3, pages 621–625, 2015.

[Regniers *et al.* 2016] O. Regniers, L. Bombrun, V. Lafon and C. Germain. *Supervised classification of very high resolution optical images using wavelet-based textural features.* IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 6, pages 3722–3735, 2016.

[Regniers 2014] Olivier Regniers. *Méthodes d'analyse de texture pour la cartographie d'occupations du sol par télédetection très haute résolution : application à la fôret, la vigne et les parcs ostréicoles.* Theses, Université de Bordeaux, December 2014.

[Ribeiro *et al.* 2016] Manoel Horta Ribeiro, Bruno Teixeira, Antônio Otávio Fernandes, Wagner Meira and Erickson R. Nascimento. *Complexity-aware assignment of latent values in discriminative models for accurate gesture recognition.* In 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pages 378–385, 2016.

[Roman & Ursu 2016] Anamaria Roman and Tudor Ursu. Multispectral satellite imagery and airborne laser scanning techniques for the detection of archaeological vegetation marks, pages 141–152. Dec 2016.

[Rosu *et al.* 2017] R. Rosu, M. Donias, L. Bombrun, S. Said, O. Regniers and J. P. Da Costa. *Structure tensor Riemannian statistical models for CBIR and classification of remote sensing images.* IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 1, pages 248–260, Jan 2017.

[Ruffini *et al.* 2016] Giulio Ruffini, Ibanez-Soria David, Marta Castellano, Stephen Dunne and Aureli Soria-Frisch. *EEG-driven RNN classification for prognosis of neurodegeneration in at-risk patients.* pages 306–313, Oct 2016.

[Russakovsky *et al.* 2014] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg and Fei-Fei Li. *ImageNet large scale visual recognition challenge.* CoRR, vol. abs/1409.0575, 2014.

[Sahadevan *et al.* 2013] Dinesh Kumar Sahadevan, Srinivasa Sitiraju and Jaswant Sharma. *Radar vegetation index as an alternative to NDVI for monitoring of soybean and cotton.* Sept 2013.

[Said *et al.* 2015a] Salem Said, Lionel Bombrun and Yannick Berthoumieu. *Texture classification using Rao's distance on the space of covariance matrices.* In Geometric Science of Information, volume 9389, pages 371–378, Oct 2015.

[Said *et al.* 2015b] Salem Said, Lionel Bombrun, Yannick Berthoumieu and Jonathan H. Manton. *Riemannian Gaussian distributions on the space of symmetric positive definite matrices.* IEEE Transactions on Information Theory, vol. 63, pages 2153–2170, 2015.

[Said *et al.* 2018] S. Said, H. Hajri, L. Bombrun and B. C. Vemuri. *Gaussian distributions on Riemannian symmetric spaces: statistical learning with structured covariance matrices.* IEEE Transactions on Information Theory, vol. 64, no. 2, pages 752–772, Feb 2018.

[Sainte Fare Garnot & Landrieu 2020] Vivien Sainte Fare Garnot and Loic Landrieu. *Lightweight temporal self-attention for classifying satellite images time series.* In Workshop on Advanced Analytics and Learning on Temporal Data, AALTD, en ligne, Belgium, September 2020.

[Saintonge 2003] F.X Saintonge. *En 2003 et 2004, l'encre et le chancre du châtaignier restent d'actualité. Département de la santé des forêts.* 2003.

[Sakoe & Chiba 1978] H. Sakoe and S. Chiba. *Dynamic programming algorithm optimization for spoken word recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 1, pages 43–49, Feb 1978.

[Salton & Buckley 1988] G. Salton and C. Buckley. *Term-weighting approaches in automatic text retrieval.* Information Processing and Management, vol. 24, no. 5, pages 513–523, 1988.

[Sánchez *et al.* 2013] J. Sánchez, F. Perronnin, T. Mensink and J. Verbeek. *Image classification with the Fisher vector: Theory and practice.* International Journal of Computer Vision, vol. 105, no. 3, pages 222–245, 2013.

[Santini *et al.* 2013] A. Santini, L. Ghelardini, C. De Pace, M. L. Desprez-Loustau, P. Capretti, A. Chandelier, T. Cech, D. Chira, S. Diamandis, T. Gaitniekis, J. Hantula, O. Holdenrieder, L. Jankovsky, T. Jung, D. Jurc, T. Kirisits, A. Kunca, V. Lygis, M. Malecka, Benoit Marçais, S. Schmitz, J. Schumacher, H. Solheim, A. Solla, I. Szabo, P. Tsopelas, A. Vannini, A. M. Vettraino, J. Webber, S. Woodward and J. Stenlid. *Biogeographical patterns and determinants of invasion by forest pathogens in Europe.* New Phytologist, vol. 197, no. 1, pages 238–250, January 2013.

[Segal-Rozenhaimer *et al.* 2020] Michal Segal-Rozenhaimer, Alan Li, Kamalika Das and Ved Chirayath. *Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN).* Remote Sensing of Environment, vol. 237, page 111446, 2020.

[Sibanda *et al.* 2019] Mbulisi Sibanda, Onisimo Mutanga, Timothy Dube, Thulile S Vundla and Paramu L Mafongoya. *Estimating LAI and mapping canopy storage capacity for hydrological applications in wattle infested ecosystems using Sentinel-2 MSI derived red edge bands.* GIScience & Remote Sensing, vol. 56, no. 1, pages 68–86, 2019.

[Simo-Serra *et al.* 2017] Edgar Simo-Serra, Carme Torras and Francesc Moreno-Noguer. *3D human pose tracking priors using geodesic mixture models.* International Journal of Computer Vision, vol. 122, no. 2, pages 388–408, Apr 2017.

[Simonyan & Zisserman 2014] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition.* CoRR, vol. abs/1409.1556, 2014.

[Simonyan *et al.* 2013] K. Simonyan, A. Vedaldi and A. Zisserman. *Deep Fisher networks for large-scale image classification.* In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13, pages 163–171, USA, 2013. Curran Associates Inc.

[Sivic *et al.* 2005] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman. *Discovering objects and their location in images.* In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 1, pages 370–377 Vol. 1, Oct 2005.

[Skalak 1996] David B. Skalak. *The sources of increased accuracy for two proposed boosting algorithms.* In AAAI 1996, 1996.

[Smith 2005] S. T. Smith. *Covariance, subspace, and intrinsic Cramér-Rao bounds.* IEEE Transactions on Signal Processing, vol. 53, no. 5, pages 1610–1630, 2005.

[Souleyman *et al.* 2019] Chaib Souleyman, Mohammed Larabi, Yanfeng Gu, Khadidja Bakhti and Moussa Sofiane Karoui. *Very high resolution image scene classification with capsule network.* Aug 2019.

[Srivastava *et al.* 2011] Anuj Srivastava, Wei Wu, Sebastian Kurtek, Eric Klassen and J. S. Marron. *Registration of functional data using Fisher-Rao metric*, 2011.

[Su *et al.* 2014a] Jingyong Su, Sebastian Kurtek, Eric Klassen and Anuj Srivastava. *Statistical analysis of trajectories on Riemannian manifolds: bird migration, hurricane tracking and video surveillance.* The Annals of Applied Statistics, vol. 8, no. 1, Mar 2014.

[Su *et al.* 2014b] Jingyong Su, Anuj Srivastava, Fillipe D. M. De Souza and Sudeep Sarkar. *Rate-invariant analysis of trajectories on Riemannian manifolds with application in visual speech recognition.* In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 620–627, 2014.

[Sumbul *et al.* 2019] Gencer Sumbul, Marcela Charfuelan, Begum Demir and Volker Markl. *BigEarthNet: A large-scale benchmark archive for remote sensing image understanding.* In IEEE International Geoscience and Remote Sensing Symposium, July 2019.

[Surowiecki 2004] James Surowiecki. The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. Doubleday, May 2004.

[Susto *et al.* 2018] Gian Antonio Susto, Angelo Cenedese and Matteo Terzi. *Chapter 9 - Time-series classification methods: review and applications to power systems data.* In Reza Arghandeh and Yuxun Zhou, editors, Big Data Application in Power Systems, pages 179–220. Elsevier, 2018.

[Sutskever *et al.* 2014] Ilya Sutskever, Oriol Vinyals and Quoc V. Le. *Sequence to sequence learning with neural networks.* In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 31043112, Cambridge, MA, USA, 2014. MIT Press.

[Szegedy *et al.* 2014] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. *Going deeper with convolutions*, 2014.

[Terras 1988] A. Terras. Harmonic analysis on symmetric spaces and applications. Number vol. 1 de Harmonic Analysis on Symmetric Spaces and Applications. Springer-Verlag, 1988.

[Thanh Noi & Kappas 2018] Phan Thanh Noi and Martin Kappas. *Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery.* Sensors, vol. 18, no. 1, 2018.

[Trudel *et al.* 2012] Melanie Trudel, François Charbonneau and Robert Leconte. *Using RADARSAT-2 polarimetric and ENVISAT-ASAR dual-polarization data for estimating soil moisture over agricultural fields.* Canadian Journal of Remote Sensing, vol. 38, no. 4, pages 514–527, 2012.

[Tuzel *et al.* 2006] O. Tuzel, F. Porikli and P. Meer. Region covariance: a fast descriptor for detection and classification, volume 3952 of *Lecture Notes in Computer Science*, pages 589–600. Springer Berlin Heidelberg, 2006.

[Tyler 1987] D. E. Tyler. *A distribution-free M-estimator of multivariate scatter.* The Annals of Statistics, vol. 15, no. 1, pages 234–251, Mar 1987.

[van der Maaten & Hinton 2008] Laurens van der Maaten and Geoffrey Hinton. *Visualizing data using t-SNE.* Journal of Machine Learning Research, vol. 9, no. 86, pages 2579–2605, 2008.

[Vannini *et al.* 2001] Andrea Vannini, Anna Maria and A.M. Vettraino. *Ink disease in chestnuts: Impact on the European chestnut.* For. Snow Landsc. Res, vol. 76, pages 345–350, Jan 2001.

[Vaswani *et al.* 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. *Attention is all you need*, 2017.

[Verbesselt *et al.* 2009] Jan Verbesselt, Andrew Robinson, Christine Stone and Darius Culvenor. *Forecasting tree mortality using change metrics derived from MODIS satellite data.* Forest Ecology and Management, vol. 258, no. 7, pages 1166–1173, 2009.

[Viola & Jones 2001] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages I–511–I–518, 2001.

[Walter & Platt 2013] Jonathan A. Walter and Rutherford V. Platt. *Multi-temporal analysis reveals that predictors of mountain pine beetle infestation change during outbreak cycles*. Forest Ecology and Management, vol. 302, pages 308–318, 2013.

[Wang *et al.* 2013] Jin Wang, Ping Liu, Mary Fenghua She, Saeid Nahavandi and Abbas Z. Kouzani. *Bag-of-words representation for biomedical time series classification*. Biomedical Signal Processing and Control, vol. 8, pages 634–644, 2013.

[Wang *et al.* 2018] Dezhi Wang, Bo Wan, Penghua Qiu, Yanjun Su, Qinghua Guo, Run Wang, Fei Sun and Xincai Wu. *Evaluating the performance of Sentinel-2, Landsat 8 and Pléiades-1 in mapping mangrove extent and species*. Remote Sensing, vol. 10, no. 9, 2018.

[Weisberg 2005] S. Weisberg. Applied linear regression. Wiley Series in Probability and Statistics. Wiley, 2005.

[Williams *et al.* 2002] G. Williams, R. Baxter, Hongxing He, S. Hawkins and Lifang Gu. *A comparative study of RNN for outlier detection in data mining*. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pages 709–712, 2002.

[Williams *et al.* 2006] Ben H. Williams, Marc Toussaint and Amos J. Storkey. *Extracting motion primitives from natural handwriting data*. In Stefanos Kollias, Andreas Stafylopatis, Włodzisław Duch and Erkki Oja, editors, Artificial Neural Networks – ICANN 2006, pages 634–643, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Woo *et al.* 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee and In So Kweon. *CBAM: Convolutional Block Attention Module*, 2018.

[Xia *et al.* 2017] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Xiaoqiang Lu and Liangpei Zhang. *AID: a benchmark data set for performance evaluation of aerial scene classification*. IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pages 3965 – 3981, Feb 2017.

[Yang & Newsam 2010] Y. Yang and S. Newsam. *Bag-of-visual-words and spatial extensions for land-use classification*. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pages 270–279, New York, NY, USA, 2010. ACM.

[Yang & Wu 2006] Qiang Yang and Xindong Wu. *10 challenging problems in data mining research*. International Journal of Information Technology & Decision Making (IJITDM), vol. 05, pages 597–604, Dec 2006.

[Yommy *et al.* 2015] Aiyeola Yommy, Rongke Liu and And Wu. *SAR image despeckling using refined Lee filter*. pages 260–265, Aug 2015.

[Yu & Salzmann 2017] Kaicheng Yu and Mathieu Salzmann. *Second-order convolutional neural networks*. CoRR, vol. abs/1703.06817, 2017.

[Yuan *et al.* 2010] C. Yuan, W. Hu, X. Li, S. Maybank and G. Luo. Human action recognition under log-Euclidean Riemannian metric, pages 343–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[Zafar & Ali 2019] Bushra Zafar and Nouman Ali. *SIRI_WHU Dataset.* 7 2019.

[Zentmyer 1988] G.A. Zentmyer. *Origin and distribution of four species of Phytophthora.* Transactions of the British Mycological Society, vol. 91, no. 3, pages 367–378, 1988.

[Zhang *et al.* 2015] Zhengwu Zhang, Jingyong Su, Eric Klassen, Huiling Le and Anuj Srivastava. *Video-based action recognition using rate-invariant analysis of covariance trajectories.* Mar 2015.

[Zhang *et al.* 2017] Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang and Thomas Corpetti. *Dynamic time warping under limited warping path length.* Information Sciences, vol. 393, pages 91 – 107, July 2017.

[Zhao *et al.* 2016] B. Zhao, Y. Zhong, G. Xia and L. Zhang. *Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery.* IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 4, pages 2108–2123, 2016.

[Zhao *et al.* 2017] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu and Dongya Wu. *Convolutional neural networks for time series classification.* Journal of Systems Engineering and Electronics, vol. 28, no. 1, pages 162–169, 2017.

[Zheng *et al.* 2014] Y. Zheng, Qi Liu, Enhong Chen, Yong Ge and J. L. Zhao. *Time series classification using multi-channels deep convolutional neural networks.* In WAIM, 2014.

List of Publications

Journal Article

1. Sara Akodad, Lionel Bombrun, Junshi Xia, Yannick Berthoumieu and Christian Germain. *Ensemble learning approaches based on covariance pooling of CNN features for high resolution remote sensing scene classification.* Remote Sensing, vol. 12, no. 20, 2020

Conference Papers

1. Sara Akodad, Lionel Bombrun, Charles Yaacoub, Yannick Berthoumieu and Christian Germain. *Image classification based on log-Euclidean Fisher Vectors for covariance matrix descriptors.* In International Conference on Image Processing Theory, Tools and Applications (IPTA), Xi-an, China, Nov 2018

2. Sara Akodad, Solène Vilfroy, Lionel Bombrun, Charles C Cavalcante, Christian Germain and Yannick Berthoumieu. *An ensemble learning approach for the classification of remote sensing scenes based on covariance pooling of CNN features.* In 27th European Signal Processing Conference, 27th European Signal Processing Conference, La Coruña, Spain, September 2019

3. Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Encodage de matrices de covariance par les vecteurs de Fisher log-Euclidien : application á la classification supervisée d'images satellitaires.* In GRETSI, Lille, France, 2019

4. Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Cluster kernel for learning similarities between symmetric positive definite matrix time series.* In 2020 IEEE International Conference on Image Processing (ICIP), pages 3304–3308, Abu Dhabi, United Arab Emirates, October 2020. IEEE

Communications to conference without acts

1. Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Encodage de matrices de covariance par les vecteurs de Fisher log-Euclidien : application à la classification supervisée d'images satellitaires.* In GDR-ISIS/CNES CCT-TSI, Paris, France, 2018

2. Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Architectures hybrides de réseaux de neurones exploitants les statistiques d'ordre 2 sur les sorties des couches convolutives d'un CNN : application à la classification supervisée d'images satellitaires.* In GDR-MADICS, Rennes, France, 2019

3. Sara Akodad, Lionel Bombrun, Yannick Berthoumieu and Christian Germain. *Méthode d'ensemble pour la classification de trajectoires temporelles de matrices symétriques définies positives.* In GDR-ISIS, France, 2020

Abstract: In view of the growing success of second-order statistics in classification problems, the work of this thesis has been oriented towards the development of learning methods in manifold. Indeed, covariance matrices are symmetric positive definite matrices that live in a non-Euclidean space. It is therefore necessary to adapt the classical tools of Euclidean geometry to handle this type of data. To do that, we have proposed to exploit the log-Euclidean metric. This latter allows to project the set of covariance matrices on a tangent plane to the manifold defined at a reference point, classically chosen equal to the identity matrix, followed by a vectorization step to obtain the log-Euclidean representation. On this tangent plane, it is possible to define parametric Gaussian models as well as Gaussian mixture models. Nevertheless, this projection on a single tangent plane can induce distortions. In order to overcome this limitation, we have proposed a GMM model composed of several tangent planes, where the reference points are defined by the centers of each cluster.

In view of the success of neural networks, in particular convolutional neural networks (CNNs), we have proposed two hybrid transfer learning approaches based on the covariance matrix computed locally and globally on the CNN convolutional layers' outputs. The local approach relies on the covariance matrices extracted locally on the first layers of a CNN, which are then encoded by the Fisher vectors computed on their log-Euclidean representation, while for the global approach, a single covariance matrix is computed on the feature maps of the CNN deep layers. Moreover, in order to give more importance to the objects of interest present in the images, we proposed to use a covariance matrix weighted by the saliency information. Furthermore, in order to take advantage of both local and global aspects, these two approaches are subsequently combined in an ensemble strategy.

On another hand, the availability of multivariate time series has aroused the interest of the remote sensing community and more generally of machine learning researchers for the development of new learning strategies dedicated to supervised classification. In particular, methods based on the calculation of point-to-point distance between series. Moreover, two series belonging to the same class can evolve in different ways, which can induce temporal distortions (translation, compression, dilation, etc.). To avoid this, warping methods allow to align the time series. In order to extend this approach to time series of covariance matrices, while ensuring invariance to the re-parametrization of the series, we were interested in the TSRVF representation. In the same context, several ensemble methods have been proposed in the literature, including TCK, which relies on similarity computation to classify time series. We have proposed to extend this strategy to covariance matrices by introducing the SO-TCK approach which relies on the log-Euclidean representation of such matrices.

Finally, the last axis of this thesis concerns the modeling of temporal trajectories of signals measured by the radar (Sentinel 1) and optical (Sentinel 2) sensors. In particular, we are interested in the forestry problem of the chestnut ink disease in the Montmorency forest. For this purpose, we developed classification and regression models to predict a health status score from the covariance matrix computed on multi-temporal radiometric attributes.

Keywords: Supervised classification, remote sensing, multivariate time series, ensemble learning, second-order statistics, Sentinel 1 & 2.

Résumé: Devant le succès grandissant des statistiques du second ordre dans les problèmes de classification, les travaux de cette thèse se sont orientés vers le développement de méthodes d'apprentissage sur variété. En effet, les matrices de covariance sont des matrices symétriques définies positives qui vivent dans un espace non Euclidien. Il est donc nécessaire de réadapter les outils classiques de la géométrie Euclidienne pour manipuler ce type de données. Pour ce faire, nous avons proposé d'exploiter la métrique log-Euclidienne. Celle-ci permet de projeter l'ensemble des matrices de covariance sur un plan tangent à la variété défini à un point de référence, classiquement choisi égal à la matrice identité, suivi d'une étape de vectorisation pour obtenir la représentation log-Euclidienne. Sur ce plan tangent, il est possible de définir des modèles paramétriques Gaussien ainsi que des modèles de mélange de Gaussiennes. Néanmoins, cette projection sur un unique plan tangent peut induire des distorsions. Afin de limiter cela, nous avons proposé un modèle de GMM composé de plusieurs plans tangents, où les points de référence sont définis par les centres de chaque cluster.

Au vu de la réussite remportée par les réseaux de neurones, en particulier les réseaux de neurones convolutifs (CNN), nous avons proposé deux approches hybrides d'apprentissage par transfert basées sur la matrice de covariance calculée de façon locale et globale sur les sorties des couches convolutives d'un CNN. D'une part, l'approche locale s'appuie sur les matrices de covariance extraites localement sur les premières couches d'un CNN, qui sont ensuite encodées par les vecteurs de Fisher calculés sur leur représentation log-Euclidienne. Tandis que pour l'approche globale, une seule matrice de covariance est calculée sur les cartes de caractéristiques des couches profondes d'un CNN. De plus, afin de donner une plus grande importance aux objets d'intérêt présents dans les images, nous avons proposé d'utiliser une matrice de covariance pondérée par l'information de saillance. Par ailleurs, afin de tirer profit des aspects local et global, ces deux approches sont par la suite combinées dans une stratégie d'ensemble.

D'autre part, la disponibilité des séries temporelles multivariées a suscité l'intérêt de la communauté de la télédétection et plus généralement du machine learning pour l'élaboration de nouvelles stratégies d'apprentissage pour la classification supervisée, notamment les méthodes basées sur le calcul de distance point à point entre les séries. Par ailleurs, deux séries appartenant à la même classe peuvent évoluer de façons différentes, ce qui peut induire des distorsions temporelles (translation, compression, dilatation, etc.). Pour s'affranchir de cela, le "warping" permet d'aligner les séries temporelles. Afin d'étendre cette approche pour des séries temporelles de matrices de covariance, tout en assurant l'invariance à la reparamétrisation des séries, nous nous sommes intéressés à la représentation TSRVF. Dans le même contexte, plusieurs méthodes d'ensemble ont été proposées dans la littérature, notamment le TCK, qui repose sur le calcul de similarité afin de classifier les séries temporelles. Nous avons proposé d'étendre cette stratégie aux matrices de covariance en introduisant l'approche SO-TCK qui s'appuie sur la représentation log-Euclidienne de ces matrices.

Finalement, le dernier axe de cette thèse concerne la modélisation de trajectoires temporelles des signaux mesurés par les capteurs radar (Sentinel 1) et optique (Sentinel 2). En particulier, nous nous sommes intéressés au problème sylvosanitaire de la maladie de l'encre du châtaignier sur la forêt de Montmorency. Pour cela, nous avons développé des modèles de classification et de régression afin de prédire une note d'état sanitaire à partir de la matrice de covariance calculée sur les attributs radiométriques multitemporels.

Mots clés: Classification supervisée, télédétection, séries temporelles multivariées, méthodes d'ensemble, statistiques du second ordre, Sentinel 1 & 2.

Unité de recherche

Laboratoire IMS, Groupe Signal et Image

351, avenue de la libération - 33405 Talence cedex