Identification de vêtements dans des images : de la segmentation d'instances à la classification multi-étiquettes faiblement supervisée

Warren Jouanneau

▶ **To cite this version:**

Warren Jouanneau. Identification de vêtements dans des images : de la segmentation d'instances à la classification multi-étiquettes faiblement supervisée. Traitement des images [eess.IV]. Université de Bordeaux, 2023. Français. NNT : 2023BORD0020 . tel-04089316

HAL Id: tel-04089316
https://theses.hal.science/tel-04089316

Submitted on 4 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE
SPÉCIALITÉ INFORMATIQUE

Par **Warren JOUANNEAU**

Identification de vêtements dans des images :
de la segmentation d'instances à la classification multi-étiquettes faiblement supervisée

Sous la direction de : **Aurelie BUGEAU**
Co-directeur : **Nicolas PAPADAKIS**

Soutenue le 07/02/2023

Membres du jury :

Baudouin DENIS DE SENNEVILLE	Directeur de Recherche	CNRS - Bordeaux	Président
Gabriele FACCIOLO	Professeur des universités	ENS Paris-Saclay	Examinateur
Sylvie TREUILLET	Maitresse de conférence	Université d'Orléans	Rapporteure
Pierre HELLIER	Cadre scientifique	Interdigital	Rapporteur
Aurélie BUGEAU	Professeure des universités	Université de Bordeaux	Directrice
Nicolas PAPADAKIS	Directeur de Recherche	CNRS - Bordeaux	Directeur
Laurent VEZARD	Cadre scientifique	Lectra	Invité
Marc PALYART	Cadre scientifique	Malt	Invité

Identification de vêtements dans des images :
de la segmentation d'instances à la classification multi-étiquettes faiblement supervisée

Résumé :

Pour l'industrie textile et de la mode, les images de vêtements sont essentielles à différentes fins : décisionnelles, collaboratives, promotionnelles, pour tous les types de produits : intemporels, saisonniers, tendance, et à toutes les phases de leur cycle de vie : de la conception à la vente. Il est donc nécessaire de faciliter leur accès et leur recherche parmi un grand nombre d'images candidates. Cela repose le plus souvent sur l'apposition de mots clefs afin de les indexer et de les référencer. Il est alors fondamental d'automatiser la saisie manuelle de ces mots clefs lors de tout dépôt ou collecte d'images. Cette opération chronophage est source d'erreurs ou de descriptions incomplètes des données. De plus, elle est irréalisable pour des cas de collecte ou d'analyse à grande échelle.

En apprentissage supervisé, l'attribution d'une étiquette (*c.-à-d.* un mot clef) se transpose en un problème de classification. Sans information sur la composition de l'image, la classification se fait sur l'image dans sa globalité. Pour avoir des prédictions plus fines, l'approche retenue dans cette thèse se décompose en deux étapes : localisation de chacun des vêtements pour les différencier, puis caractérisation plus fine du contenu de chaque détection. La localisation et la caractérisation sont indépendantes et peuvent être effectuées par différents types de méthodes.

Les méthodes de segmentation d'instances, qui sont la forme de localisation la plus fine, ont été retenues. En effet, ces approches permettent d'estimer, sous la forme d'un masque, l'ensemble des pixels constituant un objet donné. Elles offrent de plus l'avantage de différencier les objets d'une même étiquette, ce qui permet une caractérisation indépendante par la suite. Une étude des méthodes de segmentation d'instances a ainsi été réalisée pour le cas particulier des images de vêtements.

Nous disposons de données annotées et recourrons donc à des méthodes de segmentation issues de l'apprentissage supervisé, qui reposent sur un étiquetage connu a priori et considéré comme vérité à reproduire. Ces méthodes peuvent alors être évaluées en mesurant l'écart des prédictions à ces annotations. Après avoir montré que les approches classiques de validation de masques de segmentation ne sont pas adaptées à nos cas d'usages, un protocole d'évaluation à trois niveaux a été proposé : global, contour, contenu, afin de discriminer correctement les architectures retenues.

Les travaux sur la caractérisation des vêtements se sont concentrés sur trois caractéristiques : type fin du vêtement (*ex.* jean, pantalon de costume, jogging *etc.*), motif tissu (*ex.* rayure, pois, uni *etc.*), couleur dominante (*ex.* saumon, fuchsia, corail, *etc.*). Pour le type de vêtement et le motif tissu, une méthode de classification a été développée pour exploiter les images enrichies de la zone du vêtement. Cependant, certains motifs tissus apparaissent ou disparaissent selon la résolution (*ex.* rayure fine). Une approche extrayant des patchs à résolution native a alors été proposée. Pour la couleur dominante, les problématiques de dénomination et de partitionnement d'espace de couleur ont été abordées. Ces travaux nous ont permis d'obtenir un processus global d'attribution d'étiquette reversé à Lectra (intégrant segmentation et caractérisation).

Enfin, il est parfois difficile de classer une image avec une unique étiquette. Cela est notamment le cas lorsque plusieurs tissus et motifs composent un même vêtement. La classification multi-étiquettes des images est alors plus adaptée. L'application des méthodes d'apprentissage supervisé à ce type de problème nécessite cependant des données dédiées, dont la collecte et l'annotation sont complexes. Pour répondre à ce problème, nous avons développé une méthode faiblement supervisée, nécessitant seulement une étiquette positive connue par image. L'entraînement repose alors sur une nouvelle stratégie efficace pour estimer des exemples négatifs pour chaque classe.
Mots-clés : Traitement d'image, Apprentissage automatique, Classification, Segmentation, Mode, Vêtement

---

Clothing identification in images :
from instance segmentation to weakly supervised multi-label classification

Abstract:

For the textile and fashion industry, clothing images are essential for different purposes: decision-making, collaborative, promotional, for all types of products: timeless, seasonal, trendy, and at all phases of their life cycle : from design to sale. It is therefore necessary to facilitate their access and search among numerous candidate images. This is most often based on keywords to index and reference them. It is therefore essential to automate the manual entry of these keywords when submitting or collecting images. This time-consuming operation is a source of errors or incomplete descriptions of the data. Moreover, it is impractical for large-scale collection or analysis.

In supervised learning, the assignment of a label (*i.e.* a keyword) is a classification problem. Without information on the composition of the image, the classification is done on the whole image. In order to have finer predictions, the approach adopted in this thesis is decomposed in two steps: localization of each garment to differentiate them, then finer characterization of the content of each detection. Localization and characterization are independent and can be performed by different types of methods.

Instance segmentation methods, which are the finest form of localization, have been retained in the thesis. Indeed, these approaches make it possible to estimate, in the form of a mask, all the pixels constituting a given object. They also offer the advantage of differentiating objects with the same label, which later allows for independent characterization. A study of the instance segmentation methods has been realized for the particular case of clothing images.

We dispose of annotated data and consider supervised learning segmentation methods, which are based on a labeling known a priori and considered as truth to be reproduced. These methods can then be evaluated by measuring the deviation of predictions from these annotations. After showing that classical approaches to validate segmentation masks are not adapted to all cases, a three-level evaluation protocol (global, contour, content) has been proposed in order to correctly discriminate the selected architectures.

The work on garment characterization focused on three characteristics: fine type of garment (*e.g.* jeans, suit pants, sweatpants, *etc.*), fabric pattern (*e.g.* striped, polka dot, plain, *etc.*), dominant color (*e.g.* salmon, fuchsia, coral, *etc.*). For garment type and fabric pattern, a classification method has been developed to exploit the enriched images of the garment area. However, some fabric patterns appear or disappear depending on the resolution (*e.g.* fine stripe). An approach that extracts patches at native resolution was then proposed. For the dominant color, the problems of naming, taxonomy matching and color space partitioning have been addressed. This work allowed us to obtain a global process of label attribution transferred to Lectra (integrating segmentation and characterization).

Finally, it is sometimes difficult to classify an image with a single label. This is particularly the case when several fabrics and patterns compose a single garment. The multi-label classification of images is then more appropriate. However, the application of supervised learning methods to this type of problem requires dedicated data, whose gathering and annotation are complex. To address this issue, we have developed a weakly supervised method, requiring only one known positive label per image. The training then relies on a new efficient strategy to estimate negative examples for each class.

Keywords: Image processing, Machine learning, Classification, Segmentation, Fashion, Clothing

---

Unité de recherche

LaBRI, UMR 5800, Université de Bordeaux, 33400 Talence, France.

IMB, UMR 5251, Université de Bordeaux, 33400 Talence, France.

LECTRA, R&D logicielle, 33610 Cestas, France.

Remerciements

Trois années éprouvantes viennent de s'écouler et s'achèvent plaisamment. Je remercie Baudoin Denis de Senneville, Gabriele Facciolo, Sylvie Treuillet et Pierre Hellier d'avoir accepté de participer au jury et pour vos retours que j'ai grandement appréciés. Je souhaiterais alors associer à cette réussite tous ceux qui ont rendu possible ces travaux par leur soutien ou leur bienveillance.

Je remercie Lectra et l'ANRT d'avoir financé mes travaux aux travers d'une convention CIFRE. Merci à tous les collaborateurs de Lectra qui m'ont apporté, lors de repas, de cafés, de pots, de parties de tarots, la chaleur humaine nécessaire pour conserver mon enthousiasme. Je tiens alors à remercier chaudement Laurent Vezard et Marc Palyart pour la confiance qu'ils m'ont accordée, les opportunités qu'ils m'ont offertes et plus globalement pour leur gentillesse et leur bon vivre.

Je remercie le LaBRI pour son accueil. Merci aux doctorants du laboratoire pour votre soutien et nos conversations lors de groupe de travail, de pause-café et lors de biens d'autres évènements. Je tenais à remercier particulièrement Aurélie Bugeau et Nicolas Papadakis, ma directrice et mon directeur de thèse, pour leurs investissements et leurs nombreux conseils qui m'ont permis entre autre de gagner en efficacité et en rigueur. En regardant derrière moi le chemin parcouru, j'ai vraiment le sentiment d'avoir évolué grâce à vous.

Plus personnellement, je remercie ma famille de m'avoir soutenu dans ma poursuite d'étude, d'avoir soutenu mes choix. Enfin, merci à Fanny Robledo-Garcia. Tu m'as apporté tellement qu'il serait difficile de l'appréhender par ces quelques mots. Merci d'avoir compris l'importance que cela avait pour moi et d'alors m'avoir épaulé. Merci de m'avoir apporté la stabilité, le support moral et l'aide nécessaire pour entretenir et concrétiser mes efforts.

À vous, Merci !

Table des matières

Liste des acronymes

B_{IoU} ***B**ondary IoU*

$l^\star a^\star b^\star$ espace de couleurs $l^\star a^\star b^\star$ (***l**uminance **a**lpha **b**eta*)

ACWS ***A**pparel **C**lassifcation **W**ith **S**yle*

ANRT Association nationale de la recherche et de la technologie

AP ***A**verage **P**recision*

API interface de programmation d'application (***A**pplication **P**rogramming **I**nterface*)

apprentissage PU apprentissage positif et non étiqueté (***P**ositive and **U**nlabeled learning*)

Azure plate-forme applicative en nuage de Microsoft

AzureML *Microsoft **Azure** **M**achine **L**earning service*

BCE entropie croisée binaire (***B**inary **C**ross **E**ntropy*)

BJ ***B**oundary Jaccard*

CE entropie croisée (***C**ross **E**ntropy*)

CIE commission internationale sur l'éclairage

CIFRE Convention industrielle de formation par la recherche

CNN Réseau de neurones à couche de convolution (***C**onvolutional **N**eural **N**etwork*)

COCO *Microsoft **C**ommon **O**bject in **CO**ntext*

DBSCAN ***D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise*

DNN réseau de neurones profond (***D**eep **N**eural **N**etwork*)

EMD distance de transport optimal, aussi appelée EMD (***E**arth **M**over's **D**istance*)

FN faux négatifs

FP faux positifs

FPN ***F**eature **P**yramid **N**etwork*

HOG histogramme de gradient orienté (***H**istrogram of **O**riented **G**radients*)

HSV espace de couleurs HSV (***H**ue **S**aturation **V**alue*)

HTC ***H**ybrid **T**ask **C**ascade*

IMB institut de mathématiques de Bordeaux

IoU intersection sur l'union (***I**ntersection **O**ver **U**nion*)(***I**ntersection **O**ver **U**nion*)

LaBRI laboratoire Bordelais de recherche en informatique

mAP ***m**ean **A**verage **P**recision*

MIL apprentissage d'instances multiples (***M**ultiple **I**nstance **L**earning*)

MLP Réseau de neurones multicouche (***M**ulti **L**ayer **P**erceptron*)

MS R-CNN ***M**ask **S**coring R-CNN*

NMS ***N**on-**M**axima **S**uppression*

R-CNN réseau de neurones à couche de convolution à partir de région (***R**egion based **C**onvolutional **N**eural **N**etwork*)

ReLU unité linéaire rectifiée (***Re**ctified **L**inear **U**nit*)

REST ***Re**presentational **S**tate **T**ransfer*

RGB espace de couleurs RGB(***R**ed **G**reen **B**lue*)

RLE ***R**un **L**ength **E**ncodding*

ROI région d'intérêt (***R**egion **O**f **I**nterest*)

ROLE estimation régularisée des étiquettes en ligne (***R**egularized **O**nline **L**abel **E**stimation*)

RPN réseau de proposition de région (***R**egion **P**roposal **N**etwork*)

SGD Algorithme du gradient stochastique (***S**tochastic **G**radient **D**escent*)

SIFT ***S**cale-**I**nvariant **F**eature **T**ransform*

SSD ***S**ingle **S**hot multibox **D**etector*

SURF ***S**peeded-**U**p **R**obust **F**eatures*

SVM machine à vecteurs de support (***S**upport **V**ector **M**achine*)

TN vrais négatifs

TP vrais positifs

VOC *Pascal **V**isual **O**bject **C**lasses*

VSE ***V**isual **S**emantic **E**mbedding*

WN faiblement négative (***W**eak **N**egative*)

WTBI ***W**here **T**o **B**uy **I**t*

YOLO ***Y**ou **O**nly **L**ook **O**nce*

Chapitre 1

Introduction

Les vêtements et l'habillement sont intimement liés à la nature humaine. Conçus dans un premier temps par l'être humain pour lutter contre les éléments, ils ont depuis lors évolué avec lui. L'être humain est fondamentalement social, les vêtements sont alors la manifestation de notre désir d'appartenance et véhiculent des messages que nous souhaitons adresser au groupe. Ils sont alors à l'échelle collective, des marqueurs historiques, symboles de nos civilisations et vecteur de nos cultures.

Nous observons alors que globalement les vêtements ont évolué avec la société. Leurs productions et leurs diffusions ont suivi les différentes révolutions techniques. Pour ne citer que quelques exemples, les premiers outils, entre autre l'aiguille, ont introduit la couture. L'élevage a permis le tissage. Le commerce a diversifié les matières et pigments. Puis, la révolution industrielle a facilité le transport et automatisé certains procédés manuels. Enfin, l'informatisation et la numérisation ont accéléré les communications et ont permis d'optimiser toutes les étapes du cycle de vie d'un produit.

Tout d'abord limités en production, diffusion et diversité par la disponibilité locale des matières, nous observons l'éparpillement de l'ensemble des activités de la chaîne de valeur à travers le globe. À l'aune du marché globalisé, nos chaînes d'approvisionnement sont sans cesse optimisées afin de produire et de rendre disponible en tout temps, en tous lieux, ces productions. Cette accessibilité atteint son paroxysme avec la vente en ligne. C'est alors environ un vêtement sur cinq [1] qui a été acheté sur internet en 2020. Les projections prédisent un accroissement de cette proportion [2].

Ces optimisations sont conçues pour répondre au mieux à une demande qui a muté avec l'avènement d'internet. Précédemment, les styles vestimentaires et les besoins étaient forte-

1. commonthreadco.com/blogs/coachs-corner/fashion-ecommerce-industry-trends
2. fr.fashionnetwork.com/news/Le-marche-mondial-de-l-habillement-va-progresser-de-3-9-par-an-d-ici-2025,1182400.html

ment déterminés par les frontières physiques de nos territoires. Aujourd'hui, nos influences s'expriment au travers de nos groupes et réseaux à travers internet. Pour communiquer des informations visuelles, nous avons recours aux images. Ce sont plus de 3,5 millions d'images qui ont été échangées par jour en 2016 [3] sur les principaux réseaux sociaux. Un utilisateur sur trois [4] aurait acheté un vêtement après l'avoir vu sur le réseau social Instagram. Pour concrétiser les ventes, les images sont donc essentielles. Les marques présentent ainsi leurs articles avec huit images en moyenne [5] sur les sites de vente.

De nos jours, nous avons franchi une étape dans l'informatisation des procédés. En effet, l'informatisation des processus nous a permis de repenser la collecte et la gestion des flux de données. Ces données toujours plus abondantes sont désormais considérées comme une ressource devant être exploitée. De cette exploitation résulte de nombreuses optimisations et automatisations que l'on pensait précédemment impossibles et réservées à l'intellect humain. Avec l'augmentation des puissances de calculs et de la disponibilité des données, nous avons pu développer différentes méthodes. Ces méthodes nous permettent d'extraire des connaissances à partir des données. Elles peuvent être descriptives ou prescriptives à des fins décisionnelles ou pour remplacer la réflexion humaine.

L'industrie textile et de la mode cherche donc naturellement à se doter de telles méthodes. Les acteurs de la mode, comme toute industrie, manipulent une grande quantité de données. Les images de vêtements en sont une modalité. L'image est au cœur des échanges et de la production textile, pas seulement comme communication aux consommateurs, mais également comme communication entre les différentes parties prenantes. Elles ont une forte valeur à tous les niveaux du cycle de vie d'un produit. Il est donc nécessaire de faciliter leur accès et leur recherche parmi un grand nombre d'images candidates. Cela repose le plus souvent sur l'apposition manuelle de mots clefs afin de les indexer. L'automatisation de cette étape fastidieuse permettrait ainsi une économie considérable de temps et permettrait aux différents acteurs de se concentrer sur des tâches au cœur de leurs métiers. Elle permettrait aussi des analyses à grande échelle préalablement impossibles.

Nous proposons ici de répondre à cette problématique d'automatisation d'apposition de mots clefs décrivant des vêtements dans des images. Pour comprendre les enjeux, nous présentons dans la section 1.1 le contexte industriel et le cycle de vie d'un produit de la mode à travers Lectra, un acteur majeur de l'industrie textile. Nous précisons la problématique dans la section 1.2. Puis, nous abordons le contexte scientifique dans lequel s'inscrit nos travaux dans la section 1.3. Enfin, dans la section 1.4 nous décrivons notre approche du problème et listons

3. kleinerperkins.com/perspectives/2016-internet-trends-report/
4. marketingweek.com/why-brands-with-a-fashion-focus-are-most-likely-to-boost-sales-on-instagram/
5. pathedits.com/blogs/tips/product-photography-standards-how-many-images-do-you-need-to-sell-apparel

les contributions de cette thèse.

1.1 Contexte industriel : production de vêtements

Les travaux présentés dans ce document sont issus d'une collaboration entre Lectra, le Laboratoire Bordelais de recherche en Informatique (LaBRI), l'Institut de Mathématiques de Bordeaux (IMB) et l'Association nationale de la recherche et de la technologie (ANRT) au travers d'une Convention industrielle de formation par la recherche (CIFRE). Lectra est une entreprise française créée en 1973. Lectra est un leader mondial des solutions technologiques intégrées pour les entreprises de l'industrie textile. Lectra propose des équipements et des solutions logicielles couvrant toutes les étapes du cycle de vie d'un article textile. On peut par exemple citer les différents logiciels de conception, de gestion de produits pour les secteurs de l'habillement, de l'ameublement et de l'automobile.

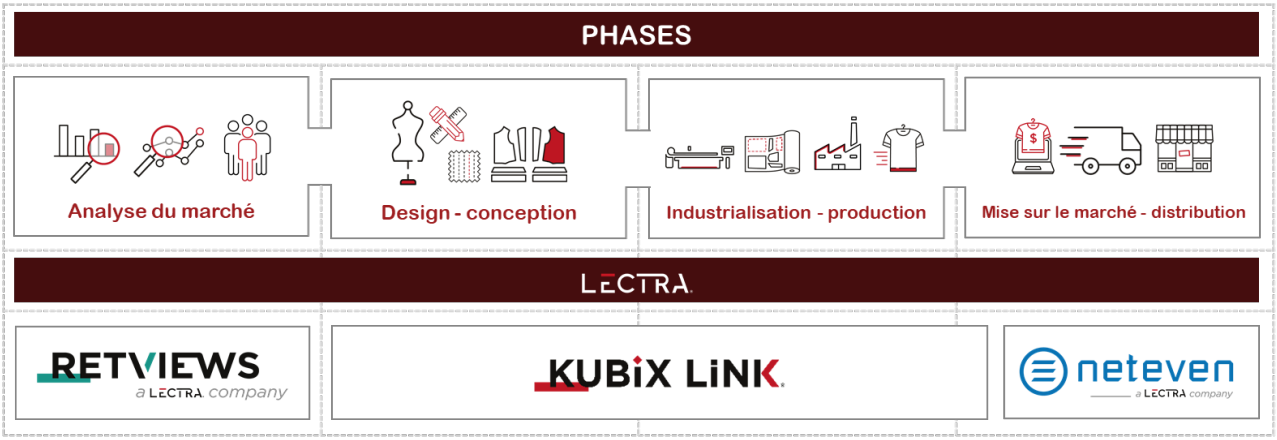


FIGURE 1.1 – Phases de la vie d'un produit textile avec des exemples de solution Lectra.

De façon simplifiée, à partir d'une analyse des besoins et du marché, un vêtement passe par différentes phases (*fig.* 1.1) de sa conception à sa commercialisation. La solution Retviews proposée par Lectra permet à un client de comparer son offre à celle de la concurrence. Ces analyses obtenues par collecte et traitement de données présentes sur internet vont fournir les indicateurs nécessaires aux clients pour adapter leurs stratégies. Dans Retviews, Le traitement de ces données s'appuie fortement sur le texte (descriptions, mots clefs, *etc.*) d'articles de vêtement collectés. Calculer de nouvelles statistiques ou améliorer celles existantes en utilisant les images permettrait d'enrichir l'analyse.

Forts de cette analyse du marché, les différents acteurs peuvent définir une collection de vêtements. Différentes images peuvent alors être échangées et conservées comme source d'inspiration. La phase de conception aboutit sur la création d'un *technical pack* par vêtement. Ce

technical pack contient de multiples informations, par exemple le patron, des images d'exemples (vêtement, matière, couleur, *etc.*), des photos d'essayage. Ce document permet la communication entre designer et manufacture. La solution Kubix Link, configurable pour répondre aux différents besoins, permet de gérer et d'éditer de façon collaborative ces divers types de documents. Lorsqu'un utilisateur dépose une image dans Kubix Link, il peut lui attribuer des mots clefs manuellement. Ces mots clefs permettent d'indexer les images dans un moteur de recherche et de les retrouver pour de futures utilisations. Cependant, les utilisateurs peuvent omettre des mots clefs, *c.-à-d.* ne pas être assez exhaustif dans la description ou commettre des erreurs, *c.-à-d.* se tromper et attribuer les mauvais mots clefs. Ces images peuvent donc s'avérer difficilement retrouvables par recherche.

Enfin, les articles vont être produits. Les pièces vont être découpées et les vêtements assemblés. Un ensemble de supports à la vente et *marketing* vont être produits, par exemple des images de présentation de l'article. Puis, le vêtement produit est disponible sur le marché, dans des magasins physiques ou sur des sites de vente en ligne. L'offre Neteven aide les clients à organiser leur distribution en vente directe aux consommateurs. Elle permet, à partir d'une plateforme unique, d'alimenter différentes *marketplaces* mondiales et de centraliser les informations de ventes. La solution doit alors s'adapter et remplir les fiches produit dans le format des différentes *marketplaces*. Cependant, les informations initialement disponibles pour remplir ces fiches produit peuvent être incomplètes, certains champs n'étant pas renseignés. Les images pourraient donc servir à compléter les informations manquantes.

1.2 Problématique : attribuer des mots clefs à des images

Pour les différents clients de Lectra, les images de vêtements ont une forte valeur à tous les niveaux du cycle de vie d'un produit. Elles se retrouvent donc naturellement dans la plupart de ses offres et solutions. Attribuer correctement des mots clefs décrivant des images de vêtements est alors essentiel. Automatiser cette attribution de mots clefs permettrait d'adresser les problématiques d'analyse de vêtement à grande échelle, de garantir de retrouver des images dans un moteur de recherche et de compléter des informations manquantes sur la description de produits. Cette automatisation garantit l'exhaustivité et permet de contrôler les erreurs. Afin de répondre aux différents cas d'utilisation, l'attribution automatique de mots clefs doit être encapsulée au sein d'une brique fonctionnelle exploitable par les différentes solutions de Lectra.

Cette brique fonctionnelle peut être composée de plusieurs méthodes. Ces méthodes doivent alors permettre d'extraire des mots clefs décrivant des vêtements à partir d'une image. Ces mots clefs sont donc des caractéristiques des vêtements contenus dans l'image (*fig.* 1.2). Ces

FIGURE 1.2 – Exemple de mots clefs pouvant être associés à une image.

caractéristiques peuvent être par exemple le type du vêtement (un pantalon, un t-shirt, *etc.*). Ils peuvent aussi décrire d'autres éléments plus fins, comme le motif (rayure, uni, *etc.*) ou la couleur du tissu d'un vêtement. Les mots clefs souhaitables par les différentes utilisations sont susceptibles d'évoluer dans le temps. Ils peuvent devenir plus nombreux, plus précis, s'appuyer sur d'autres caractéristiques. Une future utilisation pourrait aussi être l'essayage virtuel. La solution proposée doit donc bien différencier tous les vêtements contenus dans l'image et permettre leur extraction. Elle doit de plus pouvoir être facilement adaptée et enrichie avec les changements de besoin.

Dans un contexte industriel, tout processus automatisé doit répondre à certaines contraintes et faire preuve de sa valeur ajoutée. La question du coût est un enjeu capital. Les coûts d'usage direct peuvent être relatifs à la complexité des algorithmes, cette complexité ayant un impact concret sur le temps et la puissance de calcul nécessaire. Les coûts d'usage indirect sont liés, par exemple, à la mise en place de procédés de vérification et de correction métier. Il est donc primordial d'avoir la maîtrise des cas d'erreurs et du périmètre d'utilisation, pour faciliter les traitements postérieurs. Ces aspects sont à prendre en compte lors de toute réalisation, ici la caractérisation de vêtements, afin de garantir une qualité générale de service.

Enfin, obtenir un processus automatisé peut s'appuyer sur l'exploitation de données existantes. Dans ce contexte, il est nécessaire de recourir à une collecte et un pré-traitement des données. La collecte peut introduire des problématiques de réconciliation des sources de données et d'homogénéisation des informations contenues. Afin de conduire des expérimentations et développer une solution, les données peuvent être enrichies manuellement d'informations complémentaires. Ces informations complémentaires peuvent entre autre servir à valider une solution. Globalement, ces étapes complexes demandent des ressources et du temps d'opéra-

teurs, elles sont donc coûteuses. Pouvoir simplifier ou s'abstraire de ses étapes permettrait de limiter l'effort à fournir pour obtenir des données exploitables.

1.3 Contexte scientifique : vision assistée par ordinateur

L'ensemble des problématiques et des méthodes pour analyser des images sont regroupées au sein du domaine de « la vision assistée par ordinateur » (*computer vision*). Les méthodes d'apprentissage ont pris une place prédominante dans ce domaine. Ces méthodes consistent à utiliser des données pour entraîner des modèles prédictifs. Traditionnellement, les méthodes de traitement d'images pouvaient reposer sur un pré-calcul d'informations (nommées descripteurs de l'image) et la définition de règles. Les modèles tendent désormais à induire ces descripteurs et règles automatiquement lors de l'entraînement. Précédemment, plus les connaissances incorporées aux systèmes automatisés étaient fines et exhaustives et plus, ils étaient performants. Désormais, l'automatisation avec l'apport de l'apprentissage s'abstrait de plus en plus de connaissances métiers a priori.

L'adoption et l'amélioration de ces méthodes par la communauté est rendue possible par la disponibilité des données, l'évolution des infrastructures de stockage, l'augmentation des puissances de calcul. De telles méthodes sont par exemple les réseaux de neurones. On appelle alors apprentissage profond l'entraînement d'un réseau de neurones profond (*DNN: **D**eep **N**eural **N**etwork*). De multiples avancées ont dû être réalisées avant qu'ils soient exploitables et afin d'obtenir le niveau de performance qu'on leur connaît aujourd'hui, par exemple les travaux de Fukushima en 1975. La définition des architectures s'est progressivement codifiée avec un certain nombre de couches spécialisées. Par exemple, LeCun et al. en 1998 ont proposé un réseau de neurones à couche de convolution (*CNN: **C**onvolutional **N**eural **N**etwork*) avec des couches de convolution, de regroupement maximum (*max pool*) et de mise à plat (*flatten*). Krizhevsky et al. en 2017 ont alors produit le premier CNN reprenant ces codes, couches et structures, surpassant les méthodes dites « traditionnelles »- dans une compétition (Imagenet [Deng et al., 2009]). Aujourd'hui, les DNN et les CNN sont incontournables dans la plupart des tâches de traitement d'images. Ils ont entre autre permis d'adresser des tâches irréalisables précédemment.

Ces méthodes répondent à de nombreuses problématiques de l'industrie de la mode et du textile [W.-H. Cheng et al., 2021] impliquant des problèmes scientifiques encore ouverts. Les enjeux scientifiques majeurs abordés dans cette thèse concernent ainsi : la distinction et la localisation d'éléments dans des images, l'évaluation de méthode de localisation, la classification à partir de taxonomies incompatibles et l'apprentissage faiblement supervisé pour répondre à

une problématique de données partiellement étiquetées.

1.4 Contributions et organisation du manuscrit

Nous proposons d'utiliser des méthodes d'apprentissage performantes issues du domaine de la vision assistée par ordinateur. Notre processus global se décompose en deux étapes (*fig.* 1.3) : la localisation de chacun des vêtements pour les différencier, puis la caractérisation plus fine du contenu de chaque détection. La localisation et la caractérisation sont indépendantes et peuvent être effectuées par différents types de méthodes. Ces étapes peuvent être conçues, développées et améliorées indépendamment l'une de l'autre. Entre autre, disposer d'une étape de localisation permet de caractériser les vêtements par de multiples méthodes issues de différents domaines. Elles peuvent être développées au fur et à mesure de l'évolution des besoins et s'ajouter au processus sans impacter les méthodes précédemment intégrées.

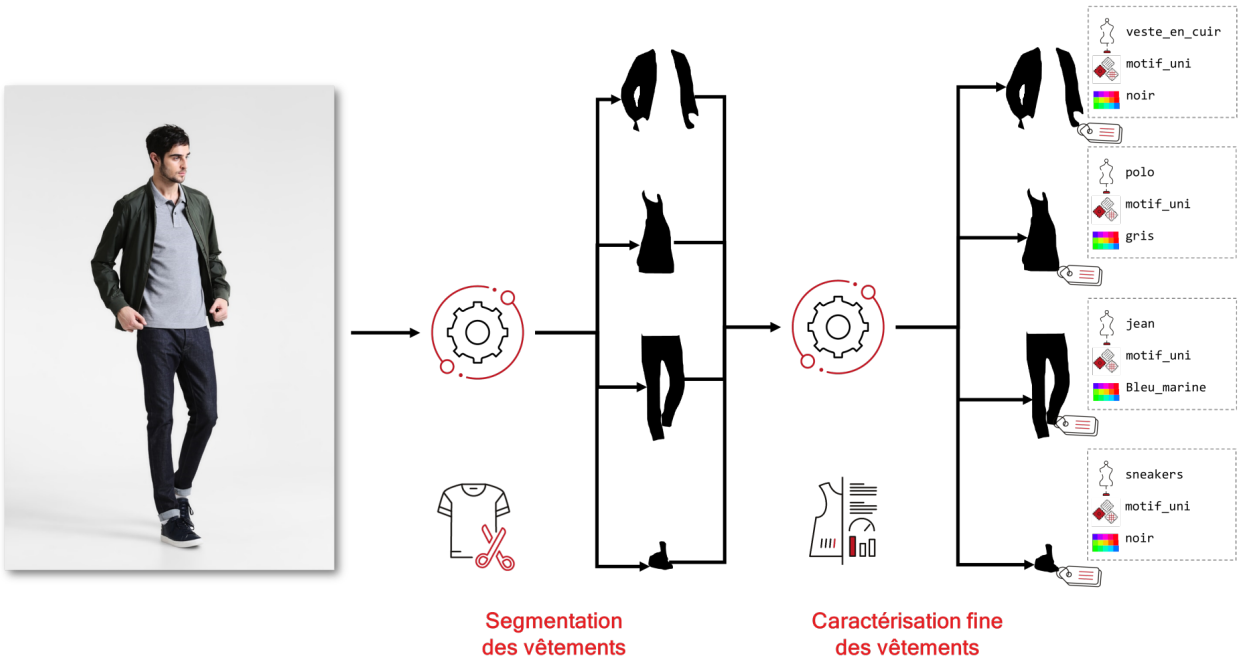


FIGURE 1.3 – Processus global de prédiction des mots clefs en deux étapes

Dans le chapitre 2 nous présentons nos travaux sur l'étape de localisation des vêtements dans des images. Nous présentons l'état de l'art des méthodes et les corpus de données permettant de réaliser cette localisation. Nous sélectionnons alors une approche par segmentation d'instances. Ces travaux ont été publiés dans une conférence nationale [JOUANNEAU et al., 2020]. Puis, Dans le chapitre 3 nous présentons nos travaux sur la validation de la performance des modèles. Nous soulevons des limites de l'évaluation globalement effectuée dans l'état de l'art. Nous proposons

alors un protocole d'évaluation des prédictions de localisation en trois axes : global, contour, contenu selon deux niveaux : localisation, corpus. Cette évaluation nous permet de sélectionner le modèle le plus performant. Ces travaux ont été publiés dans un *workshop* d'une conférence internationale [Jouanneau et al., 2021].

Dans le chapitre 4 nous exploitons les localisations obtenues pour caractériser les vêtements. Nous présentons alors la deuxième étape de notre processus. Nous appliquons la segmentation d'instances à notre contexte industriel, qui est d'associer des mots clefs décrivant des caractéristiques des vêtements. Nous nous focalisons en particulier sur trois caractéristiques : le type fin du vêtement, le motif et la couleur dominante du tissu. Nous décrivons alors nos données et les méthodes de classification et d'extraction de couleur développées. Ces travaux ont été reversés à Lectra.

Enfin, dans le chapitre 5 nous présentons une approche faiblement supervisée pour caractériser des éléments dans une image. Cette approche permet la prédiction de plusieurs étiquettes pour une donnée à partir d'un entraînement avec des exemples pour lesquels une seule étiquette est connue. Cette méthode est capable d'exploiter efficacement des données partiellement annotées, ce qui simplifie la collecte et le pré-traitement des données. Nous proposons alors une architecture orientée patch et une stratégie d'entraînement qui estime l'information manquante et plus particulièrement les exemples négatifs des étiquettes. Ces travaux ont été publiés dans une conférence internationale [Jouanneau et al., 2023].

Chapitre 2

Segmentation d'images de mode

Résumé : Nous proposons ici une étude des méthodes et corpus pour la localisation d'objets dans une image. Nous nous concentrons tout particulièrement sur la segmentation d'instances, qui est la forme la plus fine de localisation et distinction des objets dans l'image. Nous présentons alors l'application de méthodes récentes afin d'extraire tous les vêtements contenus dans des images. Ces travaux ont fait l'objet d'une publication dans une conférence nationale :

Jouanneau et al. 2020 : Warren Jouanneau, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2020). « Étude comparative de méthodologies issues de Mask R-CNN : Application au Corpus DeepFashion2 ». In : *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP'20)*, p. 1-3

2.1 Introduction

Dans ce chapitre, nous nous concentrons sur la segmentation d'instances et son utilisation pour le cas particulier des vêtements contenus dans des images. La segmentation d'instances est un problème de localisation d'éléments dans une image. Il s'agit d'une partie essentielle de notre proposition de méthodologie pour la caractérisation d'images de vêtements. La segmentation d'instances permet d'obtenir des masques qui contiennent les pixels d'un unique vêtement. Ceci permet de distinguer les potentielles différentes instances d'un même vêtement présent plusieurs fois dans l'image, ce qui est nécessaire pour la caractérisation exhaustive du contenu d'une image et une localisation fine. Une fois obtenus, ces masques peuvent de plus être donnés en entrée d'autres méthodes afin de caractériser les vêtements extraits. L'étape de segmentation permet donc de répondre à différents cas d'usages, entre autre l'indexation de contenu, cruciale pour l'industrie du textile et de la mode.

Nous commençons par présenter l'état de l'art en segmentation d'instances, en étudiant les évolutions historiques des méthodes de localisation dans la section 2.2, ainsi que les corpus de données spécifiques à la segmentation d'instances dans la section 2.3. Dans la section 2.4 nous présentons les conditions retenues pour évaluer les méthodes les plus récentes sur notre cas d'étude, *c.-à-d.* la localisation de vêtements. Nous présentons alors les résultats de premières expérimentations pour se familiariser avec la problématique dans la section 2.5. Ces expérimentations sont conduites dans l'objectif de faire ressortir les enjeux et potentiels problèmes qui peuvent advenir lors de l'industrialisation. Enfin, la section 2.6 propose une ouverture sur l'étape d'évaluation qui sera l'objet du chapitre suivant.

2.2 Méthodes de localisation d'objets

Dans cette section, nous examinons et listons les méthodes permettant de localiser et de différencier les objets contenus dans une image. Nous faisons transparaître l'historique des diverses évolutions proposées pour améliorer celles-ci. Ces méthodes peuvent être regroupées en trois grands domaines de traitement d'image. Chacun de ces domaines est constitué de problématique de localisation particulière et de tâches spécifiques à résoudre. Nous présentons d'abord la segmentation sémantique en section 2.2.1, puis en section 2.2.2 la détection d'objets, et enfin la segmentation d'instances dans la section 2.2.3. La figure 2.1 montre l'application de ces trois domaines. Elle fait figurer la classification et la segmentation panoptique [Kirillov et al., 2019] (sémantique plus instances) à titre de comparaison.

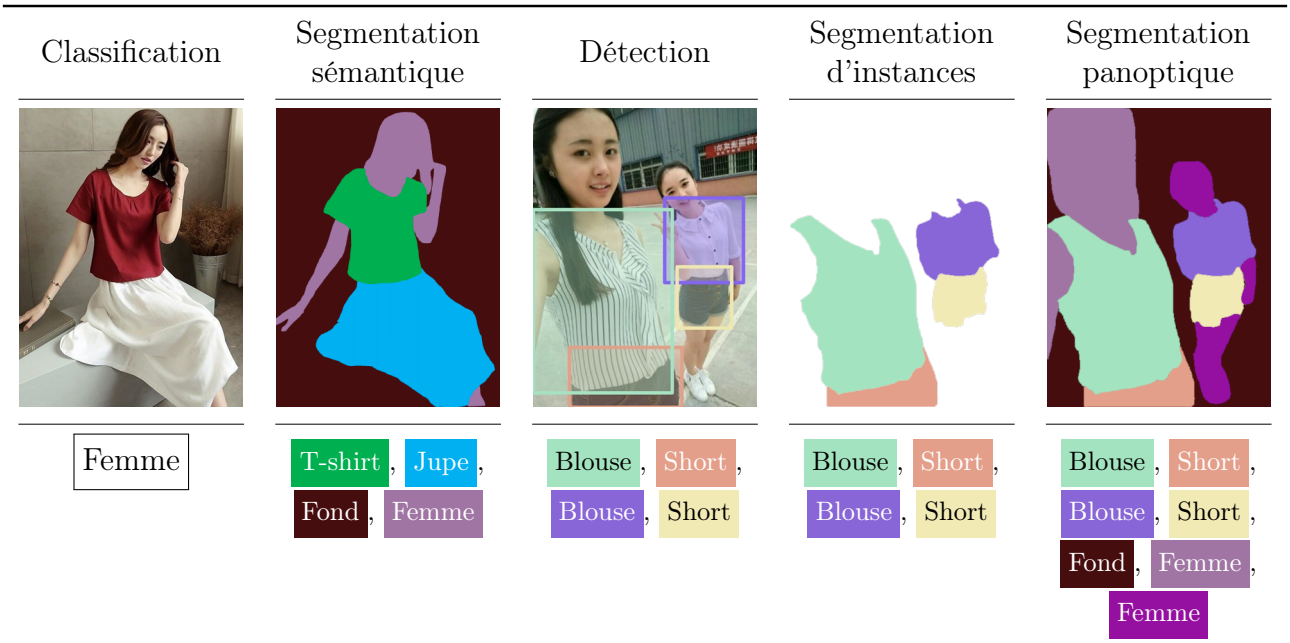


FIGURE 2.1 – Différents domaines de localisation appliqués à deux images exemples. La première contient un seul vêtement de chaque type, la deuxième, deux par type.

2.2.1 Segmentation sémantique

La segmentation sémantique consiste à attribuer une étiquette à chaque pixel d'une image, afin d'indiquer son appartenance à un groupe, appelée classe sémantique. La segmentation permet ainsi d'obtenir une partition de l'ensemble des pixels de l'image en différents sous-ensembles disjoints. En pratique, les pixels sont regroupés ensemble au sein d'une même classe si leur contenu dispose de caractéristiques similaires.

Les méthodes historiquement utilisées pour la segmentation peuvent être séparées en trois groupes : histogrammes, croissance de région ou optimisation d'une fonction objectif.

À partir d'histogrammes, différentes méthodes ont été proposées pour réaliser une segmentation. On peut mentionner les méthodes de seuillage (*ex.* binarisation OTSU en 1979), de quantification (*ex.* l'algorithme de coupure médian [HECKBERT, 1982]) ou encore de partitionnement (*ex.* k-moyennes [LLOYD, 1982] ou l'algorithme mean-shift [COMANICIU et MEER, 2002 ; FUKUNAGA et HOSTETLER, 1975]. Cependant, l'un des désavantages de ces approches respose sur le fait qu'elles ne considèrent pas suffisamment l'information géométrique contenue dans les images telle que les contours.

La seconde approche concerne les méthodes de croissance de région qui partent d'une graine initialisée par un utilisateur et font croître un contour jusqu'à ce qu'un contour saillant contenu dans l'image (un fort gradient ou une zone préalablement détectée par exemple par un filtre de CANNY en 1986) soit atteint. Des exemples de telles méthodes sont la segmentation par ligne de

partage des eaux issu du domaine de la morphologie mathématique, DBSCAN (***D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise*) [Ester et al., 1996 ; Ye et al., 2003] ou encore les méthodes de contours actifs [Kass et al., 1988]. Ces approches permettent à la segmentation de mieux respecter les contours de l'image, mais elles reposent sur une approche locale et sont souvent limitées à des objets à l'aspect uniforme.

Afin d'obtenir une segmentation globale, une dernière approche consiste à optimiser un modèle sur l'image entière. Une fonction objectif est ainsi définie comme la somme d'un terme mesurant l'homogénéité au sein des zones segmentées ainsi que la régularité spatiale des contours de ces zones et leur adéquation aux contours de l'image. Une telle fonction est alors classiquement minimisée par une approche d'optimisation variationnelle (*ex.* par ligne de niveau [Chan et Vese, 2001]) ou discrète (*ex.* coupe de graphe [Malik et al., 2001 ; Jianbo Shi et Malik, 2000]).

Ces méthodes peuvent s'appuyer sur différentes informations contenues dans des images, l'information la plus élémentaire étant la couleur des pixels. En utilisant la couleur, les méthodes ne tiennent cependant pas compte de la sémantique des groupes de pixels et de l'image, c'est-à-dire le sens du contenu. Afin d'obtenir une segmentation sémantique de l'image, il est nécessaire de s'appuyer sur des règles d'extraction de descripteurs visuels permettant d'interpréter le contenu. Les méthodes de segmentation basées sur l'apprentissage d'un classifieur de descripteurs visuels construits manuellement (*ex.* les SIFT (***S**cale-**I**nvariant **F**eature **T**ransform*) [Lowe, 1999], SURF (***S**peeded-**U**p **R**obust **F**eatures*) [Bay et al., 2006] ou les histogramme de gradients orientés (*HOG: **H**istrogram of **O**riented **G**radients*) [Dalal et Triggs, 2005]) ont connu un réel essor en vision par ordinateur jusqu'au début des années 2010.

Les travaux en apprentissage profond ont ensuite démontré la capacité des réseaux multicouches à automatiquement extraire les descripteurs nécessaires pour répondre à une tâche leur étant donnée. En se basant sur ce principe, Ciresan et al. en 2012 ont proposé d'utiliser un réseau de neurone profond. Ce réseau encode des patchs de l'image dans un vecteur de descripteurs, qui est ensuite fournit à un classifieur. Ceci permet de classifier les descripteurs du patch et attribuer une classe au pixel en position centrale dans le patch.

Afin d'obtenir une prédiction pour l'ensemble des pixels constituant une image avec cette approche, il est nécessaire de traiter tous les patchs séparément. Ces patchs se superposent et le même contenu est réutilisé de multiple fois. L'utilisation de réseau de neurones à couche de convolution (*CNN: **C**onvolutional **N**eural **N**etwork*) [Zeiler et Fergus, 2014] a ensuite permis d'obtenir efficacement la prédiction de tous les pixels simultanément. Dans ce contexte, Ronneberger et al. en 2015 propose l'architecture nommée U-Net constituée d'un encodeur et d'un décodeur. L'encodeur fonctionne de façon analogue à la méthode de Ciresan et al. en

2012. Le but est d'obtenir une représentation de l'image, *c.-à-d.* d'extraire automatiquement des descripteurs. Un décodeur reconstruit ensuite une image aux mêmes dimensions que celles d'entrée par sur-échantillonnage successif. De plus, pour chaque niveau de résolution, il y a une connexion entre l'encodeur et le décodeur, *c.-à-d.* un raccourci permettant de conserver l'information d'un niveau du décodeur et de la concaténer à la reconstruction des niveaux inférieurs faite par le décodeur (*skip connection, shortcut connection, residual connection*). Parmi les différents travaux proposés pour améliorer cette architecture, nous discernons notamment les réseaux U-Net++ [Zhou et al., 2018], U^2-Net [Qin, Z. Zhang et al., 2020] et DisNet [Qin, Dai et al., 2022].

Avec de telles méthodes, il est possible d'obtenir des masques de segmentation à partir d'une image ou sous-image. De plus, chaque pixel est associé à une unique classe. En segmentation sémantique, les différentes instances d'une classe se retrouvent néanmoins dans un même masque, sans aucune distinction. Ceci est problématique notamment lorsque les différentes instances d'une classe sont adjacentes.

2.2.2 Détection

Afin de différencier les instances d'une classe, les méthodes de détection visent à retrouver la localisation grossière des différents éléments d'une image. Cette localisation prend la forme d'une boîte englobante, *c.-à-d.* une sous-image, ne contenant qu'une seule instance. Les méthodes dédiées à cette tâche reposent sur une extraction de descripteurs non plus seulement pour définir le type *c.-à-d.* la classe du contenu, mais aussi pour retrouver et différencier les éléments de ce type.

Ainsi, la méthode SIFT proposée par Lowe en 1999 extrait des descripteurs locaux à partir de règles empiriques pré-établies. Ces descripteurs locaux sont ensuite comparés à ceux d'objets candidats conservés dans une base de données. Cette méthode a donc pour inconvénient d'être dépendante des images déjà décrites au préalable et de devoir procéder à une comparaison à celles-ci pour chaque nouvelle prédiction. Afin d'y pallier, Viola et Jones en 2001 passe par une classification de caractéristiques pseudo-Haar extraites à partir de multiples fenêtres. Les règles d'extraction de ces descripteurs sont préalablement définies et les fenêtres répondant positivement à la classification de toutes leurs caractéristiques sont alors considérées comme une détection. Viola et Jones en 2001 ont alors pu obtenir de très bons résultats sur la détection de visage. Une approche similaire, mais avec des descripteurs obtenus à partir d'HOG, a ensuite été proposée par Dalal et Triggs en 2005. Il a été montré que cette méthode améliore les résultats, surtout dans un contexte moins standardisé, avec des photographies personnelles. Toutefois, ce type de méthodologie repose sur la qualité et la pertinence des

descripteurs extraits.

Les CNN ayant montré leurs capacités à extraire des descripteurs pertinents en segmentation (et bien d'autres tâches), GIRSHICK et al. en 2014 suggèrent que ces descripteurs extraits peuvent aussi être pertinents pour la détection et proposent le réseau de neurones à couche de convolution à partir de région (*R-CNN: **R**egion based **C**onvolutional **N**eural **N**etwork*). R-CNN conserve l'approche par fenêtres, mais a recours à une extraction de descripteurs par le biais de CNN. La méthode s'articule en trois étapes : une proposition de régions (*c.-à-d.* fenêtres) obtenue par recherche sélective, une extraction de descripteurs par CNN puis une classification par machine à vecteurs de support (*SVM: **S**upport **V**ector **M**achine*). L'évolution « Fast » R-CNN [GIRSHICK, 2015] fusionne alors la classification et l'extraction de descripteur au sein du même CNN. De plus, ce réseau traite l'image dans sa globalité pour obtenir une carte de descripteurs. Les régions d'intérêt (*ROI: **R**egion **O**f **I**nterest*) obtenues par recherche sélective sont alors projetées sur cette carte et les boîtes englobantes sont affinées par régression en parallèle de la classification. Cette proposition améliore grandement le temps d'entraînement et d'inférence des modèles, mais l'algorithme de recherche sélective de ROI constitue toujours une part importante de la charge de calcul. Afin d'y remédier, « Faster » R-CNN [REN et al., 2015] réalise la recherche de ROI par un réseau de proposition de région (*RPN: **R**egion **P**roposal **N**etwork*) et résout ainsi le problème de la détection avec une architecture CNN de bout en bout. « Faster » R-CNN a grandement amélioré la performance, en temps et en précision.

De manière générale, les méthodes assimilées R-CNN peuvent être regroupées et présentées comme des méthodologies en deux étapes : une première étape de proposition de ROI, et une seconde, de prédiction et d'affinage. L'étape de proposition de ROI, malgré l'utilisation de RPN, reste la principale raison du temps d'exécution. En effet, pour produire de bonnes régions candidates pour la détection, le choix des régions doit être exhaustif et inclure différentes tailles et des fenêtres pouvant se superposer. Les méthodologies dites « en une passe » (*single shot*) essaient de s'abstraire de cette proposition de régions, ou du moins de la simplifier. Les méthodes SSD (***S**ingle **S**hot multibox **D**etector*) [W. LIU et al., 2016] et YOLO (***Y**ou **O**nly **L**ook **O**nce*) [REDMON et al., 2016] apposent une grille figée sur l'image. Le même ensemble fixe de boîtes d'ancrage (*anchor boxes*) est associé à chacune des cellules de cette grille. C'est à partir de l'ensemble des boîtes d'ancrage de toutes les cellules que la confiance du réseau sur la présence d'un objet et la boîte englobante associée sont prédites. SSD met l'accent sur la précision au détriment de la vitesse d'exécution, bien qu'étant quand même nettement plus rapide que les méthodes en deux étapes (*c.-à-d.* R-CNN). La méthode SSD repose alors sur plusieurs grilles à différents niveaux du champ réceptif, pouvant ainsi détecter des objets de différentes tailles. À l'inverse, YOLO mise sur le temps d'exécution en ne s'appuyant que sur une seule grille et

en éliminant les prédictions superflues par NMS (***N**on-**M**axima **S**uppression*). Il existe de nombreuses versions de YOLO correspondant à des propositions d'optimisation et d'améliorations. L'une des dernières en date lors de la rédaction de ce document est YOLOv7 [C.-Y. Wang et al., 2022].

Que ce soit en suivant une méthodologie en deux étapes ou en une passe, ces méthodes permettent de répondre au problème de la détection d'objets dans une image. Plus généralement, la détection permet de retrouver et de différencier les instances d'un type d'élément, néanmoins les localisations obtenues sont grossières. Les boîtes englobantes peuvent être vues comme des sous-images contenant au moins un objet. Ces sous-images contiennent des pixels qui n'appartiennent pas à l'objet détecté et qui peuvent même appartenir à un autre objet. De plus, selon la morphologie de l'objet, la proportion de pixels appartenant réellement à l'objet est plus ou moins grande. Pour tous les problèmes dans lesquels la localisation doit être précise et les sous-images exploitées, ces méthodes ne sont donc pas les plus adaptées.

2.2.3 Segmentation d'instances

La segmentation d'instances peut être définie comme la localisation fine des objets, *c.-à-d.* retrouver tous les pixels qui appartiennent à une instance d'éléments dans une image. Elle diffère ainsi de la segmentation sémantique où le problème posé est d'attribuer une classe à chaque pixel. Le but ici, n'est pas d'être exhaustif en termes de pixel, mais bel et bien en termes d'objet. Cette problématique est donc à la jonction de deux autres problèmes de traitement d'image : comment segmenter une image, puis différencier les zones (*c.-à-d.* les masques) qui appartiennent à des instances différentes. Certaines méthodes proposent de prendre le problème dans l'autre sens : réaliser tout d'abord une détection, des instances, puis obtenir les masques par segmentation des boîtes englobantes détectées.

Ainsi, « Mask »R-CNN [He, Gkioxari et al., 2017] s'appuie sur les performances de la méthode de classification multi-objets (*c.-à-d.* détection) « Faster » R-CNN [Ren et al., 2015] en ajoutant la prédiction du masque de l'instance en parallèle de la prédiction de la boîte englobante et de la classe de l'instance. À partir de cette approche, plusieurs types d'amélioration ont été proposées.

Dans « Mask » R-CNN, le score du masque est identique à celui de la classification, cependant il y a rarement une corrélation entre celui-ci et la qualité du masque. MS R-CNN (***M**ask **S**coring R-CNN*) [Z. Huang et al., 2019] propose une correction de ce score en ajoutant un bloc prédisant l'indice de Jaccard (*c.-à-d.* intersection sur l'union entre les masques prédits et la vérité terrain) par régression. Cela permet au réseau d'avoir connaissance de la qualité de sa propre prédiction du masque. Dans « Mask » R-CNN, l'indice de Jaccard sur les boîtes englo-

bantes ou les masques permet de distinguer les prédictions positives des négatives. Un seuil trop élevé implique la disparition de cas positifs lors de l'entraînement et un sur-apprentissage. Pour y remédier, « Cascade » R-CNN [Cai et Vasconcelos, 2018] propose de chaîner plusieurs détecteurs associés à des niveaux croissants de seuils. Les prédictions des boîtes englobantes d'un détecteur sont alors fournies en entrée au suivant. « Cascade » R-CNN peut être couplé avec« Mask » R-CNN, en adoptant le même chaînage sur les branches de prédiction des masques. HTC (***H**ybrid **T**ask **C**ascade*) [K. Chen, Pang et al., 2019] propose d'entremêler les chaînes de détection et de prédiction des masques au lieu de les chaîner séparément, et aussi d'ajouter une branche de contexte à l'architecture.

Ces méthodes reposant sur « Mask » R-CNN, et par extension sur la méthodologie en deux étapes, privilégient la qualité de la prédiction au détriment de la vitesse d'exécution. Inspiré par la proposition de « Mask » R-CNN de partir de la détection de« Faster » R-CNN, une démarche analogue a été entreprise avec la méthodologie en une passe et les méthodes SSD et YOLO. Ainsi, YOLACT [Bolya et al., 2019] repart de la détection effectuée par YOLO, mais, ne se contente pas de segmenter les boîtes englobantes. Des masques prototypes sont générés à partir de l'image dans son entièreté en utilisant la carte de descripteur. Puis pour chaque instance, des coefficients sont prédits pour fusionner ces prototypes et obtenir les masques. De nombreuses autres méthodes de segmentation d'instances proposent également de repartir des architectures dites en « une passe », par exemple Blendmask [H. Chen et al., 2020], Centermask [Lee et Park, 2020] ou SOLO [X. Wang et al., 2020].

2.3 Corpus de données pour la localisation

Dans cette section, nous examinons les corpus disponibles afin d'obtenir des modèles répondant à notre besoin de localiser des éléments dans une image. En apprentissage automatique, ces données sont nécessaires non seulement pour entraîner des modèles, mais aussi pour valider et sélectionner les méthodes. La performance et la capacité des méthodes à correctement répondre à une tâche étant liées à la quantité et à la variabilité des données, la taille et la diversité sont des facteurs cruciaux dans la sélection d'un corpus. Dans la section 2.3.1, nous listons les principaux corpus disponibles pour la localisation et la segmentation d'instances. Puis, dans la section 2.3.2 nous présentons les corpus spécifiques à notre cas d'étude concernant les images contenant des vêtements.

2.3.1 Corpus pour la détection et la segmentation

De nombreux corpus d'images existent, le plus populaire étant ImageNet [Deng et al., 2009]. Certains de ces corpus incorporent des informations de localisation dans les images. Bien entendu, une proportion plus faible de ceux-ci différencie les éléments ou objets, *c.-à-d.* qu'ils contiennent des images annotées au niveau des instances. Cela s'explique par l'exhaustivité requise et la complexité d'annoter le type et la position de tous les éléments contenus dans une image. Cette complexité est d'autant plus importante pour la tâche d'annotation de tous les pixels d'un objet qui est nécessaire pour la segmentation d'instances. De tel corpus sont souvent spécifiques à un contexte particulier. Ainsi, pour des problématiques de véhicule autonome, les corpus Cityscapes [Cordts et al., 2016] et BDD100K [Yu et al., 2020] contiennent des images dans un contexte routier et les éléments y sont annotés en fonction (*ex.* éléments de type : voiture, piéton, *etc.*). Un autre contexte est par exemple la détection par vue aérienne ou satellite, le corpus DOTA [Xia et al., 2018] contient ainsi des images vues du ciel et des annotations sur les bâtiments, les lieux et des gros véhicules.

Certains corpus sont cependant moins spécifiques et rassemblent des photographies personnelles. Ces images sont capturées dans différents contextes : sportif, repas, tourisme, *etc.* et dans différentes conditions : de prise de vue, d'illumination, d'environnements. Il est aussi possible d'avoir une grande diversité d'objets présents dans une image. Cette diversité d'objets, contextes et conditions complexifie le problème de localisation et les méthodes doivent s'y adapter. Ce sont les raisons pour lesquelles de tels corpus ont été adoptés par la communauté et font référence pour tester et évaluer les modèles. Il est maintenant d'usage de mener des expérimentations en utilisant ces corpus afin de permettre à la communauté de rapidement évaluer l'amélioration et l'apport d'une nouvelle proposition. VOC (*Pascal **V**isual **O**bject **C**lasses*) [Everingham et al., 2010] a rempli ce rôle pour la détection et la segmentation d'instances, mais actuellement COCO (*Microsoft **C**ommon **O**bject in **CO**ntext*) [T.-Y. Lin, Maire et al., 2014] tend à le supplanter ou le complémenter *a minima*.

VOC (*Pascal Visual Object Classes*)

VOC [Everingham et al., 2010] a subi différentes évolutions et améliorations. La version originale de 2005 contient 1578 images pour la détection de quatre types d'objets : personne, voiture, moto, vélo. La version la plus récente et la plus utilisée est celle de 2012 (VOC2012). Cette version contient 5 717 images pour l'entraînement des modèles et 5 823 pour la validation. Les images sont en couleur et peuvent atteindre une résolution de 640×640 pixels. VOC2012 contient les annotations d'instances des objets présents dans les images. Ces annotations sont le type, la boite englobante et le masque des objets. Les objets peuvent être de 20 classes

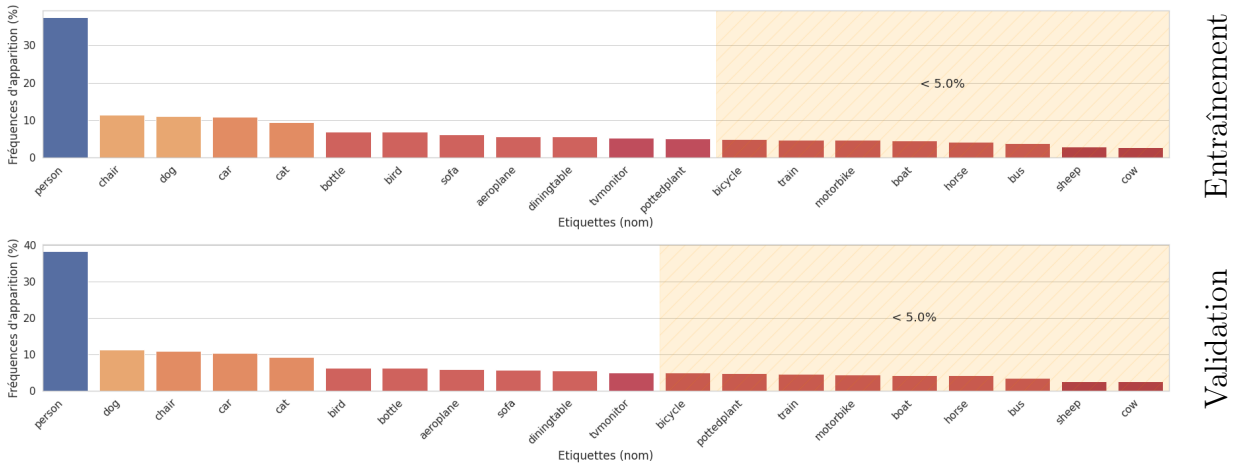


FIGURE 2.2 – Fréquence des étiquettes en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de VOC2012 *(La zone hachurée en orange contient les étiquettes qui ont une fréquence inférieure à 5%)*

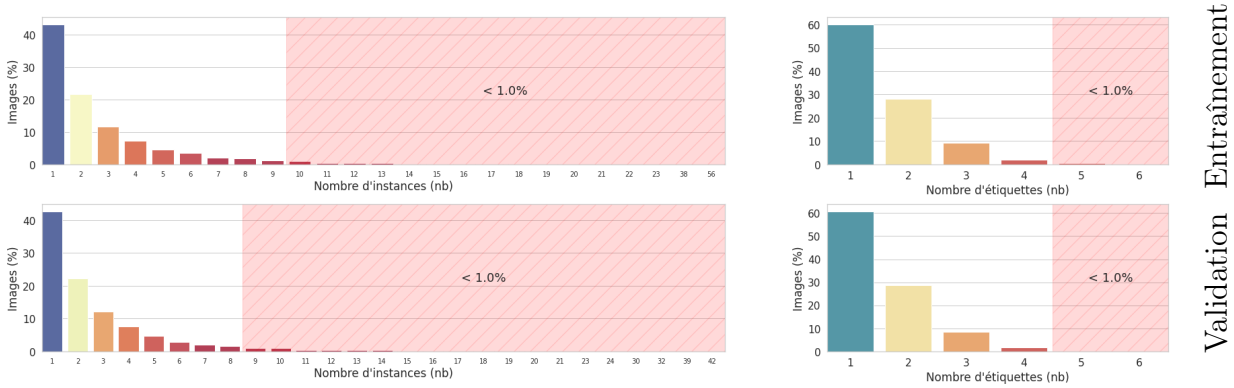


FIGURE 2.3 – Fréquence du nombre d'instances (objets, à gauche) et d'étiquettes (à droite) en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de VOC2012 *(La zone hachurée en rouge contient les nombres étiquettes-instances qui ont une fréquence inférieure à 1%)*

différentes, plus ou moins représentées au sein des images (*fig.* 2.2). Chaque image contient en moyenne 1, 38 types d'objet différents (*fig.* 2.3). VOC2012 est facilement accessible [1] ce qui en facilite l'utilisation.

COCO (*Microsoft Common Object in COntext*)

COCO [T.-Y. LIN, MAIRE et al., 2014] est un corpus plus récent qui propose plus d'images et plus d'annotations (types et instances) que VOC. C'est pourquoi il a rapidement été adopté par la communauté. La version originale de 2014 de COCO (COCO-14) contient 82 783 images pour l'entraînement des modèles et 40 504 images pour la validation. La version la plus ré-

1. host.robots.ox.ac.uk/pascal/VOC/voc2012

cente de 2017 (COCO-17) contient les mêmes images et annotations, mais réorganisées avec 118 287 images pour l'entraînement et 5 000 images pour l'évaluation. Les images en couleurs

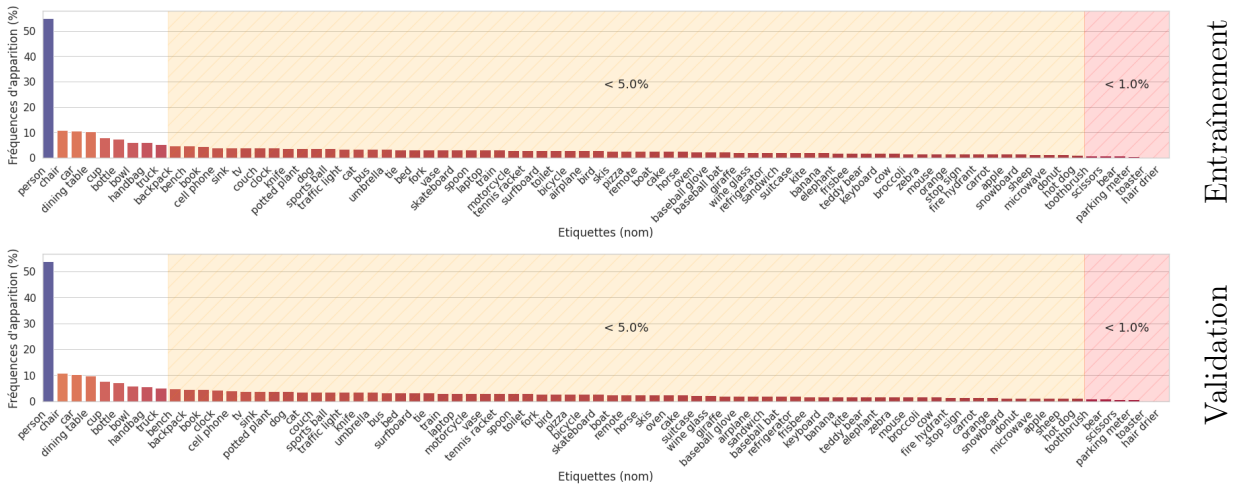


FIGURE 2.4 – Fréquence des étiquettes en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de COCO-14 *(La zone hachurée en orange contient les étiquettes qui ont une fréquence inférieure à* 5% *et inférieure à* 1% *dans la zone rouge)*

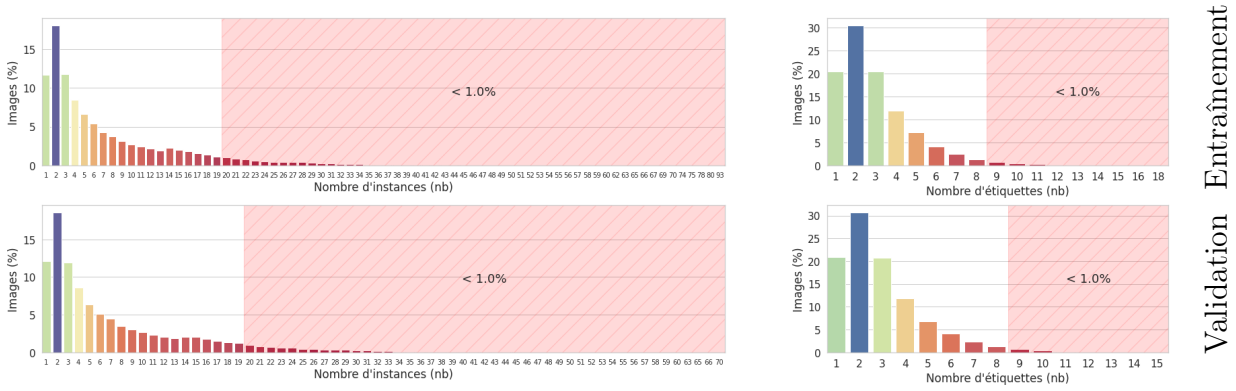


FIGURE 2.5 – Fréquence du nombre d'instances (objets, à gauche) et d'étiquettes (à droite) en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de COCO-14 *(La zone hachurée en rouge contient les nombres étiquettes-instances qui ont une fréquence inférieure à* 1%*)*

ont une résolution pouvant aller jusqu'à 500×500 pixels et sont annotées avec 80 étiquettes différentes plus ou moins présentes dans les images (*fig.* 2.4). Ce corpus fournit lui aussi le type, la boîte englobante et le masque des objets. Chaque image contient en moyenne $2,92$ types d'objet différents (*fig.* 2.5). COCO est facilement accessible [2], et une interface de programmation d'application (*API: **A**pplication **P**rogramming **I**nterface*) [3] optimisée est disponible afin de

2. cocodataset.org/#download
3. github.com/cocodataset/cocoapi

visualiser et d'évaluer les résultats. COCO utilise le format RLE (***R**un **L**ength **E**ncodding*) pour les masques et l'API met à disposition les fonctions pour les compresser et les décompresser.

2.3.2 Corpus spécifiques aux vêtements

Parmi les corpus d'images de vêtements disponibles, peu incorporent les masques nécessaires à la segmentation d'instances. À notre connaissance, il existe trois grands corpus disponibles fournissant des annotations sur la localisation de vêtements dans des images.

Le corpus Paper Doll [Yamaguchi et al., 2013] contient [4] 1 097 474 images de vêtements. Cependant, ces vêtements y sont faiblement annotés. À partir de 52 255 de ces images, ModaNet [Zheng et al., 2018] propose [5] alors une annotation plus fine des vêtements. Les annotations enrichies sont constituées de boîtes englobantes et de masques de treize types de vêtements et accessoire. Le corpus ModaNet est donc plus adapté à l'apprentissage de modèle de détection et de segmentation d'instances. Certains problèmes de qualité et des erreurs ont cependant pu être observés [6] sur ces annotations.

Fashionpedia [Jia et al., 2020] met à disposition [7] 48 823 images annotées. Ces annotations contiennent les masques et types de vêtements. Par ailleurs, ce corpus propose une ontologie des différentes sous-parties qui constitue un vêtement et les localise dans l'image (*ex.* le col, les manches, les poches d'une chemise). Il propose donc des graphes de connaissance et une localisation fine de 46 type de vêtements et de 294 attributs.

Le corpus Deepfashion2 [Ge et al., 2019] est actuellement celui qui propose le plus grand nombre d'images et d'annotations avec masques. Ce corpus est une amélioration de Deepfashion [Ziwei Liu et al., 2016]. Les données sont séparées en deux, 312 186 instances dans 191 961 images pour l'entraînement, 52 490 instances dans 32 153 images pour l'évaluation. Chacune des instances sont annotées suivant 13 catégories (*fig.* 2.6 et *fig.* 2.7). Les images proviennent de particuliers ou de professionnels et comprennent des vêtements dans de nombreux contextes (*ex.* en extérieur, studio). Il y a une grande variabilité des prises de vues et des conditions de capture. On y retrouve trois niveaux de visibilité du vêtement, trois niveaux de zoom, si le vêtement est porté ou non et enfin la position (vue de face ou de côté). Les masques sont représentés sous la forme de contours discrétisés.

4. github.com/kyamagu/paperdoll
5. github.com/eBay/modanet
6. github.com/eBay/modanet/issues/4
7. fashionpedia.github.io/home/Fashionpedia_download.html

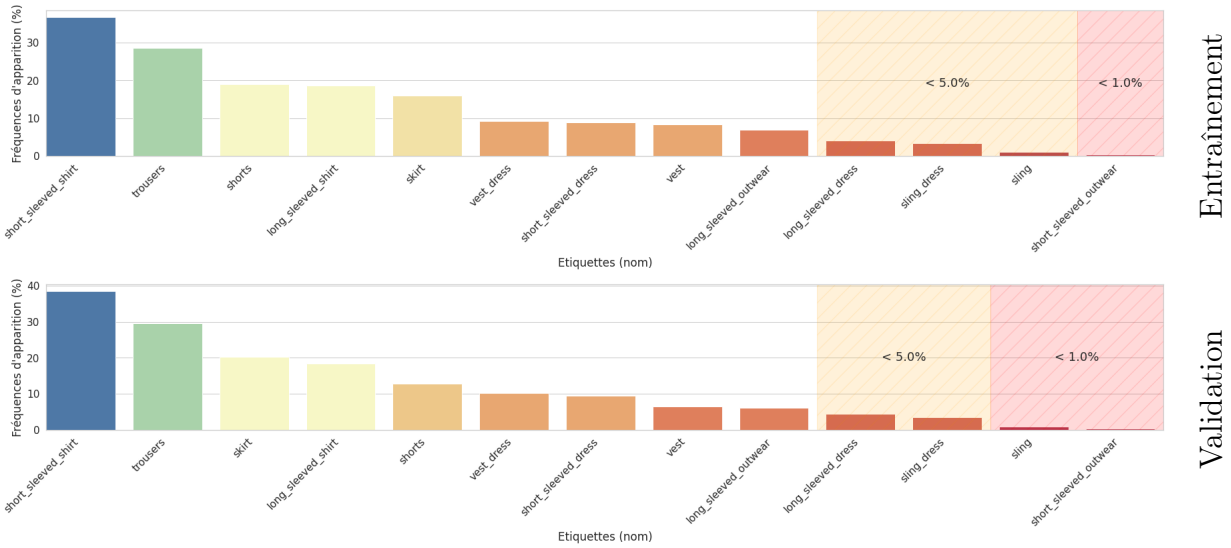


FIGURE 2.6 – Fréquence des étiquettes en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de Deepfashion2 *(La zone hachurée en orange contient les étiquettes qui ont une fréquence inférieure à* 5% *et inférieure à* 1% *dans la zone rouge)*

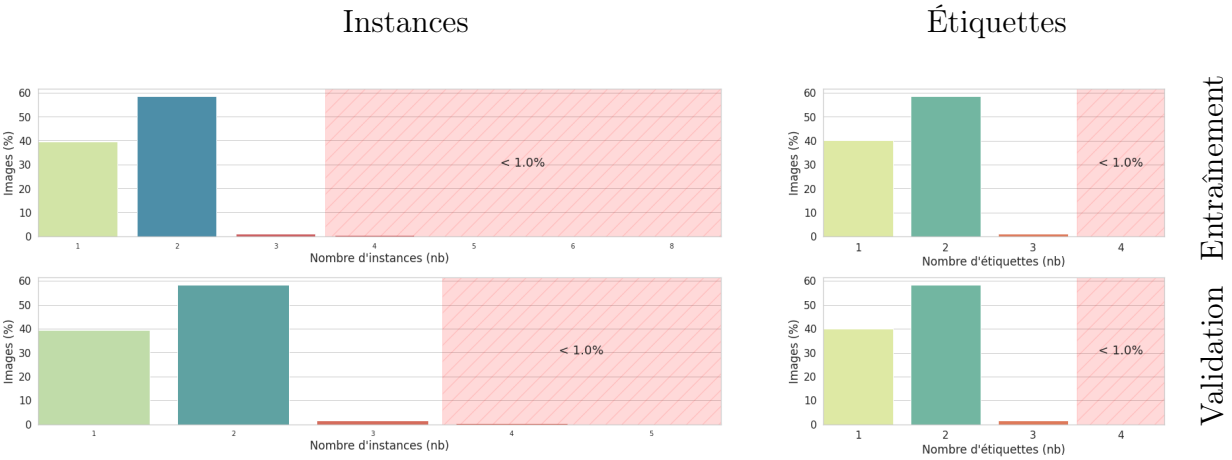


FIGURE 2.7 – Fréquence du nombre d'instances (objets, à gauche) et d'étiquettes (à droite) en pourcentage d'images dans les jeux d'entraînement (en haut) et validation (en bas) de Deepfashion2 *(La zone hachurée en rouge contient les nombres étiquettes-instances qui ont une fréquence inférieure à* 1%*)*

2.4 Conditions expérimentales

Dans cette partie, nous détaillons la mise en place et les conditions de nos premières expérimentations sur le problème de segmentation d'instances de vêtements. Dans la section 2.4.1 nous présentons la bibliothèque réalisée pour lancer des expérimentations reproductibles et conserver des informations sur leurs déroulement. Ensuite, nous justifions le choix du corpus dans la section 2.4.2 et des méthodes dans la section 2.4.3. Enfin, dans la section 2.4.4, nous décrivons la stratégie d'entraînement des modèles retenus.

2.4.1 Développement d'une bibliothèque d'expérimentations

Afin de faciliter, reproduire, suivre les expérimentations, une bibliothèque python a été développée. Proposant une API et des méthodes pour le contrôle d'expérience, son intérêt principal est de pouvoir lancer une expérience à partir d'une configuration donnée. Une expérience peut reposer sur plusieurs sous-exécutions et les conditions pour les exécuter.

Ainsi, chaque sous-exécution dispose de sa propre configuration. Elle est définie par : - des fichiers sources, - des paramètres d'exécution, - un point d'entrée de l'exécution (*c.-à-d.* comment lancer l'exécutable), - une cible (*c.-à-d.* où l'exécution doit être lancée), - un environnement, - une ou plusieurs méthodes de suivi, - un corpus.

Chacun de ces éléments est mis à disposition par l'expérience. La configuration d'expérience contient alors au minimum les configurations des éléments suivants : - une exécution, - une cible d'exécution, - un environnement nécessaire aux exécutions, une méthode de suivi des exécutions, - un corpus de données. À partir de chacune de ces sous-configurations, la bibliothèque va construire le nécessaire pour lancer l'expérimentation, *c.-à-d.* l'ensemble des sous-éxecutions et leurs combinaison d'éléments mis à disposition par l'expérience.

Pour les besoins des expériences, le code nécessaire à l'interprétation de différentes configurations des éléments a dû être réalisé. Ainsi, les exécutions peuvent être lancées localement ou déployées sur la plate-forme applicative en nuage de Microsoft Azure. Les environnements sont définis à partir de Docker, pip, et Conda. Le suivi des expériences peut être local, sur AzureMl (*Microsoft* ***Azure*** ***M****achine* ***L****earning service*) ou sur MLflow. Enfin, les exécutions peuvent accéder à des données locales ou être déployées sur le stockage blob d'Azure.

2.4.2 Corpus

Le corpus retenu pour les premières expérimentations est Deepfashion2 [Ge et al., 2019]. Ce corpus étant actuellement celui qui contient le plus d'images aux contextes variés avec des annotations de qualité, il répond donc le mieux à des enjeux industriels. Il permet de rapidement appréhender les problématiques liées à la segmentation d'instances et permet de soulever les problèmes potentiels liés à la prédiction de masques de vêtements.

Nous l'avons retravaillé et standardisé pour mieux correspondre au format d'annotation défini par COCO [T.-Y. Lin, Maire et al., 2014] afin d'utiliser l'API de COCO et des architectures de CNN disponibles dans différentes bibliothèques. Ce corpus a par la suite été déployé sur le stockage blob d'Azure afin de faciliter son utilisation et pouvoir s'appuyer sur les ressources de calcul d'Azure.

2.4.3 Méthodes évaluées

Lors de la constitution du corpus Deepfashion2, Ge et al. en 2019 ont expérimenté et évalué la segmentation d'instances à partir de « Mask » R-CNN [He, Gkioxari et al., 2017]. Les méthodes dites en deux étapes (section 2.2.2) étant décrites comme celles fournissant les prédictions les plus précises dans la littérature, nous proposons d'évaluer différentes méthodes issues de « Mask » R-CNN (section 2.2.3). Les méthodes retenues sont : MS R-CNN [Z. Huang et al., 2019], « Cascade » R-CNN [Cai et Vasconcelos, 2018], HTC [K. Chen, Pang et al., 2019]. Afin de leur garantir des conditions d'entraînement équivalentes, nous utilisons les implémentations de ces architectures mises à disposition [8] par la bibliothèque MMDetection [K. Chen, J. Wang et al., 2019] et le même sous-réseau extracteur de descripteur. Ce réseau est un ResNet-50 [He, X. Zhang et al., 2016] utilisé en FPN (***F**eature **P**yramid **N**etwork*) [T.-Y. Lin, Dollár et al., 2017]. Une surcouche à MMDetection a dû être développée dans notre bibliothèque pour pouvoir charger et configurer des architectures et pouvoir lancer des entraînements et des évaluations.

2.4.4 Hyper-paramètres et stratégie d'entraînement

L'entraînement des modèles a été déployé sur Azure. La machine virtuelle utilisée disposait d'une carte graphique NVIDIA Tesla P100. L'entraînement des architectures étant long, consommateur d'énergie et coûteux, les résultats présentés dans la partie suivante ont été obtenus après une seule epoch. Au vu de la taille de la base de données, nous soutenons que cela est suffisant pour observer de premières tendances et voir les points problématiques à adresser. L'optimiseur utilisé est SGD (***S**tochastic **G**radient **D**escent*) avec un taux d'apprentissage (*learning rate*) $\mathrm{lr} = 0.02$ un moment de 0.9 et une dégradation des pondérations de 10^{-4}.

2.5 Résultats

La figure 2.8 montre la qualité des masques inférés à partir des différents modèles. Les masques prédits par les modèles de « Mask » R-CNN et de « Cascade » R-CNN semblent subjectivement moins précis. On peut ainsi noter des imperfections ou des absences de détection sur la jupe et le haut en colonne (2), le pantalon en colonne (3), le col en colonne (4), la jupe et le haut en colonne (5). La robe en colonne (6) semble poser problème pour l'ensemble des modèles, cela pouvant être dû à l'orientation du sujet peu commun dans le corpus.
La table 2.1 présente différentes métriques observées lors de l'entraînement et de l'évaluation.

8. https://github.com/open-mmlab/mmdetection

FIGURE 2.8 – Images où figurent les masques réels (a) et prédits avec un score supérieur à 0.4. Vêtement(s) sur cintre (1), à plat (2), avec occlusion due aux cheveux (3), de face (4), avec occlusion due aux bras (5), de dos (6), plusieurs instances (7), images d'exemples de la figure 2.1 (7-8)

La moyenne des précisions moyennes (*mAP : **m**ean **A**verage **P**recision*) correspond à la moyenne des AP (***A**verage **P**recision*) par classe. L' AP est l'aire sous la courbe de précision-rappel. Afin, d'obtenir cette courbe, un seuil sur l'intersection sur l'union (*IoU: **I**ntersection **O**ver **U**nion*) entre masque prédits et masques de vérité terrain permet de discriminer les vrais positifs (TP) des faux positifs (FP). Plus cette métrique est proche de 1.0 meilleur est le modèle. Nous détaillons plus précisément cette métrique dans le chapitre suivant (chapitre 3). La mAP50 et la mAP75 dans le tableau 2.1 correspondent alors aux IoU seuillés spécifiquement à 0.5 et 0.75. La mAP correspond à la moyenne des AP pour des IoU allant de 0.5 à 0.95 par incrément de 0.05.

On peut noter ainsi que MS R-CNN et HTC ont des mAP et mAP75 proches, avec des scores supérieurs en mAP50 pour la première méthode. Cela traduit donc une meilleure mAP aux indices supérieurs à 0.75 pour Hybrid task Cascade. La table 2.2 présente le détail des mAP par classes. Les meilleurs mAP se répartissent de nouveau entre les méthodologies MS R-CNN et HTC, sauf deux exceptions. Cependant, pour certaines classes ces résultats sont à prendre avec précaution. Les mAP peuvent en effet être proches entre deux méthodologies (*e.g.* label shorts), ou présenter de faibles valeurs, dues à une possible ambiguïté entre labels (*e.g.* label sling dress) ou une disparité et un faible nombre d'instances pour certaines classes (*fig.* 2.7 labels short sleeved outwear et sling). Le temps nécessaire pour l'inférence varie du simple au double entre les modèles reposant sur « Mask » R-CNN et HTC. La variation relative en temps n'est pas aussi élevée pour l'entraînement.

Les architectures suivant la méthodologie de MS R-CNN et HTC se distinguent. En effet, ces méthodologies ont les meilleures mAP en évaluation. Toutefois, HTC prédit des masques de meilleures qualités (*c.-à-d.* mAP aux IoU supérieur à 0.75). Cependant, en continuant l'entraînement, si les résultats entre ces approches restent proches, MS R-CNN aura un avantage en termes de complexité.

TABLE 2.1 – Résultats des évaluations

architectures	mAP	mAP50	mAP75	temps d'inférence	temps d'entraînement
Mask R-CNN	0.244	0.361	0.277	0.11 s/image	10.94 h/epoch, 0.41 s/batch
MS R-CNN	0.256	0.382	0.294	0.11 s/image	11.20 h/epoch, 0.42 s/batch
Cascade R-CNN	0.243	0.362	0.276	0.13 s/image	14.68 h/epoch, 0.55 s/batch
HTC	0.260	0.375	0.295	0.22 s/image	16.87 h/epoch, 0.63 s/batch

TABLE 2.2 – AP par classes (*s.s. : short sleeved, l.s. : long sleeved*)

	shirt		outwear		vest	sling	shorts	trousers	skirt
	s.s.	l.s.	s.s.	l.s.					
Mask R-CNN	0.558	0.339	0.0	0.174	0.175	**0.019**	**0.378**	0.395	0.330
MS R-CNN	0.516	0.288	0.003	**0.188**	**0.282**	0.012	0.355	0.402	**0.398**
Cascade R-CNN	0.521	0.288	0.0	0.183	0.248	0.011	0.344	0.404	0.346
HTC	**0.584**	**0.341**	**0.007**	0.157	0.230	0.017	0.373	**0.418**	0.318

	dress			
	s.s.	l.s.	vest	sling
Mask R-CNN	0.281	0.158	0.311	0.053
MS R-CNN	0.269	0.153	0.346	**0.110**
Cascade R-CNN	0.257	0.148	0.333	0.071
HTC	**0.317**	**0.171**	**0.348**	0.094

2.6 Conclusion

Dans ce chapitre, nous avons présenté différentes méthodes et corpus pour la segmentation d'instances. Nous avons sélectionné quatre méthodes que nous avons appliquées au corpus Deepfashion2. Ces travaux ont été publiés [Jouanneau et al., 2020] et les premiers résultats ont montré que la méthode HTC a la meilleure mAP au détriment de la complexité en temps.

Le corpus Deepfashion2 présente différents avantages pour la segmentation d'instances de vêtements. Néanmoins, quelques limites apparaissent : les sujets sont principalement féminins, les masques contiennent des pixels qui n'appartiennent pas aux vêtements (*e.g.* main, cheveux), la taxonomie choisie pour les labels pose question, et certaines classes sont faiblement représentées. Il pourrait ainsi être utile de se tourner vers d'autres corpus d'images de vêtements ou d'en constituer un nouveau présentant des segmentations plus précises et une granularité plus fine et hiérarchique (*e.g.* segmentation au niveau des manches, des cols, des jambes appartenant aux vêtements).

Le corpus Fashionpedia pourrait permettre d'obtenir plus d'informations sur la composition des vêtements avant d'être caractérisés plus finement. Cependant, on observe que certains types de vêtements dans Deepfashion2 sont plus difficiles à détecter et à segmenter (en particulier ceux plus faiblement présents dans le corpus). Fashionpedia propose une très grande diversité de types, et nombre de ces types sont faiblement présents dans le corpus. Il est donc nécessaire d'adapter les méthodes à cette problématique d'étiquettes aux fréquences faibles. Le corpus Modanet propose la localisation d'accessoires. Ces accessoires posent d'autres difficultés, entre autre liées à la taille dans l'image. Ce corpus pourrait compléter l'apprentissage avec Deepfashion2 après correction des annotations [9]. Cependant, notre cas d'étude se concentre sur les vêtements.

Les premiers résultats présentés dans ce chapitre sont néanmoins encourageants dans une optique d'exploitation industrielle. Les méthodes de segmentation d'instances étudiées semblent en mesure de fournir des extractions de vêtements suffisamment précises pour permettre leur caractérisation ultérieure. Se concentrer sur ces méthodes et potentiellement expérimenter d'autres sous-réseaux extracteur de descripteurs est donc une piste à suivre.

Cependant, ces premières expérimentations montrent aussi qu'il est difficile d'interpréter les résultats. Une évaluation visuelle et subjective est impossible dû au volume des données. L'évaluation automatique de la segmentation d'instances avec la mAP couplée à l'IoU ne permet pas de distinguer les cas d'erreur et de contrôler le périmètre d'utilisation. Définir un protocole et des métriques d'évaluation est essentiel afin de correctement sélectionner les méthodes. Nous abordons cette problématique dans le chapitre suivant.

9. github.com/eBay/modanet/issues/4

Chapitre 3

Évaluation des méthodes de segmentation

Résumé : Nous proposons ici une étude des méthodes et métriques d'évaluations de modèles de segmentation d'instances. Nous proposons un protocole d'évaluation en trois axes : global, contour, contenu et sur deux niveaux : masque, corpus. Pour chacun de ces axes et niveaux, nous sélectionnons ou proposons une métrique. Enfin nous appliquons ce protocole à l'évaluation de cinq méthodes sur le corpus Deepfashion2. Ces travaux ont fait l'objet d'une publication dans un workshop d'une conférence internationale [Jouanneau et al., 2021].

Jouanneau et al. 2021 : Warren Jouanneau, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2021). « Where are my clothes ? A multi-level approach for evaluating deep instance segmentation architectures on fashion images ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3951-3955

3.1 Introduction

La segmentation des vêtements est une étape cruciale de nombreuses tâches de traitement des images dans l'industrie de la mode. Même si la segmentation est utile en elle-même pour isoler un vêtement d'une tenue à des fins d'affichage, elle est surtout utilisée comme étape de traitement intermédiaire pour de nombreuses applications : en essayage virtuel pour obtenir un vêtement source [Minar et al., 2020], dans l'obtention de représentations incorporant de la sémantique VSE (***V**isual **S**emantic **E**mbedding*) pour retrouver les produits à partir d'une tenue [Bastan et al., 2020], ou encore dans des applications sur les couleurs pour l'harmonie des coloris [Goree et Crandall, 2020].

Outre ces applications dans lesquelles la segmentation est déjà bien implantée, d'autres cas d'utilisation qui s'appuient exclusivement sur la détection comme étape de pré-traitement peuvent être améliorés grâce à la segmentation. Par exemple, la récupération des vêtements est traditionnellement effectuée de manière grossière grâce avec des boîtes englobantes [Ji et al., 2020a] mais la segmentation a déjà fait ses preuves en tant qu'alternative viable [Bhattacharyya et Nag, 2020].

Au fil des années, de nombreuses méthodes de segmentation ont été proposées (section 2.2). Cependant, sélectionner la meilleure d'entre elles pour un cas d'utilisation spécifique n'est pas aisé (section 2.6). Dans les faits, l'approche standard actuelle pour évaluer une architecture de segmentation est le calcul de la moyenne des précisions moyennes (*mAP : **m**ean **A**verage **P**recision*) au niveau corpus, qui s'appuie sur l'intersection sur l'union (*IoU: **I**ntersection **O**ver **U**nion*) entre les masques estimés et vérités terrain. Son principal atout réside dans sa capacité à résumer les performances grâce à une mesure unique. Néanmoins, cette approche est affectée par deux limites importantes. Elle ne capture pas correctement la qualité du contour et ne tient pas compte du contenu associé aux masques identifiés.

Ces inconvénients suggèrent qu'il est nécessaire d'adopter une méthode d'évaluation plus vaste, incluant différents aspects qui sont déterminants dans le contexte de la mode. Un protocole d'évaluation fournissant des informations sur la qualité des masques prédits serait particulièrement intéressant pour suivre l'entraînement des modèles, pour la sélection des méthodes et enfin, pour analyser la dérive dans le temps des modèles en production. Par conséquent, pour évaluer les architectures de segmentation, nous proposons, d'adopter une approche qui s'appuie sur trois niveaux : globale, contour et contenu.

Dans ce chapitre, nous commençons par étudier les différents protocoles, méthodes et mesures d'évaluation existants dans la section 3.2. Dans la section 3.3 nous rappelons les conditions retenues pour conduire nos expériences et nous précisons les détails d'implémentations des mesures choisies. Dans la section 3.4 nous présentons des résultats nous permettant de sélectionner

les méthodes de segmentation qui répondent le mieux à notre cas d'usage. Enfin, dans la section 3.5 nous concluons sur l'utilisation de la segmentation dans notre processus plus global de caractérisation de vêtements.

3.2 Mesures et évaluation

Généralement, l'évaluation permet d'estimer et de valider l'apport d'une proposition. Dans un contexte industriel, cette évaluation permet entre autre de quantifier le taux d'erreur des méthodes et de sélectionner celle qui en fait le moins. En effet, les erreurs sont critiques dans un but d'exploitation des modèles, car elles doivent être corrigées manuellement. Ces corrections reposent le plus souvent sur la mise en place de procédures coûteuses. Il est donc primordial de définir un protocole de validation et de test. Hors, souvent moins d'efforts sont fournis sur cette tâche que sur la définition de nouvelles méthodes de segmentation [Y. J. Zhang, 1996]. L'évaluation a pour but de confronter un modèle au cas d'usage auquel il est supposé répondre. Le modèle peut alors être validé ou sanctionné par le biais de métriques, une métrique étant une mesure.

L'évaluation est critique à tous les niveaux du cycle de vie d'un modèle. Par exemple, elle est critique, comme contrôle lors de l'entraînement, puis lors de la sélection du ou des modèles répondant au mieux à la tâche, enfin comme détection de dérive en production.

Dans la section 3.2.1 nous présentons une organisation des méthodes d'évaluation. En section 3.2.2, nous posons les notations utilisées dans ce chapitre. Enfin, dans la section 3.2.3, nous listerons différentes mesures permettant de calculer la similarité entre deux masques et comment les utiliser pour obtenir une évaluation des méthodes sur un corpus.

3.2.1 Ontologie d'évaluation

Les méthodes d'évaluation peuvent être regroupées selon la méthodologie suivie. H. Zhang et al. en 2008 proposent ainsi une ontologie qui regroupe les différentes évaluations possibles en quatre niveaux dichotomiques (*fig.* 3.1). Le premier fait la distinction entre les évaluations subjectives et objectives.

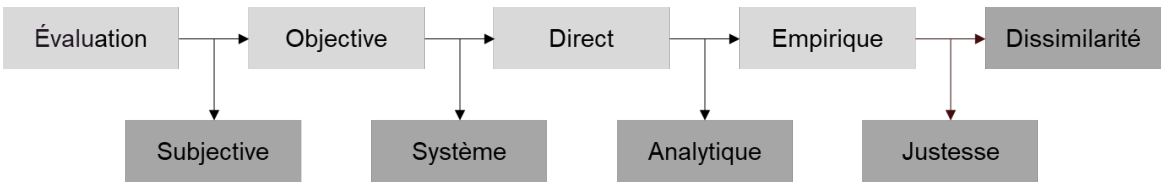


Figure 3.1 – Ontologie des méthodes d'évaluation proposée par H. Zhang et al. en 2008. (Niveaux intermédiaires plus clairs)

Subjectif - Objectif

L'évaluation subjective repose sur le jugement humain. L'observateur prend alors la décision de qualifier l'algorithme comme répondant ou non à la problématique. Cette décision peut être nuancée en utilisant un système de notation, il est alors nécessaire de définir et de poser un cadre à la prise de décision. De plus, les avis étant possiblement divergents, ces notes doivent être agrégées [Q. Wang et Z. Wang, 2009]. Ce type d'évaluation peut permettre de s'appuyer sur l'avis d'experts du domaine et, dans un contexte industriel, de s'appuyer sur les retours d'expériences d'utilisateur. À l'inverse, l'évaluation objective que nous considérons tend à simplifier, accélérer, automatiser l'évaluation et s'abstraire ou contrôler les biais et variances dû à l'interprétation humaine. L'évaluation peut être réalisée sur le système global ou sur la méthode directement.

Système - Direct

L'évaluation au niveau système s'opère en remettant la méthode dans son contexte d'utilisation finale, puis en évaluant le système qui en découle. La segmentation est souvent une étape intermédiaire permettant de répondre à un autre problème. L'évaluation au niveau système permet alors de déterminer indirectement si cette étape de segmentation est adaptée au problème. Dans notre cas, cela reviendrait à évaluer la caractérisation des vêtements basée sur leurs segmentations. Dans ce chapitre, nous nous focalisons sur l'évaluation directe des méthodes, évaluation qui peut être analytique ou empirique.

Analytique - Empirique

L'évaluation analytique tend à qualifier ou quantifier des propriétés de la méthode indépendamment de son application. On distingue les évaluations liées : aux connaissances a priori, aux bases théoriques de l'algorithme, à la stratégie de traitement [Y. Chen et al., 2018]. Néanmoins, les réseaux de neurones profonds infèrent leurs règles d'extraction de descripteurs à partir d'un apprentissage. Il est donc difficile de les évaluer sans s'appuyer sur leurs prédictions. L'évaluation analytique des réseau de neurones à couche de convolution (*CNN: **C**onvolutional **N**eural **N**etwork*) est alors principalement l'étude des complexités en espace, en temps. Elle diffère ainsi des méthodes d'évaluation empirique qui reposent sur l'application de l'algorithme et l'analyse des résultats obtenus.

Justesse - Dissimilarité

Lors d'une évaluation empirique, ce sont les résultats du modèle qui sont évalués. Ce type d'évaluation se prête donc plutôt bien aux CNN. En effet, par le choix des mesures et en maitrisant les jeux de données d'évaluation, il est possible d'étudier le comportement d'un modèle face à un contexte. Cela permet d'interpréter les résultats et par extension le modèle. Globalement, les métriques et la sélection des données d'évaluation permettent de s'assurer empiriquement que le modèle répond correctement au problème. Cela soulève les limites du modèle face au cas d'usage et ainsi fixe le périmètre d'utilisation. Pour le problème de segmentation, Y. J. Zhang en 1996 distingue la justesse empirique (*empirical goodness*), où seul le masque prédit est considéré, de la dissimilarité empirique (*empirical discrepancy*), où le masque prédit est comparé à un masque « idéal ».

En apprentissage non supervisé, seules les prédictions du modèle sont disponibles. Une métrique de justesse ne peut donc s'appuyer que sur la mesure d'une caractéristique intrinsèque aux données et aux prédictions. Ce type de métrique est souvent une approximation de la perception humaine. Pour le cas de la segmentation, elle peut être, par exemple, une mesure de l'uniformité et l'homogénéité intra-régions et de la divergence inter-régions [H. Zhang et al., 2008].

En apprentissage supervisé, les modèles sont entraînés à reproduire une vérité terrain fournie. Il est alors naturel de mesurer la dissimilarité d'une prédiction à la vérité terrain disponible. Cette dissimilarité est mesurée grâce à une métrique de distance ou de similarité entre deux éléments. Plus la distance est faible et la similarité élevée, plus la prédiction est considérée correcte.

3.2.2 Notations

Nous notons M un masque de vérité terrain. Ce masque M est l'ensemble des pixels x d'une image $I \in \mathbb{R}^{H \times W}$ de largeur W et de hauteur H qui appartiennent à un objet. Le masque prédit $\hat{M}$ contient les pixels prédits comme appartenant à l'objet du masque de vérité terrain M.

Nous notons respectivement ∂M et $\partial \hat{M}$ les contours des masques M et $\hat{M}$. Ces contours contiennent les pixels délimitant les masques, c'est-à-dire la frontière des ensembles de pixels.

Au niveau masque, nous nous appuyons sur la prédiction de chaque pixel x par un modèle. Chaque pixel de l'image peut donc être prédit comme positif ou négatif. Les prédictions correctes des pixels sont donc des vrais positifs (TP) ou des vrais négatifs (TN), les prédictions erronées sont des pixels faux positifs (FP) ou faux négatifs (FN).

Au niveau corpus ou image, nous considérons le masque dans sa globalité comme prédiction d'un modèle. les masques prédits corrects sont donc des TP et les autres des FP. L'absence de

prédiction pour des masques de vérité terrain produit des FN.

3.2.3 Métriques d'évaluation de la segmentation

De nombreux travaux proposent de lister et d'organiser les mesures de dissimilarité (respectivement pour les métriques de justesse) utilisables pour évaluer la segmentation de façon supervisée [Taha et Hanbury, 2015 ; Unnikrishnan et Hebert, 2005] (respectivement de façon non supervisée [H. Zhang et al., 2008]). En apprentissage supervisé, nous disposons d'une vérité terrain qui nous permet de déterminer la qualité des prédictions. Cette qualification de la qualité est obtenue par le biais de mesure de distance ou similarité entre les masques de vérité terrain et les masques prédits. Ces mesures doivent donc être conçues pour correctement renseigner l'écart à l'attendu.

Le problème qui se pose alors est « quelle mesure entre deux masques choisir ? », mais surtout « Comment modéliser un masque ? », *c.-à-d.* comment et sur quoi s'appliquent les mesures. Un masque peut être modélisé de différentes manières. Un masque peut être vu comme un ensemble de pixels où chacun des pixels sont indépendants les uns des autres. Cependant, avec cette modélisation, la notion de localisation des pixels est perdue. Une autre modélisation est alors de donner plus d'importance à certains pixels selon leurs positions dans l'image et dans le masque. Enfin, dans le cas de la segmentation d'instances, le modèle doit aussi être en mesure de discriminer les différentes instances d'une étiquette. Cette problématique doit aussi être évaluée.

Ensemble de pixels

La segmentation peut être vue comme une classification où le but est d'attribuer une étiquette à chaque pixel. En apprentissage automatique, il s'agit de prédire la classe (l'étiquette) de chacun des pixels d'une image. Les modèles peuvent donc être évalués de façon empirique en considérant chaque prédiction comme la classification d'un individu (*c.-à-d.* un pixel). Les métriques d'évaluation généralement utilisées peuvent provenir de différents domaines : partitionnement de données (*ex.* l'indice de rand [Hubert et Arabie, 1985 ; Rand, 1971]), théorie de l'information (*ex.* l'information mutuelle [Russakoff et al., 2004 ; Viola et Wells III, 1997]), *etc.*

Plus classiquement, l'évaluation d'un modèle de classification repose sur l'obtention de la matrice de confusion. En segmentation, on considère les pixels appartenant au masque prédit comme positifs et les autres pixels de l'image comme négatifs. On peut alors obtenir les TP, FP et les TN, FN (*fig.* 3.2). À partir de cette matrice de confusion, différentes métriques peuvent être calculées, par exemple l'exactitude (*accuracy*) $\frac{TP+TN}{TP+FP+TN+FN}$.

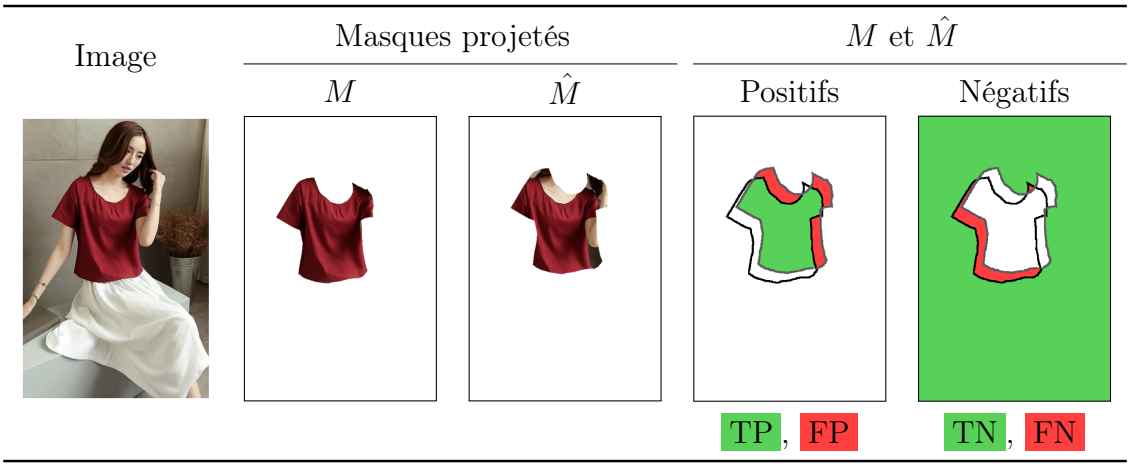


FIGURE 3.2 – Pixels vrai-faux positifs (TP, FP) et négatifs (TN, FN) d'un masque prédit $\hat{M}$ pour une référence en vérité terrain M.

On note cependant que de nombreuses métriques issues de la matrice de confusion sont équivalentes à des métriques statistiques ou à des métriques de recouvrement d'ensemble. En effet, comme montré par exemple dans la figure 3.2, les vrais positifs (TP) peuvent être obtenus à partir de l'intersection des deux masques ($TP = M \cap \hat{M}$). Ainsi, la mesure F1 ($\frac{2TP}{2TP+FP+FN}$) est équivalente à l'indice de Dice [DICE, 1945] ($\frac{2|M \cap \hat{M}|}{|M|+|\hat{M}|}$). L'indice de score critique ($\frac{TP}{TP+FP+FN}$) est lui équivalent à l'indice de Jaccard [JACCARD, 1912] ($\frac{|M \cap \hat{M}|}{|M \cup \hat{M}|}$). Ce dernier porte aussi le nom de son interprétation ensembliste dans la littérature : l'intersection sur l'union (*IoU: **I**ntersection **o**ver **U**nion*) ((3.1)). L'IoU était dans un premier temps la métrique utilisée par la communauté pour évaluer les performances des modèles sur le corpus VOC (*Pascal **V**isual **O**bject **C**lasses*) :

$$IoU(M, \hat{M}) = \frac{|M \cap \hat{M}|}{|M \cup \hat{M}|} = \frac{|\quad \cap \quad|}{|\quad \cup \quad|} \Leftrightarrow \frac{TP}{TP + FP + FN} = \frac{|\quad|}{|\quad|} \tag{3.1}$$

Position des pixels et contour

La segmentation peut aussi être formulée comme une tâche de détection des contours où la frontière d'un masque est un contour clos à prédire. L'évaluation des masques prédits peut donc se faire en utilisant des mesures de dissimilarité entre contours. Ces métriques permettent de renseigner l'exactitude des frontières des objets segmentés. La distance de Hausdorff et la mesure F1 des frontières (*boundary F1-measure*) [CSURKA et al., 2013 ; MARTIN et al., 2004]

sont deux exemples de telles métriques. Cependant, ces deux métriques sont coûteuses en temps de calcul pour la première, et difficile à interpréter pour l'autre. Fernandez-Moral et al. en 2018 ont alors proposé une amélioration de la mesure F1 des frontières avec BJ (***B**oundary Jaccard*). BJ compare la frontière $\partial\hat{M}$ d'un masque prédit $\hat{M}$ avec celle de la vérité terrain ∂M. Pour exprimer BJ, nous définissons d comme la distance d'un pixel x à un masque B : $d(\mathrm{x}, B) = \inf_{y\in B} ||\mathrm{x} - y||$ et la distance D d'un contour ∂A au masque B avec un seuil de précision $\theta > 0$

$$D(\partial A, B) = \sum_{\mathrm{x}\in\partial A,\ d(\mathrm{x},B)<\theta} \left(1 - (d(\mathrm{x}, B)/\theta)^2\right). \tag{3.2}$$

BJ s'exprime alors

$$BJ(M, \hat{M}) = \frac{D(\partial M, \hat{M}) + D(\partial\hat{M}, M)}{|\partial M| + |\partial\hat{M}|}. \tag{3.3}$$

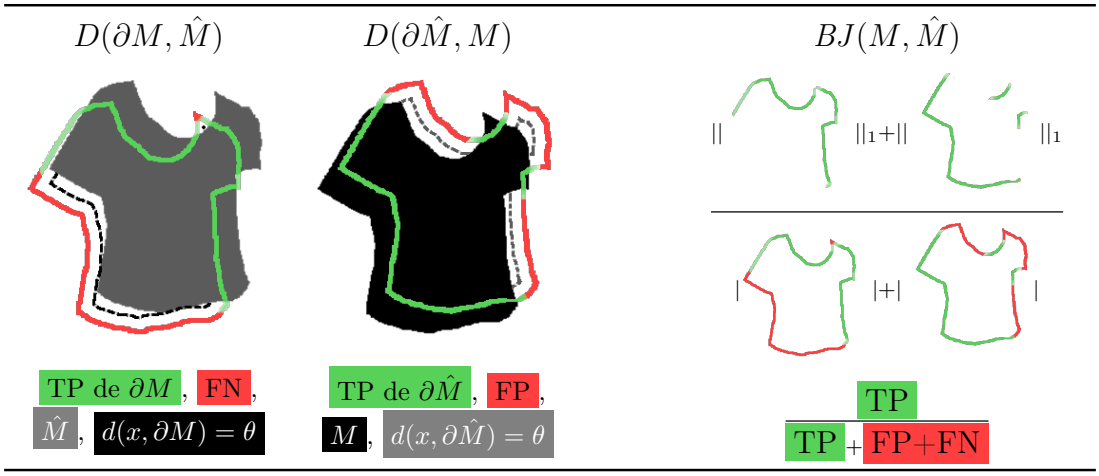


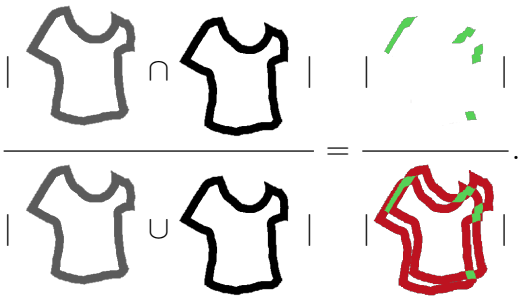
Figure 3.3 – Calcul de BJ entre M et $\hat{M}$, à partir des points vrais positifs et faux négatifs (TP,FN) du contour ∂M et des points vrais-faux positifs (TP,FP) du contour $\partial\hat{M}$, pour $\theta = 10$

Cette métrique apporte donc une expressivité aux vrais positifs (TP) (*c.-à-d.* les considérer plus ou moins « important », plus ou moins vert dans la figure 3.3) en les pondérant avec la distance d, là où la mesure F1 des frontières donne une réponse binaire avec cette même tolérance θ.

Une amélioration semblable de la mesure F1 des frontières est proposée par B. Cheng et al. en 2021 avec la B_{IoU} (***B**ondary IoU*). La B_{IoU} réalise l'IoU des contours élargis internes aux masques prédits $\hat{M}_\theta = \{\mathrm{x}|\mathrm{x} \in \hat{M}, d(\mathrm{x}, \partial\hat{M}) < \theta\}$ et de la vérité terrain $M_\theta = \{\mathrm{x}|\mathrm{x} \in$

$M, d(\mathrm{x}, \partial M) < \theta\}$:

$$B_{IoU}(M, \hat{M}) = \frac{|M_\theta \cap \hat{M}_\theta|}{|M_\theta \cup \hat{M}_\theta|} = \frac{|\quad \cap \quad|}{|\quad \cup \quad|} = \frac{|\quad|}{|\quad|}. \tag{3.4}$$



Cette métrique ajoute une pondération par la proportion de pixels des contours élargis au seuil θ. Cette pondération peut être vue comme équivalente à la distance d dans la métrique BJ. Cependant, pour la BJ la distance d est entre un point et un masque, pour la B_{IoU} cette distance d est entre un pixel et un contour élargit. Cela signifie qu'un même pixel peut être considéré comme un TP pour BJ et un FP ou FN pour B_{IoU}. Concrètement, les points vrais positifs (verts) contenus dans les masques de la figure 3.3 peuvent être des faux positifs ou faux négatifs (rouge) dans l'(3.4). La métrique B_{IoU} est donc plus stricte.

3.2.4 La mAP, de la recherche d'information à la segmentation d'instances

Pour une image donnée, un modèle de segmentation d'instances peut prédire plusieurs masques. Un modèle répondant parfaitement au problème en prédira un par objet localisé. Il est donc nécessaire d'être en mesure de pouvoir regrouper les métriques obtenues à partir de l'évaluation de chacun des masques. La moyenne de ces mesures peut être utilisée comme métrique niveau image ou même niveau corpus. Il peut aussi être intéressant d'analyser les distributions des métriques sur toutes les images d'un corpus.

Cependant, le but de la segmentation d'instances n'est pas seulement d'obtenir une localisation fine des objets dans une image, mais aussi de les distinguer. Les instances d'un certain nombre d'étiquettes doivent être dissociées. Une autre façon de formaliser le problème est « la capacité de retrouver différents éléments dans une image ». La segmentation d'instances partage cette caractéristique avec la détection. Retrouver correctement les objets, c'est-à-dire de façon exhaustive sans oubli, doit donc aussi être évalué.

L'approche actuelle dominante pour évaluer les méthodes de segmentation d'instances est la moyenne des précisions moyennes (*mAP : **m**ean **A**verage **P**recision*) (introduite depuis COCO (*Microsoft **C**ommon **O**bject in **CO**ntext*) [T.-Y. Lin, Maire et al., 2014]). La mAP est la moyenne des AP (***A**verage **P**recision*). Dans le cadre de la détection et de la segmentation

d'instances, l'AP_l est calculé par étiquette l pour toutes les étiquettes L :

$$mAP_\alpha = \sum_{l \in L} \frac{AP_{l,\alpha}}{|L|}, \tag{3.5}$$

où α est un seuil utilisé pour discriminer les vrais des faux positifs nécessaires pour calculer la courbe de précision en fonction du rappel. Nous rappelons que l'AP est l'aire sous la courbe précision-rappel.

Ici, un positif est un objet détecté. Une métrique niveau masque (ou boîte englobante pour la détection) est donc requise en conjonction du seuil α pour déterminer si une instance donnée est un TP ou un FP. La métrique qui remplit ce rôle est l'IoU ((3.1)). Les masques prédits sont appariés à une vérité terrain selon leur plus haute valeur d'IoU si elle est supérieure au seuil α. À partir de cet appariement, chaque masque de vérité terrain non associé est un FN. Pour chaque vérité terrain et leurs masques prédits associés, les prédictions qui ont les plus hautes IoU sont des TP, et les autres sont des FP.

3.3 Proposition de protocole d'évaluation

Dans cette section, nous présentons le protocole retenu pour évaluer les modèles de segmentation d'instances de vêtements dans des images. Dans la section 3.3.1 nous rapportons certaines limites de l'IoU et de la mAP qui sont couramment utilisés pour évaluer les modèles. En effet, il n'existe pas de métrique parfaite. Ce constat est également partagé entre autre par Reinke et al. en 2021. Chaque métrique dispose de ses propres désavantages et écueils d'évaluation. Dans la section 3.3.2, nous proposons alors de combler certaines limites liées à notre cas d'usage par trois axes d'évaluation.

3.3.1 Limites de la mAP et de l'IoU

La mAP repose sur l'IoU pour discriminer les « vrais » des « faux » masques prédits. Cependant, si utiliser exclusivement l'IoU peut avoir du sens dans un problème de localisation grossière tel que la détection, plusieurs limitations apparaissent lorsqu'on l'utilise pour une tâche plus fine telle que la segmentation d'instances. Par exemple, dans la figure 3.4 tous les masques M_i ont une IoU similaire à la vérité terrain M malgré le fait qu'ils soient relativement différents (masque décalé, sur et sous segmentation, boîte très grossière). Ces masques sont néanmoins tous des TP viables par exemple pour la $mAP_{0.5}$. Toutefois, il est clairement visible que l'IoU échoue à caractériser les erreurs de détection des contours (*ex.* le masque grossier M_4 et les erreurs de contenu (*ex.* le masque M_3 obtenu en agrandissant M). Ces exemples

permettent une analyse subjective de ce qu'est un bon résultat de segmentation.

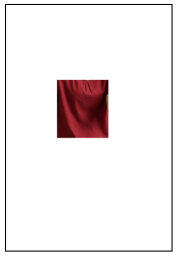
image	M	masques générés			
		M_1	M_2	M_3	M_4
					
	IoU	0.614	0.609	0.609	0.610

FIGURE 3.4 – Masques de T-shirt M_i générés manuellement de telle sorte que leur IoU à la vérité terrain M ait approximativement la même valeur 0.61.

Une des limites principales de la mAP basée sur l'IoU est qu'elle ne s'appuie que sur la modélisation d'un masque comme un ensemble de classification indépendante. En effet, chacun des pixels est traité de façon indépendante et seule l'information d'inclusion dans un masque est considérée. L'information contenue dans les pixels (*ex.* la couleur) et leur localisation dans l'image ne sont pas pris en compte.

3.3.2 Métriques d'évaluation au niveau masques

Afin d'analyser en profondeur la qualité d'un masque prédit, nous proposons une évaluation multi-niveaux décomposée en trois axes dans le contexte de la segmentation d'instances de vêtements dans l'industrie de la mode. Effectivement, il n'existe pas de mesure parfaite répondant à tous les cas d'usage [REINKE et al., 2021]. Une métrique conçue pour capturer différents aspects en une unique valeur ne permettrait pas de différencier les cas d'erreurs. ARBELAEZ et al. en 2010 ; CSURKA et al. en 2013 proposent d'évaluer le recouvrement des masques et le contour séparément avec l'IoU et la mesure F1 des frontières. B. CHENG et al. en 2021 font la même proposition, mais avec la B_{IoU}. Avec notre évaluation multi-niveaux, nous proposons d'exploiter toutes les informations disponibles au niveau du pixel : l'appartenance au masque (global), la localisation (contour) et la couleur (contenu). Dans la suite, nous sélectionnons une métrique candidate pour chaque niveau d'évaluation.

Métrique d'évaluation globale

Par niveau global, nous entendons l'évaluation des masques sous la forme d'ensemble de pixels, où seule l'information d'inclusion est évaluée. Cette inclusion à un masque peut aussi

être vue comme une classification niveau pixel. À ce titre, nous rappelons donc que les métriques issues de la matrice de confusion sont directement équivalentes à des mesures de recouvrement d'ensemble (section 3.2.3).

De surcroît, certaines de ces mesures sont aussi relativement équivalentes entre elles, il est possible d'obtenir l'une à partir d'une autre. Ces métriques mesurent les mêmes aspects. Par exemple, L'indice de Jaccard (IoU) est lié à l'indice de Dice par $Dice = \frac{2IoU}{1+IoU}$, l'indice de Tversky [Tversky, 1977] en est une généralisation.

Taha et Hanbury en 2015 ont aussi montré que de nombreuses métriques n'utilisant que l'information d'inclusion à un masque sont fortement corrélées entre elles. Il apparaît alors que de telles métriques capturent les mêmes aspects et évaluent « globalement » la segmentation. Il n'est donc pas nécessaire de les multiplier.

L'évaluation est simplifiée si on ne s'appuie que sur une seule métrique globale. L'IoU (3.1) est une bonne métrique candidate. En effet, c'est une métrique populaire pour évaluer la segmentation et bien établie dans la littérature. Elle fournit une lecture efficace de la qualité globale du masque et bénéficie d'une facilité d'utilisation à des fins de comparaison.

Métrique d'évaluation des contours

La segmentation est souvent une étape intermédiaire d'un processus plus global. Dans certains cas, tous les pixels prédits dans un masque n'ont pas la même importance. Obtenir une information précise des frontières d'un masque peut être un des enjeux principaux. Par exemple, un contour de qualité peut grandement améliorer la qualité visuelle à des fins d'affichage. Dans notre cas, il est crucial que les masques préservent la géométrie et la localisation des vêtements.

L'évaluation des contours prend en compte la localisation relative d'un pixel dans l'image (et dans les masques où il est inclus). Évaluer cet axe séparément est souvent conseillé dans la littérature [Arbelaez et al., 2010 ; B. Cheng et al., 2021 ; Csurka et al., 2013]. Nous proposons d'évaluer le contour des masques prédits avec la BJ [Fernandez-Moral et al., 2018] (3.3). Pour rappel, la BJ nuance la discrimination binaire des TP et FP sans être trop restrictive avec une pondération basée sur la distance aux contours de la vérité terrain et des masques prédits (section 3.2.3).

Avec des images rastérisées (*c.-à-d.* que les localisations des pixels sont discrétisées), il est possible de pré-calculer les distances et les pondérations utilisées dans l'équation (3.2). La distance étant calculée au maximum à une longueur θ dans l'équation (3.2) nous pouvons pré-calculer un filtre $K \in \mathbb{R}^{w \times h}$ associé à cette valeur θ. La fenêtre du filtre est de taille

$w = h = 2 \times \theta + 1$ et ses poids sont obtenus à partir de leur distance au centre de la fenêtre :

$$K_{i,j} = 1 - \left(\left\| \begin{bmatrix} i \\ j \end{bmatrix} - \begin{bmatrix} \theta \\ \theta \end{bmatrix} \right\| / \theta \right)^2 . \tag{3.6}$$

Nous pouvons ainsi calculer la forme discrète de d dans l'équation (3.2) de la manière suivante. Si le pixel x appartient au masque B ($\mathbb{1}_B(x) = 1$), alors :

$$d(x_{i,j}, B) = \max_{\substack{i'=-\theta\cdots\theta \\ j'=-\theta\cdots\theta}} (\mathbb{1}_B(x_{i+i',j+j'}).K_{\theta-i'+1,\theta-j'+1}). \tag{3.7}$$

Enfin, le filtre K peut être appliqué sur tous les pixels $x_{i,j}$ de l'image de taille $W \times H$ si le pixel appartient au contour ∂A ($\mathbb{1}_{\partial A}(x_{i,j}) = 1$) :

$$D(\partial A, B) = \sum_{\substack{i=1\cdots W \\ j=1\cdots H}} (\mathbb{1}_{\partial A}(x_{i,j}).d(x, B)). \tag{3.8}$$

Pour calculer $D(\partial A, B)$ il faut appliquer la fenêtre K à tous les pixels de l'image. La complexité du calcul de $D(\partial A, B)$ est donc de l'ordre de $\mathcal{O}(W.H.\theta^2)$.

Nous proposons alors une autre implémentation utilisant l'opérateur morphologique de dilatation $\oplus_m$ pour calculer $D(\partial A, B)$ (3.2)(3.8). Nous notons $D(\partial A, B) = TP_{\partial A}$, ∂A et B sont dans ce cas des images binaires où 1 signifie l'inclusion du pixel et 0 l'exclusion. Le but

Algorithm 1: Implémentation morphologique du calcul de $D(\partial A, B)$ (3.8)

Data: ∂A et B deux images binaires, K' un filtre
Result: $TP_{\partial A}$

1 $TP_{\partial A} \leftarrow 0$;
2 **for** $\delta = 0 \cdots \theta$ **do**
3 $\quad C \leftarrow \partial A \cap B$;
4 $\quad TP_{\partial A} \leftarrow TP_{\partial A} + \|C\|_1 \times (1 - (\frac{\delta}{\theta})^2)$;
5 $\quad \partial A \leftarrow \partial A \backslash C$;
6 $\quad B \leftarrow B \oplus_m K'$;
7 **end**
8 **return** $TP_{\partial A}$;

est alors de progressivement agrandir le masque B par dilatation avec un filtre K' (ligne 6 de l'algorithme 1. Puis, à partir de l'intersection entre ce masque agrandit avec le contour (ligne 3 de l'algorithme 1), les pixels sont itérativement retirés du contour (ligne 5 de l'algorithme 1) et ajoutés aux résultats (ligne 4 de l'algorithme 1 avec une pondération). La pondération δ est similaire à la distance d pour D dans l'équation (3.2). En répétant l'opération avec plusieurs

δ on obtient progressivement les pixels de plus en plus éloignés jusqu'à θ.

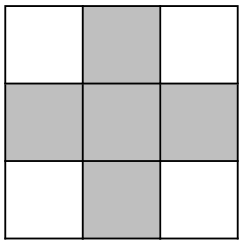


FIGURE 3.5 – Filtre binaire : « croix »

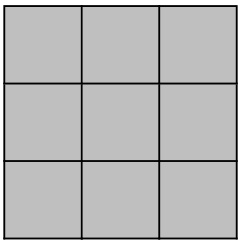


FIGURE 3.6 – Filtre binaire : « plein »

Pour $\delta \in \mathbf{N}^*$, utiliser le filtre présenté dans la figure 3.5 comme filtre K' est strictement équivalent à utiliser la norme 1 pour le calcul de la distance dans d (3.2) et celui de la figure 3.6 équivalent à une norme ∞. De plus, l'algorithme 1 parcourt l'image θ fois ce qui donne une complexité de l'ordre de $\mathcal{O}(\theta.W.H)$. Empiriquement (*fig.* 3.7), on observe cette complexité en temps entre les deux implémentations. Pour $\theta = 5$ on observe un facteur dix entre la moyenne des calculs de la forme discrétisée (284.5 ms) et de l'implémentation avec dilatation (28.4 ms) (*tab.* 3.1).

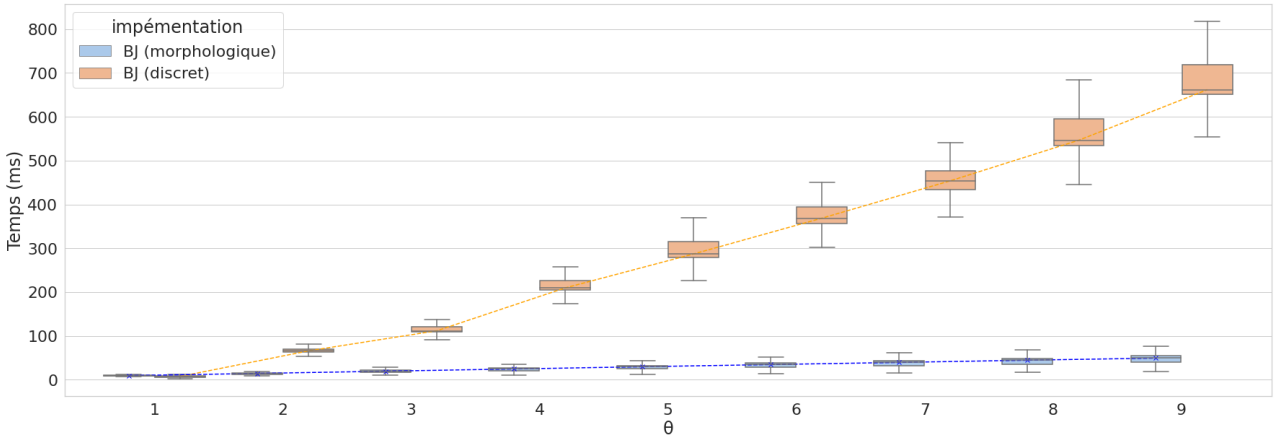


FIGURE 3.7 – Diagramme en boîte des temps de calcul des implémentations de BJ sur 1 000 paires de masques (vérité terrain et prédit par « Mask » réseau de neurones à couche de convolution à partir de région (*R-CNN: **R**egion based **C**onvolutional **N**eural **N**etwork*)) pour $\theta = 1 \cdots 9$.

Métrique d'évaluation du contenu

La segmentation est dans notre cas une étape intermédiaire de la caractérisation de vêtements. La couleur et la texture sont entre autre des caractéristiques à retrouver. Il est donc primordial de s'assurer de la validité de l'information contenue dans le masque. Pour évaluer cet aspect, nous proposons une métrique mesurant la différence de contenu entre deux masques.

θ	1	2	3	4	5	6	7	8	9
BJ (discrétisé) *moyenne ms*	7.4	63.8	111.1	208.5	284.5	359.0	439.0	534.0	643.9
BJ (morphologie) *moyenne ms*	9.4	13.9	19.1	23.6	28.4	33.2	38.1	42.5	47.5

TABLE 3.1 – Moyennes des temps de calcul de la figure 3.7.

Afin d'évaluer cette différence dans les contenus segmentés, nous proposons d'analyser la distribution des couleurs au sein de leurs masques. En pratique, nous considérons la distribution des couleurs des pixels dans un masque comme un histogramme discret $3D$ de n *bins* défini dans l'espace couleur L*a*b* *CIE76* [ROBERTSON et al., 1977]. Cette espace est supposé mieux organiser les couleurs selon la perception humaine (les espaces de couleurs sont davantage développés dans le chapitre 4). L'évaluation est alors réalisée en comparant les histogrammes de couleur de la vérité terrain et des masques estimés. Les outils de comparaison d'histogrammes discrets peuvent être divisés en quatre catégories principales [PUZICHA et al., 1999] : distance heuristique entre histogrammes, test statistique, divergence issue de la théorie de l'information et distance de transport entre distributions.

Nous proposons ici de nous appuyer sur la distance de transport optimal, aussi appelée EMD (***E**arth **M**over's **D**istance*) [RUBNER et al., 1998]. Cette distance s'est en effet révélée être une métrique robuste pour la recherche d'images [PELE et WERMAN, 2009], le transfert de couleurs [PITIÉ et al., 2007] ou la segmentation d'images [PAPADAKIS et RABIN, 2017]. Contrairement aux mesures classiques *bin*-à-*bin* telles que la divergence de Kullback-Leibler, l'EMD est naturellement conçue pour prendre en compte les *bins* vides. Par contre, il n'existe pas de formule explicite pour calculer l'EMD entre des histogrammes définis sur des espaces de dimension supérieure à un. Comme nous traitons des histogrammes couleur $3D$, nous devons résoudre un problème d'optimisation linéaire pour calculer l'EMD.

Nous désignons par $h_{\hat{M}}$ et h_M les histogrammes de couleur des pixels contenus respectivement dans les masques prédits et de vérité terrain $\hat{M}$ et M. L'EMD est obtenu à partir d'une matrice de couplage f qui donne le coût minimal pour transporter $h_{\hat{M}}$ vers h_M, étant donné une matrice de taille $n \times n$ dont les composantes $c_{i,j}$ représentent un coût entre les *bins* i et j. La valeur optimale f_{ij}, qui indique la part de la masse dans la *bin* d'histogramme $h_{\hat{M}}(i)$ transportée vers la *bin* d'histogramme $h_M(j)$, est estimée en résolvant

$$EMD(h_M, h_{\hat{M}}) = \inf_f \sum_i^n \sum_j^n f_{i,j} c_{i,j} \tag{3.9}$$

sujet aux contraintes : (i) $\sum_i^n f_{i,j} = h_{\hat{Y}}(j)$, $j : 1 \cdots n$, (ii) $\sum_j^n f_{i,j} = h_Y(i)$, $j : 1 \cdots n$ et (iii) $f_{i,j} \geq 0$, $i, j = 1 \cdots n$. Nous utilisons comme matrice de coût $c_{i,j} = ||b_i - b_j||$, avec $\{b_i\}_{i=1}^n$ les centres des *bins* de l'histogramme, et nous résolvons (3.9) avec un solveur linéaire.

Afin de définir une similarité à partir de l'EMD, nous proposons la transformation non-linéaire suivante :

$$sEMD(h_M, h_{\hat{M}}) = e^{(-\beta . EMD(h_M, h_{\hat{M}}))}. \tag{3.10}$$

Les expérimentations numériques suggèrent que prendre $\beta = 5$ et des histogrammes à 16^3 *bins* est un choix pertinent pour discriminer les histogrammes de couleur acceptables $h_{\hat{M}}$ de ceux qui sont visuellement trop différents de h_M.

Grâce à cette évaluation du contenu, nous sommes en mesure d'estimer la précision des couleurs des masques estimés. Pour certaines applications de la mode et de l'habillement, l'extraction des tissus des vêtements peut être aussi importante que les vêtements eux-mêmes. Pouvoir quantifier les erreurs basées sur la sur-segmentation (*ex.* incluant l'arrière-plan, un autre vêtement, *etc.*) et la sous-segmentation (*ex.* manquant les parties du vêtement composées de différents tissus) est extrêmement précieux.

3.3.3 Métrique d'évaluation au niveau du corpus

À partir des trois métriques sélectionnées et proposées (section 3.3.2), il est possible de distinguer les cas d'erreurs (présentés dans la figure 3.4) comme illustré dans la figure 3.8. L'IoU nous donne la précision globale des masques. En ajoutant la BJ et la $sEMD$ il est possible d'évaluer la capacité des modèles à bien estimer les contours et les contenus des vérités terrain. Ces trois axes : l'appartenance au masque (global), la localisation (contour) et la couleur (contenu), rendent les résultats et les modèles plus interprétables.

Ces trois métriques concernent l'évaluation individuelle des masques, mais peuvent aussi être utilisées sur l'ensemble d'un jeu de données. Appliquer ces métriques à un faible nombre d'images et de masques bien sélectionnés peut révéler des problèmes caractéristiques de segmentation produits par les modèles. Cependant, avec un jeu de données plus conséquent et représentatif du cas d'usage, souhaitable en évaluation, il devient difficile d'analyser tous les résultats des mesures de similarité entre masques prédits et de vérité terrain. Sur un corpus, l'information doit être agrégée afin d'analyser les résultats (section 3.2.4). On peut alors avoir recours à des statistiques récapitulatives à partir des distributions de ces résultats (*ex.* la moyenne : m_{IoU}, m_{BJ}, m_{sEMD}).

Comme autre métrique au niveau corpus, nous proposons d'adapter la mAP (3.5). Les métriques IoU (3.1), BJ (3.3), $sEMD$ (3.10) et le seuil α sont alors utilisés comme mesure sous-jacente de discrimination des TP. Chacune des formes adaptées sont respectivement

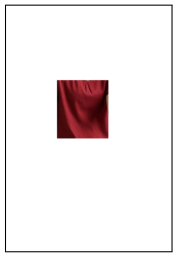
image	M	masques générés			
		M_1	M_2	M_3	M_4
					
	IoU	0.614	0.609	0.609	0.610
	BJ	**0.492**	0.447	<u>0.472</u>	0.459
	sEMD	0.807	<u>0.957</u>	0.644	**0.973**

FIGURE 3.8 – Les masques M_i de T-shirt générés manuellement que de la figure 3.4 sont reproduits avec leur IoU, BJ et sEMD à la vérité terrain M. Les scores les plus élevés sont en gras et les seconds soulignés.

nommées : $mAP_{IoU\alpha}$, $mAP_{BJ\alpha}$, $mAP_{sEMD\alpha}$.

3.4 Expérimentation

Dans cette partie, nous rapportons les résultats obtenus avec notre position de protocole d'évaluation présenté dans la section 3.3. Pour évaluer les contours avec la métrique BJ nous utilisons $\theta = 5$. Pour le contenu, nous rappelons que nous utilisons $\beta = 5$ et des histogrammes discrétisés en 16 *bins* par dimension pour la $sEMD$. Une surcouche de l'interface de programmation d'application (*API: **A**pplication **P**rogramming **I**nterface*) de COCO [1] a été réalisée pour y intégrer ces métriques.

Nous évaluons quatre méthodes dites en « en deux étapes » issues de R-CNN et une méthode dite « en une passe » issue de YOLO (***Y**ou **O**nly **L**ook **O**nce*). Suite aux travaux présentés dans le chapitre 2, les méthodes retenues sont : « Mask » R-CNN [HE, GKIOXARI et al., 2017], MS R-CNN (***M**ask **S**coring R-CNN*) [Z. HUANG et al., 2019], « Cascade » R-CNN [CAI et VASCONCELOS, 2018], HTC (***H**ybrid **T**ask **C**ascade*) [K. CHEN, PANG et al., 2019] et YOLACT [BOLYA et al., 2019]. Nous rappelons que ces méthodes sont présentées dans la section 2.2.3. Les modèles ont été entraînés pendant 5 epochs sur le corpus Deepfashion2 [GE et al., 2019] (chapitre 2 section 2.3.2) dans les mêmes conditions que celles présentées dans le chapitre 2 section 2.4.

1. github.com/cocodataset/cocoapi

3.4.1 Distribution des métriques

Afin d'obtenir les distributions des métriques (IoU, BJ, $sEMD$) à partir des prédictions des cinq méthodes (*fig.* 3.9), nous réalisons un appariement des masques de vérité terrain M et des masques prédits $\hat{M}$. Concrètement, nous associons à chaque vérité terrain, les prédictions qui se trouvent dans la même image et qui sont de même type (*ex.* t-shirt, short, *etc.*). Puis, nous conservons uniquement une prédiction par méthodes qui a la plus grande IoU avec la vérité terrain associée. Les vérités terrain non-associées sont rejetées.

	Mask R-CNN	MS R-CNN	Cascade R-CNN	HTC	YOLACT
vérité terrain ($\%M$)	98.43	98.31	96.83	99.90	96.47
prédiction ($\%\hat{M}$)	16.35	14.29	20.36	4.43	37.87

TABLE 3.2 – Pourcentage des vérités terrain M et des prédictions $\hat{M}$ appariées.

Le tableau 3.2 montre le pourcentage par méthodes de vérité terrain et de prédictions retenues dans les paires (M, $\hat{M}$). Globalement, la plupart des masques de vérité terrain sont associés à une prédiction. Cependant, HTC prédit de nombreux masques (95.57%) pouvant être considérés comme du bruit. YOLACT est en ce sens la méthode la plus précise (37.87%).

À partir de chacun des couples (M, $\hat{M}$), les trois métriques : IoU, BJ, $sEMD$ peuvent être calculées. La figure 3.9 fait alors figurer les distributions des résultats par métriques et par méthodes. Le modèle HTC donne les meilleurs résultats avec IoU. Cependant, l'ordre avec YOLACT est inversé pour les deux autres métriques (BJ, $sEMD$). Ce comportement est aussi observable avec « Cascade » R-CNN et MS R-CNN. YOLACT semble prédire des masques aux contenues et aux contours plus proches de la vérité terrain.

Décomposer l'évaluation en trois axes permet donc de mieux interpréter les prédictions. Afin de s'en assurer, la figure 3.10 présente les fonctions de densité jointe pour chaque pair de distribution de métriques. Pour des valeurs élevées de métrique, la densité est forte. Ce comportement est souhaitable, car un masque parfaitement précis globalement, l'est aussi sur les contours et le contenu. Pour des valeurs faibles, la densité est cependant bien diffuse. Il n'est pas possible à partir d'une métrique d'obtenir les deux autres si sa valeur est faible.

Les trois métriques apportent donc bien des informations différentes. L'ajout de la BJ et de la $sEMD$ permettent d'obtenir des interprétations supplémentaires des modèles. À titre de comparaison, la figure 3.11 montre la fonction de densité jointe de deux métriques fortement corrélées. La BJ est presque linéairement fonction de B_{IoU}. La B_{IoU} étant néanmoins plus stricte, les valeurs sont globalement plus faibles.

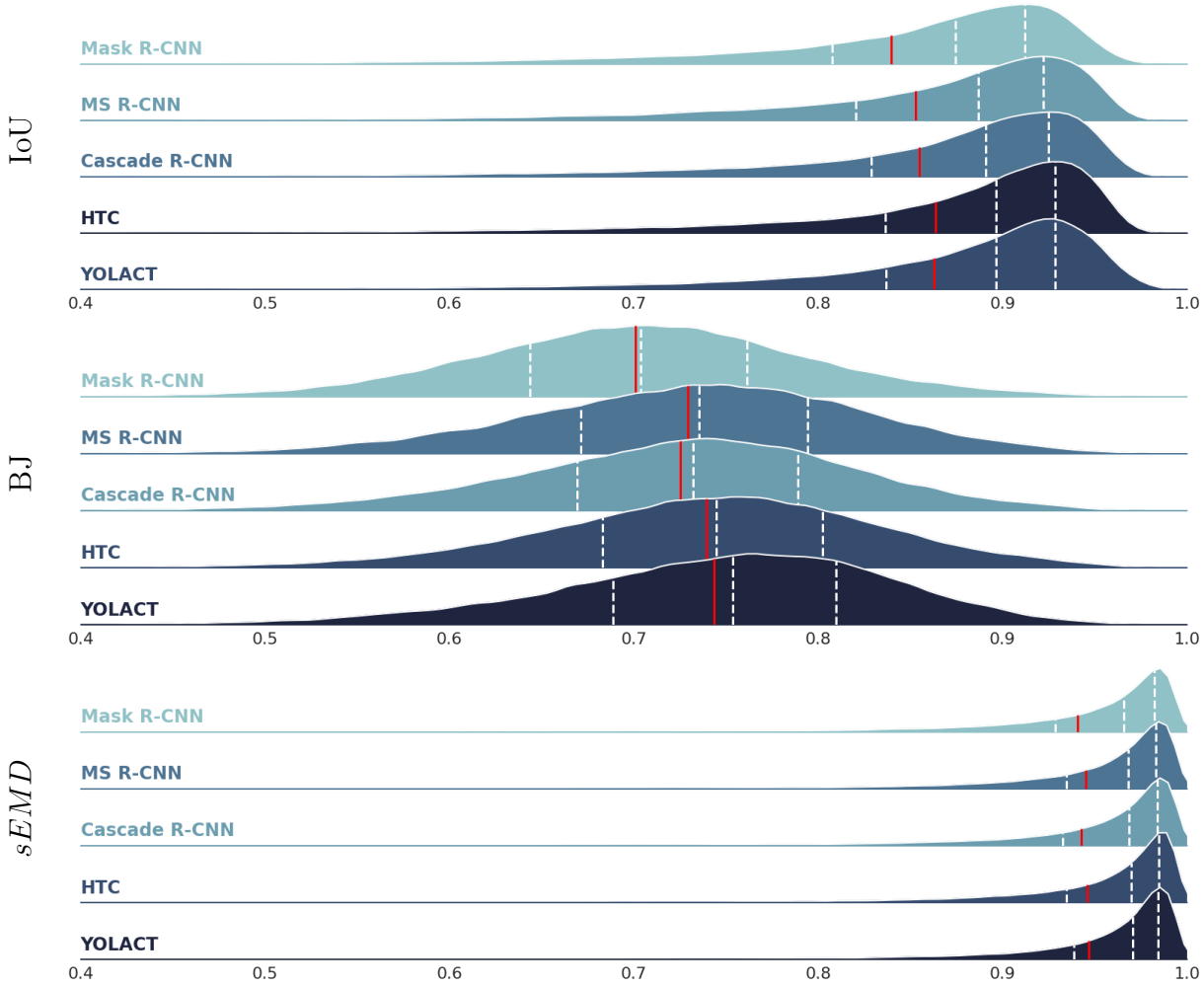


FIGURE 3.9 – Fonctions de densité des métriques appliquées sur l'appariement, par méthodes, des masques de vérités terrain et prédits ayant la plus grande IoU (*tab.* 3.2). Les lignes blanches en pointillés représentent les quartiles et la ligne rouge la moyenne. Les fonctions de densité varient du bleu clair au bleu foncé selon le classement croissant obtenu à partir de leurs moyennes.

3.4.2 mAP et sélection de la meilleure méthode

Les différentes méthodes de segmentation produisent de nombreuses prédictions de masques pour chaque image et chaque type de vêtements. Avec les distributions des métriques (*fig.* 3.9) nous évaluons les meilleurs masques candidats en ayant connaissance de la vérité terrain. Cette approche nous permet d'obtenir la qualité maximale des masques prédits par les méthodes. Cependant, de nombreuses prédictions ont été rejetées (*tab.* 3.2), car elles sont des multiples localisations des mêmes objets. En conditions d'utilisation réelles, nous ne disposons pas de vérité terrain nous permettant de classer et de rejeter des multiples localisations.

Nous nous appuyons alors sur le score de prédiction du type de vêtements pour ordonner les prédictions par image et par type. Afin d'évaluer la capacité des modèles à retrouver et à

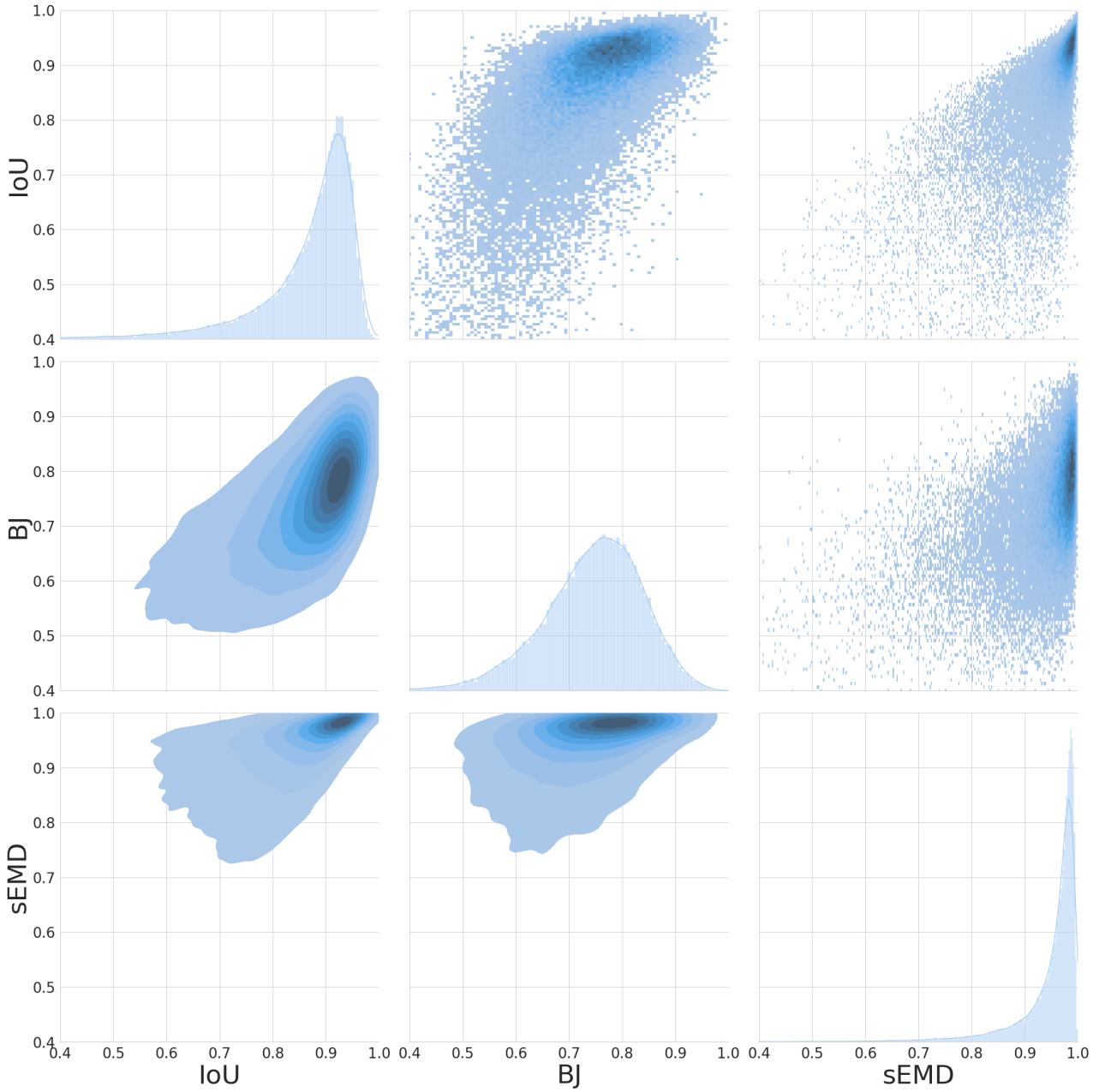


FIGURE 3.10 – Distributions jointes des valeurs des métriques sur les prédictions de YOLACT. Sur la diagonale figure les histogrammes et les fonctions de densité présentés dans la figure 3.9. Le coin inférieur gauche contient, les fonctions de densité jointes et le coin supérieur droit, les histogrammes $2D$. Plus la valeur des distributions jointes est élevée, plus le bleu est sombre.

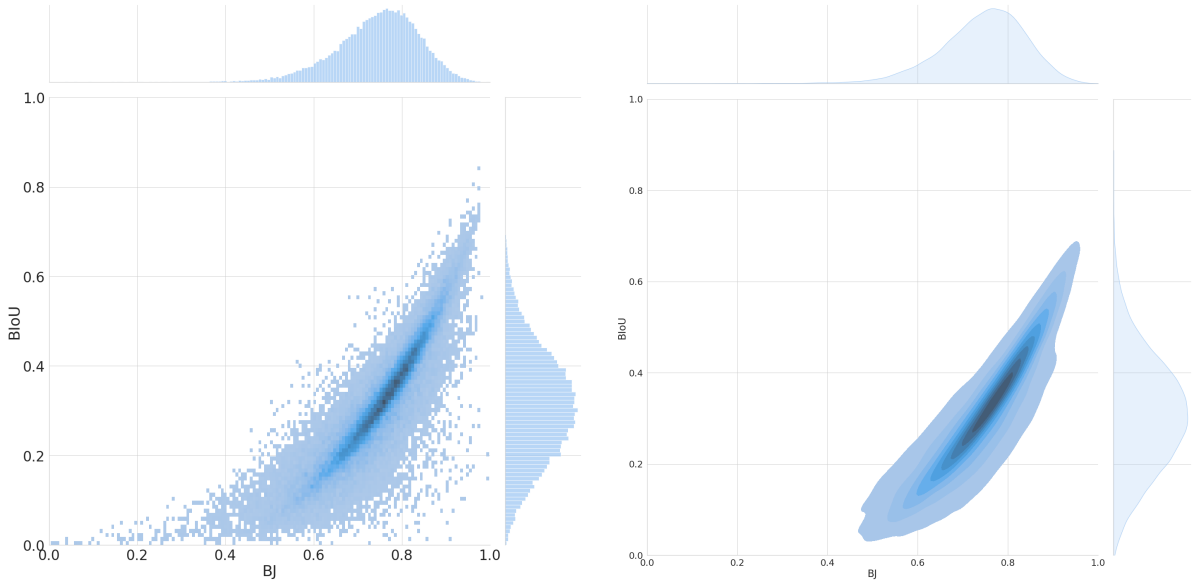


FIGURE 3.11 – Distribution jointe des valeurs de BJ et IoU sur les prédictions de YOLACT. Plus la valeur de la distribution jointe est élevée, plus le bleu est sombre dans l'histogramme $2D$ (à gauche) et la fonction de densité jointe (à droite).

Architecture	m_{IoU}	mAP_{IoU}			m_{BJ}	mAP_{BJ}			m_{sEMD}	mAP_{sEMD}		
		$[0.5, 0.95]$	0.5	0.75		$[0.5, 0.95]$	0.5	0.75		$[0.5, 0.95]$	0.5	0.75
Mask R-CNN	0.840	0.399	0.567	0.464	0.701	0.226	0.568	0.081	0.941	0.522	0.580	0.560
MS R-CNN	0.853	0.421	0.567	0.490	0.730	0.264	0.569	0.174	0.945	0.530	0.577	0.563
Cascade R-CNN	0.855	0.424	0.577	0.493	0.725	0.257	0.578	0.145	0.943	0.533	0.589	0.568
HTC	**0.863**	0.440	0.594	0.508	0.739	0.283	0.600	0.187	0.946	0.547	0.608	0.584
YOLACT	0.863	**0.516**	**0.687**	**0.601**	**0.744**	**0.341**	**0.689**	**0.265**	**0.947**	**0.642**	**0.699**	**0.679**

TABLE 3.3 – Évaluation après 5 epochs d'entraînement, les m_{IoU}, m_{BJ} et m_{sEMD} sont rapportées de la figure 3.9, la ligne dessous la mAP correspond aux valeurs de α' utilisées comme seuils.

distinguer les vêtements dans l'image, nous utilisons alors la mAP. Nous utilisons un seuil sur la IoU, la BJ et la $sEMD$ pour discriminer les masques prédits TP des FP, ce qui nous donne respectivement trois métriques niveau corpus : mAP_{IoU}, mAP_{BJ}, mAP_{sEMD}. Nous présentons dans le tableau 3.3 les résultats obtenus avec une moyenne des mAP pour des seuils allant de 0.5 à 0.95 par incrément de 0.05 et les mAP aux seuils spécifiques de 0.5 et 0.75.

YOLACT donne les meilleurs résultats après 5 epochs d'entraînement comme rapporté dans la tableau 3.3. De plus, le temps nécessaire pour l'entraîner est presque trois fois inférieur à celui nécessaire pour « Mask » R-CNN (*tab.* 3.4). HTC produit les seconds meilleurs résultats et est la meilleure méthode issue de R-CNN en termes de qualité des masques prédits. Cependant, la complexité de son architecture accroit de 50% le temps nécessaire à son entraînement par rapport à « Mask » R-CNN et d'un facteur 4 par rapport à YOLACT. De plus, mAP_{IoU} est

	Mask R-CNN	MS R-CNN	Cascade R-CNN	HTC	YOLACT
temps d'inférence *s/image*	0.11	0.11	0.13	0.22	0.07
temps d'entraînement *h/epoch*	10.94	11.20	14.68	16.87	3.89

TABLE 3.4 – Complexités en temps des cinq méthodes évaluées.

nettement plus faible que celle de YOLACT, malgré une m_{IoU} légèrement plus élevée. Cette différence s'explique par les multiples duplications de masques prédits par HTC qui pénalise son mAP.

Les $mAP_{IoU\alpha}$, $mAP_{BJ\alpha}$ et $mAP_{sEMD\alpha}$ diminuent légèrement plus rapidement avec l'augmentation de α pour « Mask » R-CNN et « Cascade » R-CNN par rapport aux autres modèles. Cela suggère que les deux méthodes produisent plus de masques de qualité moyenne. En ce sens, MS R-CNN améliore la qualité des masques prédits par rapport à « Mask » R-CNN pour une faible augmentation de la complexité en temps. En effet, ses m_{BJ} et m_{sEMD} sont supérieures à celles de « Mask » R-CNN et de « Cascade » R-CNN.

3.5 Conclusion

Dans ce chapitre, nous avons proposé un protocole d'évaluation de la segmentation d'instances de vêtement. Ce protocole, publié et présenté en conférence [JOUANNEAU et al., 2021], s'articule en trois axes : global, contour, contenu et sur deux niveaux : masques, corpus. Ce protocole a été appliqué à l'évaluation de cinq modèles entraînés sur Deepfashion2. Nous avons pu montrer qu'il permet effectivement d'améliorer l'interprétation des résultats et de donner des indications utiles sur l'inférence des modèles de segmentation. Cela nous a aussi permis de sélectionner la méthode YOLACT pour réaliser la localisation de vêtements dans une image.

Le protocole pourrait néanmoins être amélioré. Au niveau masque, la $sEMD$ pourrait être perfectionnée. Donnant des valeurs globalement élevées, une meilleure normalisation serait utile afin de mieux exploiter toutes les valeurs possibles entre $[0, 1]$. Étant coûteuse en calcul, son implémentation pourrait être optimisée. Enfin, seul la distribution des couleurs est utilisée, il serait alors intéressant d'aussi prendre en compte les textures.

Au niveau corpus, les méthodes prédisent de nombreux masques dont la plupart doivent être rejetés. Pour obtenir les distributions des métriques, nous conservons uniquement les masques les plus pertinents en regard de la vérité terrain. Cette méthode a pour avantage d'évaluer une qualité optimiste des prédictions, mais ne correspond cependant pas à une application réelle

des méthodes. Lors de l'application des méthodes, la vérité terrain n'est pas disponible. Il serait judicieux d'aussi évaluer les méthodes avec d'autres stratégies d'appariement des masques de vérité terrain et prédits. Enfin, la mAP prend bien en compte cette problématique. Elle permet d'évaluer la capacité des méthodes à retrouver les vêtements dans des images. Le seuil utilisé dans la mAP pourrait néanmoins être adapté à chaque métrique (BJ et $sEMD$).

Toutefois, en l'état, Le protocole proposé permet de classer les méthodes. Il pourrait alors servir à évaluer de futures propositions de méthodes de segmentation d'instances. Ces résultats pourraient ensuite être comparés à ceux des méthodes déjà évaluées. Évaluer ces méthodes sur d'autre corpus d'images permettrait aussi de garantir la qualité des masques prédits. Pour de futures expérimentations, si le classement des méthodes n'est pas le même selon les axes d'évaluation, un modèle par cas d'utilisation pourrait être sélectionné (*ex.* affichage, extraction de couleur, *etc.*). Il serait alors question de fusionner leurs prédictions.

Toutefois, lors de nos expérimentations, YOLACT donnait les meilleurs résultats sur tous les axes d'évaluation. Cette méthode semble donc en mesure de prédire des masques de bonne qualité aux contenus similaires à la vérité terrain. Nous présentons alors son application à la caractérisation de vêtement dans le chapitre suivant.

Chapitre 4

De la segmentation à la caractérisation de vêtements

Résumé : Dans ce chapitre, nous présentons deux méthodes distinctes pour caractériser (i) le type et le motif tissu, et (ii) la couleur dominante des vêtements à partir des masques obtenus par segmentation d'instances. La première méthode, basée sur un réseau de neurones à couches de convolutions, utilise les images et les masques pour classifier les vêtements en fonction de deux caractéristiques : leur type et le motif sur leur tissu. L'intégration de patchs dans l'architecture permet alors d'améliorer les performances de classification des motifs. Pour ce problème, nous avons défini une taxonomie des types et motifs possibles des vêtements, afin de constituer un corpus d'entraînement. La deuxième approche s'intéresse à l'extraction de la couleur dominante du vêtement. Le choix de l'espace de couleur et de sa discrétisation sont étudiés afin de proposer une méthode simple et capable de s'appuyer efficacement sur des nomenclatures de couleurs standard dans la mode. Les méthodes obtenues ont pu être déployées en test à Lectra et les travaux ont été reversés à la filiale Retviews.

4.1 Introduction

Dans notre objectif d'apposition de mots clefs sur des images de vêtements, la segmentation tient un rôle essentiel. Elle nous permet de distinguer les vêtements dans l'image, mais surtout de les localiser finement. À partir des travaux présentés dans le chapitre 2 nous disposons de modèles de localisation sous la forme d'un masque pour chaque vêtement contenu dans une image. Les travaux présentés dans le chapitre 3 nous ont permis de sélectionner le modèle YOLACT donnant la meilleure qualité de masques.

Dans ce chapitre, nous tirons parti des masques de segmentation pour caractériser les mots clefs qui décrivent tous les vêtements d'une image. Pour notre usage, la description des caractéristiques des vêtements prend la forme d'étiquettes apposées sur l'image. Cet étiquetage des images est utile à plusieurs fins. Il peut servir à indexer les images dans des moteurs de recherche et permettre leur récupération par requêtes. Il peut aussi servir pour des analyses statistiques à grande échelle.

Nos travaux concernent trois types d'étiquettes : le type du vêtement, le motif du tissu et la couleur dominante du vêtement. Ils sont néanmoins transposables à d'autres caractéristiques, par exemple le sexe et la tranche d'âge pour lequel le vêtement est conçu.

La caractérisation peut être réalisée de différentes manières, avec des modèles de classification ou de simples règles d'extraction de l'information. Notre approche est alors de développer empiriquement des méthodes dans le cadre d'une exploitation industrielle des expérimentations. Les différentes valeurs possibles que peuvent prendre chaque type d'étiquette définissent la taxonomie de la classification.

Pour réaliser nos expérimentations, une taxonomie de types et de motifs de vêtements a été définie avec Lectra et sa filiale Retviews. Nous avons alors collecté et annoté automatiquement des images par rapport à cette taxonomie. Le corpus de données finalement obtenu est présenté dans la section 4.2.

Dans la section 4.3 nous utilisons ce corpus pour proposer une méthode de classification supervisée dédiée au type et au motif d'un vêtement. Nous décrivons une première architecture de classification utilisant une image et un masque de segmentation. Nous montrons alors comment l'intégration de patchs dans l'architecture permet d'améliorer la classification des motifs imprimés sur les tissus.

Dans la section 4.4 nous présentons une méthode empirique d'extraction de la couleur dominante du vêtement en tant que référence couleur dans une bibliothèque.

Enfin, nous soulevons des limites et proposons des pistes d'améliorations pour nos méthodes dans la section 4.5.

4.2 Corpus de données

Nous allons à présent introduire les corpus spécialisés dans la caractérisation du contenu, dont certains ont déjà été présentés dans la section 2.3 du chapitre 2. Dans le cadre de la caractérisation de vêtements, ces corpus sont composés d'images annotées au minimum avec le type d'un des vêtements contenus. Ces corpus peuvent contenir d'autres informations, par exemple différentes caractéristiques du tissu, de la matière, des pièces (manches, cols, poches). Ces annotations sont le plus souvent des étiquettes qui peuvent servir à entraîner des modèles de classification par apprentissage.

Pour des tâches de classification de caractéristiques, il est nécessaire de définir une taxonomie, *c.-à-d.* les différentes étiquettes possibles et leur nom pour chaque caractéristique. Les corpus ACWS (***A**pparel **C**lassifcation **W**ith **S**tyle*) [1] [Bossard et al., 2012], WTBI (***W**here **T**o **B**uy **I**t*) [Hadi Kiapour et al., 2015] et Clothing1M [2] [Tong Xiao et al., 2015] mettent respectivement à disposition dans leurs versions publiques environ un million, 420 000 et 80 000 images annotées avec des étiquettes correspondant respectivement à 15, 14 et 11 types de vêtements. Dans un contexte plus large, le corpus par exemple iFashion-Attribute [3] [Guo et al., 2019] met à disposition environ un million d'images annotées avec de multiples étiquettes (*c.-à-d.* pas seulement le type de vêtement). Seules les annotations de 50 000 images ont néanmoins été vérifiées manuellement. iFashion-Attribute propose une taxonomie de différents éléments de vêtements comme par exemple les types de cols et de manches mais aussi du tissu (*ex.* le motif, la couleur, la matière).

Toutefois, nous souhaitons nous appuyer sur des données correspondant spécifiquement à notre problème industriel et considérer une taxonomie d'étiquettes qui soit exploitable par Lectra et ses filiales. Nous avons donc constitué un corpus en collectant des fiches produits [4] pour effectuer les travaux détaillés dans les parties suivantes. Ces fiches produits incorporent des images, des descriptions, des mots clefs pour chaque article de mode, *c.-à-d.* pour chaque vêtement. Après pré-traitement et nettoyage des données, 63 365 articles ont été conservés. Nous présentons les images de ces articles dans la section 4.2.1, la taxonomie des étiquettes dans la section 4.2.2 et les masques de segmentation dans la section 4.2.3.

1. data.vision.ee.ethz.ch/cvl/lbossard/accv12/
2. github.com/Cysu/noisy_label
3. github.com/visipedia/imat_fashion_comp
4. Nous ne détaillerons pas cette collecte pour des raisons de propriété intellectuelle. Les données sont similaires à celles pouvant être présentes sur les divers sites de vente en ligne.

4.2.1 Images

Chaque article est associé à plusieurs images (*fig.* 4.1). Il est donc garanti qu'au moins un même vêtement soit présent dans toutes ces images. Ce vêtement peut apparaître seul ou porté dans un exemple de tenue. Il peut y avoir différentes prises de vue et postures du mannequin (*ex.* de dos ou de face). Certaines images peuvent être des zooms sur des détails particuliers d'un vêtement.

FIGURE 4.1 – Exemples d'images pouvant être associées à un article.

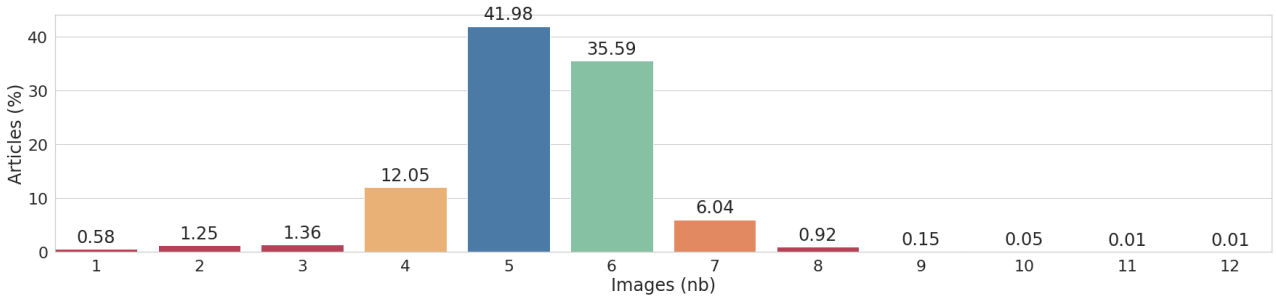


FIGURE 4.2 – Fréquences d'articles par leurs nombres d'images en pourcentage dans le corpus.

Le jeu de données comporte au total 336 195 images pour les 63 365 vêtements. Pour chaque article, nous disposons ainsi d'une à douze images (*fig.* 4.2). Nous avons en moyenne cinq images qui contiennent le même vêtement. Dans 54% des images figurent des vêtements pour individus de sexe masculin et dans 46% des vêtements pour individus de sexe féminin. Elles sont de qualité studio, issues d'un contexte professionnel, avec un éclairage et une scène maîtrisée. Les images ont des résolutions qui peuvent aller jusqu'à 3 600 pixels en hauteur et 2 400 en largeur.

4.2.2 Étiquettes

Afin de constituer un corpus d'apprentissage et d'expérimentation, nous devons associer des étiquettes à chaque image. Lors de l'inférence de nouvelles images, une méthode ne pourra alors prédire que les étiquettes préalablement définies et attribuées. Il est donc nécessaire de poser une taxonomie des étiquettes couvrant l'intégralité des caractéristiques et des valeurs que l'on souhaite être en mesure de retrouver. Une fois l'ensemble des étiquettes possibles et leur organisation définies, il nous reste à les attribuer à chaque article (*c.-à-d.* à chaque vêtement).

FIGURE 4.3 – Étiquettes (`jeans_jeans`, `Plain`) associées à l'article d'exemple de la figure 4.1.

Pour ce faire, nous nous appuyons sur les mots clefs de chaque article. Cette approche est classique lors de la constitution d'un corpus par collecte de données [HADI KIAPOUR et al., 2015 ; Tong XIAO et al., 2015]. En réalisant une cartographie de ces mots clefs, il est possible de constituer des sous ensembles de mots et d'établir des liens avec la taxonomie. Dans notre cas, ces liens entre groupes de mots et étiquettes ont été définis manuellement. Pour un article donné, ils nous permettent de lui associer différentes étiquettes selon l'occurrence des mots clefs (*fig.* 4.3). Nous présentons dans la figure 4.4 les étiquettes de types de vêtement et de motifs tissus associés aux articles. À cette étape, les étiquettes ne sont pas encore associées aux images. En effet, une image peut contenir plusieurs vêtements. Nous présentons alors dans la partie suivante comment nous les associons à un masque de vêtement dans l'image.

Pour le type de vêtement, Retviews filiale de Lectra nous a fourni une liste d'étiquettes exploitables. Ces étiquettes sont de granularité relativement fine. Nous en avons sélectionné un sous-ensemble de 33 directement attribuable aux données en utilisant les mots clefs. L'association des étiquettes aux articles fait ressortir certains types de vêtement comme très présents (*ex.* les t-shirts : `tshirts_tshirts`) et d'autres très faiblement fréquents (*ex.* les shorts de bain : `swimwear_trunks`) dans le corpus (*fig.* 4.4).

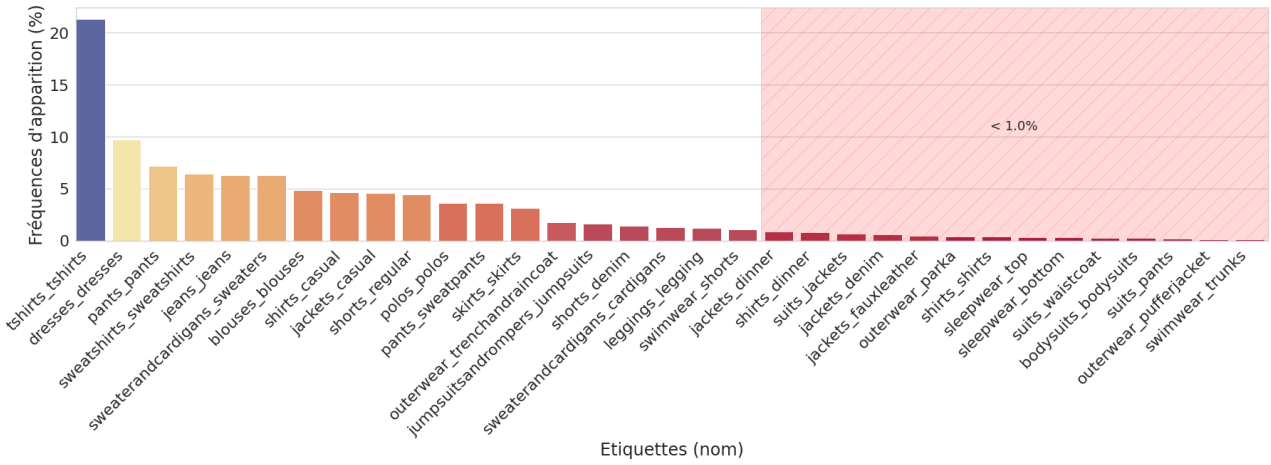


FIGURE 4.4 – Fréquences d'articles par types de vêtement de la taxonomie.

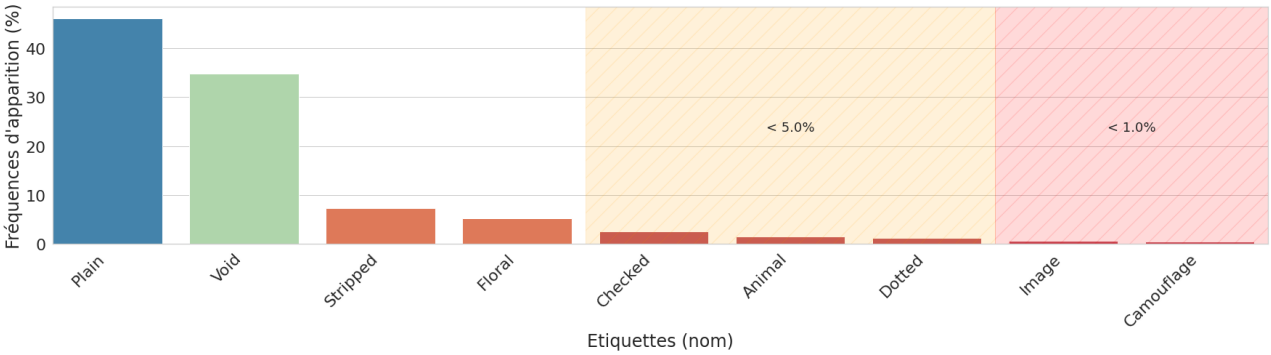


FIGURE 4.5 – Fréquences d'articles par motifs de tissu de la taxonomie.

Pour le motif tissu, différents échanges avec Lectra et Retviews nous ont permis de converger vers 8 étiquettes. 35% des articles (*fig.* 4.5) n'ont pas pu être associés à des étiquettes à partir des mots clefs. Ils sont nommés « void » dans la figure 4.5.

Globalement, le jeu de données est déséquilibré, ce qui peut rendre son analyse et son exploitation complexe.

4.2.3 Masques

Nous disposons d'images et d'étiquettes associées aux articles. Il est possible de directement associer ces étiquettes aux images. Cependant, nous aimerions précisément les localiser dans l'image. À partir des travaux présentés dans les chapitres 2 et 3, nous considérons le modèle de segmentation d'instances de vêtement YOLACT [BOLYA et al., 2019]. Ce modèle nous permet d'obtenir les masques de tous les vêtements contenus dans une image. Nous rappelons que le corpus contient, pour chaque image, la description par mot clefs d'un unique vêtement. Parmi tous les masques de vêtements prédits par le modèle de segmentation d'instances, il est donc

nécessaire de déterminer celui qui correspond à l'annotation contenue dans le corpus, afin d'y associer les étiquettes de type et de motif.

En plus du masque, le modèle YOLACT prédit une étiquette de la taxonomie posée par Deepfashion2. En effet, nous rappelons que nous avons entraîné ce modèle sur le corpus Deepfashion2. Nous souhaitons alors utiliser ces étiquettes prédites pour retrouver le masque qui correspond à nos étiquettes. Toutefois, il n'y a pas de bijection entre la taxonomie de Deepfashion2 de type de vêtement et la nôtre. De plus, aucune des deux taxonomies n'est injective dans l'autre. Cela signifie qu'il n'est pas possible de définir une règle directe d'association entre étiquettes. Par exemple, un masque `short sleeve top` Deepfashion2 peut correspondre à un vêtement : `shirts_casual`, `tshirts_tshirts` ou encore `polos_polos` de notre taxonomie. De même, une chemise `shirts_casual` peut correspondre à un masque : `long sleeve top` ou `short sleeve top` (*fig.* 4.6).

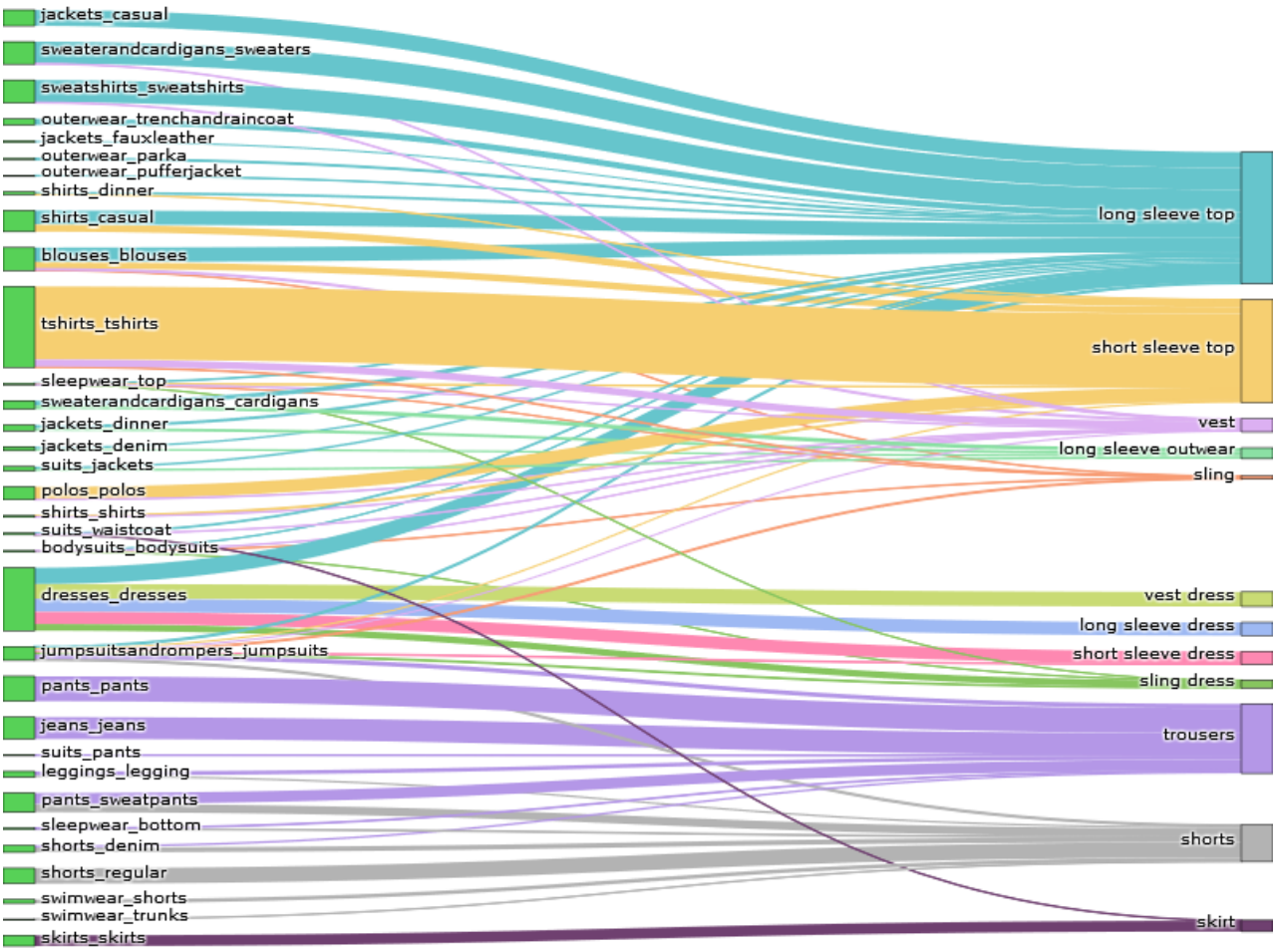


FIGURE 4.6 – Liens entre type de vêtement de notre taxonomie (en vert) et type de masque de Deepfashion2 (toutes les autres couleurs) après sélection d'un masque par image. La taille des liens est relative au nombre d'images et la couleur à celle des types de Deepfashion2.

Pour résoudre ce problème, une possibilité serait de parcourir toutes les images et de sélec-

tionner manuellement le meilleur masque prédit. Cependant, au vu de la quantité de données à traiter, cette option n'est pas envisageable. Nous proposons alors de s'appuyer sur les informations disponibles afin de constituer des règles d'associations.

Dans notre corpus, nous savons avec certitude que certaines images ne contiennent qu'un seul article. Nous pouvons donc tirer parti de ces images pour conserver tous les liens possibles entre type de masque et type d'article. En effet, nous considérons que tous les masques prédits pour ces images correspondent bien aux vêtements de l'article et par extension aux étiquettes de notre taxonomie. Si plusieurs masques du même type sont prédits pour la même image, nous conservons celui qui a le score de prédiction le plus élevé. Nous utilisons la fréquence des liens entre les deux taxonomies sur l'ensemble du corpus pour ordonner le choix des masques possibles. Nous obtenons ainsi des règles nous permettant de sélectionner un masque pour toutes les images contenant plusieurs vêtements. Pour chaque image, nous y associons le masque avec le type le plus haut dans l'ordre de fréquence des liens.

FIGURE 4.7 – Masques et étiquettes associées aux images de l'article d'exemple de la figure 4.1.

Après association des masques, notre corpus de données est composé d'images et pour chacune d'entre elles un unique masque annoté avec une étiquette de type et une étiquette de

motif de notre taxonomie (*fig.* 4.7). Pour nos expérimentations, les masques sont considérés comme des données à part entière du corpus et nous les considérons comme parfaits malgré le fait qu'ils soient le résultat de prédictions. Pour faciliter l'exploitation du corpus, les annotations sont dans le format proposé par T.-Y. Lin, Maire et al. en 2014 pour le corpus COCO (*Microsoft **C**ommon **O**bject in **CO**ntext*).

4.3 Classification du type de vêtement et du motif du tissu

Nous souhaitons à partir d'une image retrouver le type des vêtements contenus et le motif des tissus. Pour obtenir une méthode permettant de réaliser cette tâche, nous avons à notre disposition un corpus composé d'images. Chaque image du corpus est annotée avec une étiquette renseignant le type et une étiquette renseignant le motif. Ces étiquettes peuvent donc être utilisées pour entraîner des modèles.

En apprentissage, il s'agit d'un problème de classification multi-étiquettes et multi-classes, qui peut être résolu efficacement avec un réseau de neurones à couche de convolution (*CNN: **C**onvolutional **N**eural **N**etwork*). Il existe une multitude d'architectures dédiées à ce problème de classification (*ex.* LeNet [LeCun et al., 1998], AlexNet [Krizhevsky et al., 2017], EfficientNet [Tan et Le, 2019], *etc.*). La communauté propose constamment de nouvelles améliorations qui peuvent répondre à deux enjeux distincts. Le premier est l'amélioration des résultats, *c.-à-d.* la précision des prédictions, au détriment de la complexité des méthodes. Elles sont donc coûteuses en infrastructure et en temps. Le deuxième vise à diminuer la complexité des modèles tout en conservant de bons résultats. Le but est alors d'optimiser les architectures pour obtenir le maximum de précision pour chaque *Flop* (unité de mesure de la vitesse de calcul).

Il existe de nombreux bancs d'essai permettant de classer les modèles [Bianco et al., 2018 ; Reddi et al., 2020]. Nous nous appuyons sur ces travaux pour sélectionner une méthode. Notre but est alors d'obtenir un CNN extracteur de descripteurs entraîné sur nos données.

4.3.1 Classification à partir des masques

Notre objectif est de réaliser une classification de tous les vêtements contenus dans une image. Disposant d'un masque par image, nous souhaitons conditionner la prédiction à une zone de l'image. Le modèle étant entraîné avec des masques de tout type de vêtements, il pourra prédire les étiquettes de tous les vêtements par généralisation. Cela signifie que pour une même image, deux masques de vêtements différents produiront différentes prédictions.

Notre architecture (*fig.* 4.8) prend alors en entrée une image et un masque. Le masque est ajouté comme image binaire (*c.-à-d.* que 1 signifie que le pixel appartient au masque et

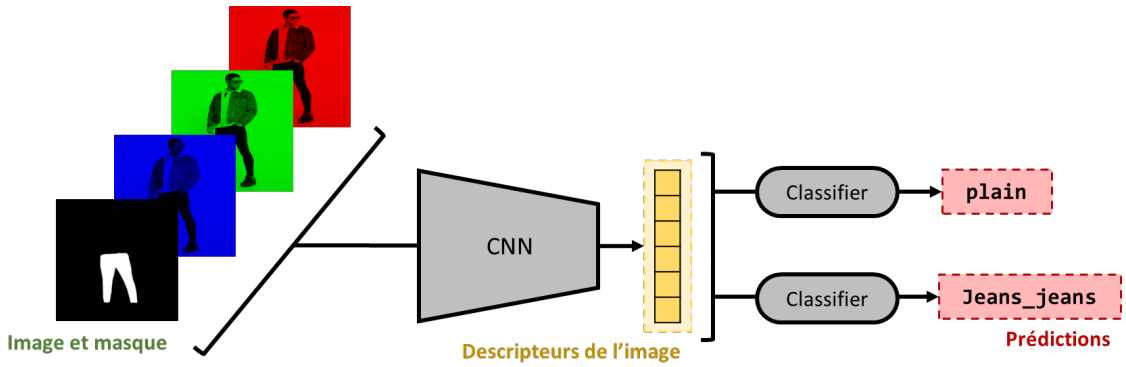


FIGURE 4.8 – Architecture de classification à partir d'une image associée à un masque.

0 sinon) en quatrième composante de l'image. Cette approche nous permet de conserver le contexte dans lequel se trouve le vêtement. En effet, appliquer le masque et ne fournir que les pixels du vêtement, supprimerait de l'information qui pourrait être utile au modèle. Ici le modèle est libre de sélectionner l'information et les descripteurs nécessaires à la prédiction. Le réseau prédit alors deux classes, une pour le type (*ex.* t-shirt), une pour le motif (*ex.* rayure). Ces deux tâches (*c.-à-d.* une classification du type, et une pour le motif) sont entraînées en parallèle en utilisant deux entropies croisées distinctes comme fonctions de coût.

Conditions expérimentales

Afin d'entraîner et d'évaluer des modèles, nous avons séparé le corpus de données en deux et les images, réduites à 224 pixels en hauteur et largeur. Nous souhaitons réserver au moins 10% des images pour chaque étiquette dans les données d'évaluation. Nous avons commencé par réserver des images pour chaque étiquette de motif tissu, puis complété, pour chaque étiquette de type de vêtement. Dues aux différentes fréquences des étiquettes dans le corpus, il reste finalement deux tiers des données pour l'entraînement.

Il existe différentes stratégies pour s'adapter à des données déséquilibrées. Par exemple, les données majoritaires peuvent être sous-échantillonnées ou les minoritaires dupliquées. Une alternative est de pondérer la fonction de coût en fonction de la fréquence des classes dans le jeu d'entraînement [Y. CUI et al., 2019]. Nous décidons de suivre cette approche. Nous conservons toutes les données dans leur proportion naturelle pour l'entraînement, mais pondérons la fonction de coût pour chaque image avec un poids inversement proportionnel à la fréquence de sa classe par rapport à celle de la classe dominante. Ainsi, plus la fréquence d'une classe est faible, plus sa pondération est grande, ce qui permet d'atténuer le déséquilibre dans les données d'entraînement. Notons enfin que certaines images du corpus n'ont qu'une annotation de type et pas d'étiquette de motif. Pour traiter ces images, un poids de 0 est considéré dans la fonction

de coût du motif.

Un réseau EfficientNet-B0 [Tan et Le, 2019] nous sert d'extracteur de descripteurs. Nous sélectionnons ce réseau, car il peut fournir une bonne précision des prédictions pour complexité en temps faible. Notre architecture devant être appelé pour chaque masque d'une image, nous privilégions un sous réseau efficace et performant (privilégiant la complexité en temps). L'architecture globale contient alors environ 4 millions de paramètres. Les résultats présentés par la suite ont été obtenus en entraînant notre architecture pendant 25 epochs avec des lots de 32 images. Une epoch prend environ 1 heure et 45 minutes et l'inférence d'une image associée à un masque environ 30 ms. La machine virtuelle de la plate-forme applicative en nuage de Microsoft Azure disposait d'une tesla P40.

Résultats

Sur nos données de validation, nous obtenons des *accuracy* ($\frac{TP+TN}{TP+FP+TN+FN}$) de 0.74 pour le type et 0.85 pour le motif. La figure 4.9 montre des prédictions du type de vêtement et la figure 4.10 du motif tissu, sur le jeu d'évaluation. Ces résultats sont en adéquation avec les résultats présentés dans Tan et Le en 2019, où les auteurs reportent une *accuracy* de 0.77 sur Imagenet [Deng et al., 2009] pour EfficientNet-B0. Sur nos données, on observe que plus la fréquence d'une étiquette diminue dans le jeu de données, plus le modèle produit des prédictions erronées.

Nous soulignons que pour le type de vêtements, les erreurs de classification impliquent principalement des étiquettes à la sémantique proche. On peut noter par exemple que presque un quart des prédictions de pantalon sont en fait des jeans ou des bas de survêtement, et un quart des prédictions de chemisier sont en réalité des t-shirts. Les erreurs peuvent alors être regroupées selon des mauvaises attributions de sous type de t-shirts, de chemises, de pantalons, de shorts, de pulls. La granularité de la taxonomie est sans doute trop fine pour les données, l'architecture et le temps d'entraînement consacré.

Pour le motif, nous observons une confusion entre motif `animal` et `floral`. Globalement, l'erreur principale est entre la classe `plain`, qui est sur-représentée dans le corpus, et toutes les autres étiquettes. Une autre explication vient du changement de résolution réalisé (réduction des images à la taille 224×224 pixels). De nombreux motifs fins ne sont ainsi plus visibles (*ex.* la robe de la figure 4.11 apparaît comme de couleur unie bleue alors qu'en réalité elle est à rayures). Les modèles sont alors entraînés avec des images qui contiennent des motifs unis, mais qui peuvent être annotées avec des étiquettes : `stripped`, `checked`, `dotted`. Les vêtements unis étant déjà surreprésentés dans le corpus, il est alors difficile d'obtenir un modèle qui discrimine les vêtements en s'appuyant sur les motifs.

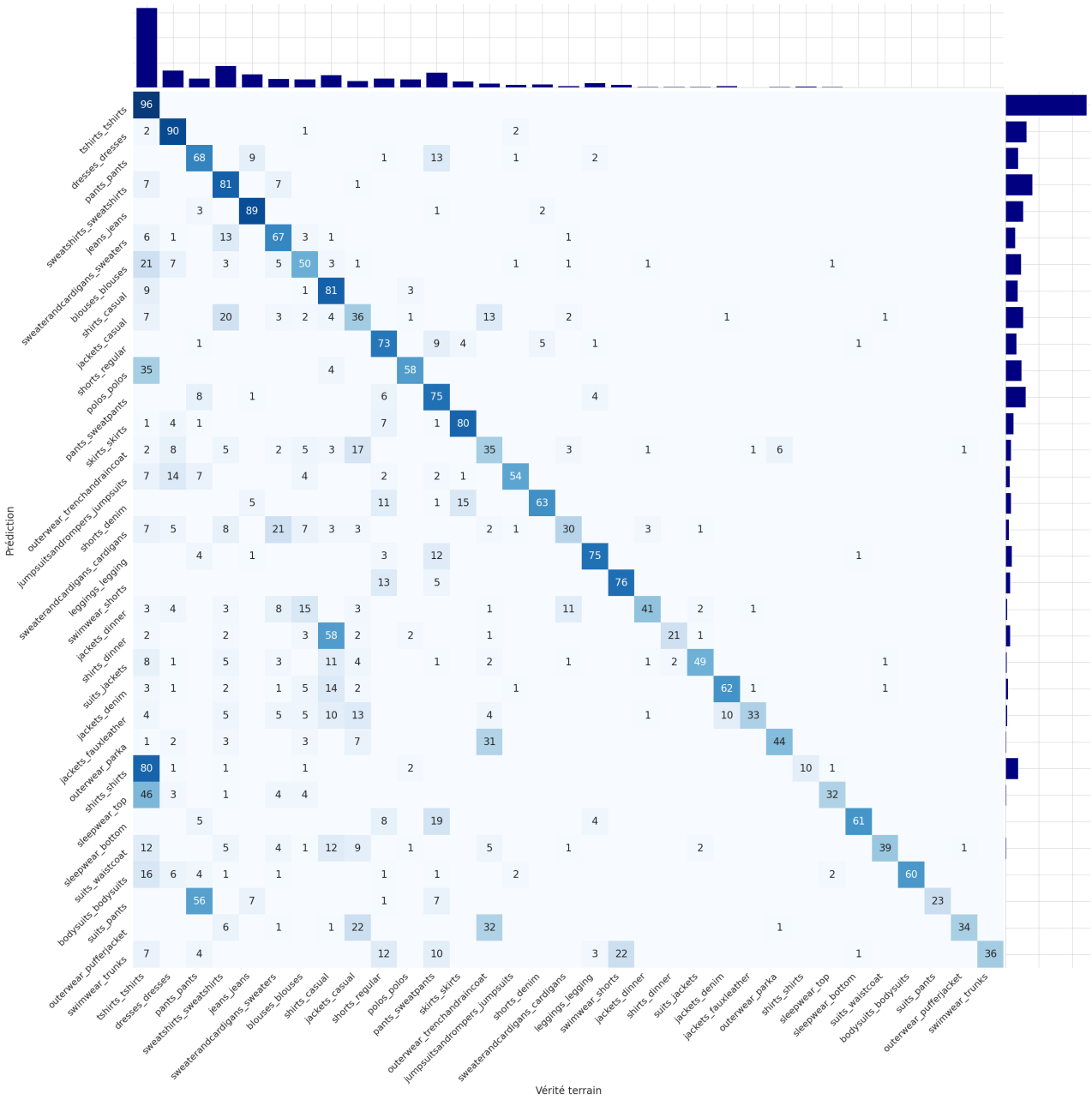


FIGURE 4.9 – Matrice de confusion des prédictions (lignes) par rapport à la vérité terrain (colonnes) du type de vêtement du masque. Les valeurs sont exprimées en pourcentage de prédiction arrondi à l'entier inférieur.

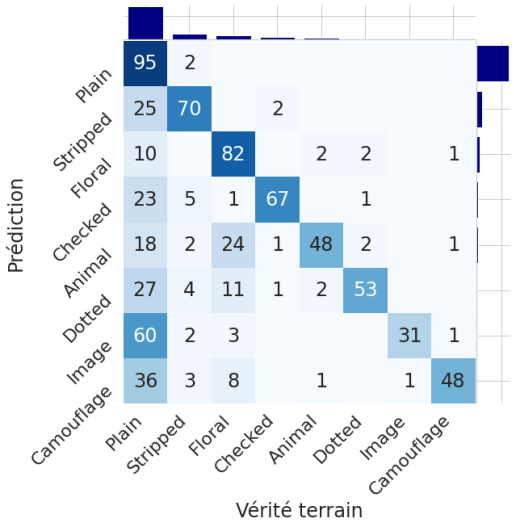


FIGURE 4.10 – Matrice de confusion des prédictions (lignes) par rapport à la vérité terrain (colonnes) du motif tissu du masque. Les valeurs sont exprimées en pourcentage de prédiction arrondi à l'entier inférieur.

4.3.2 Intégration de patchs dans l'architecture

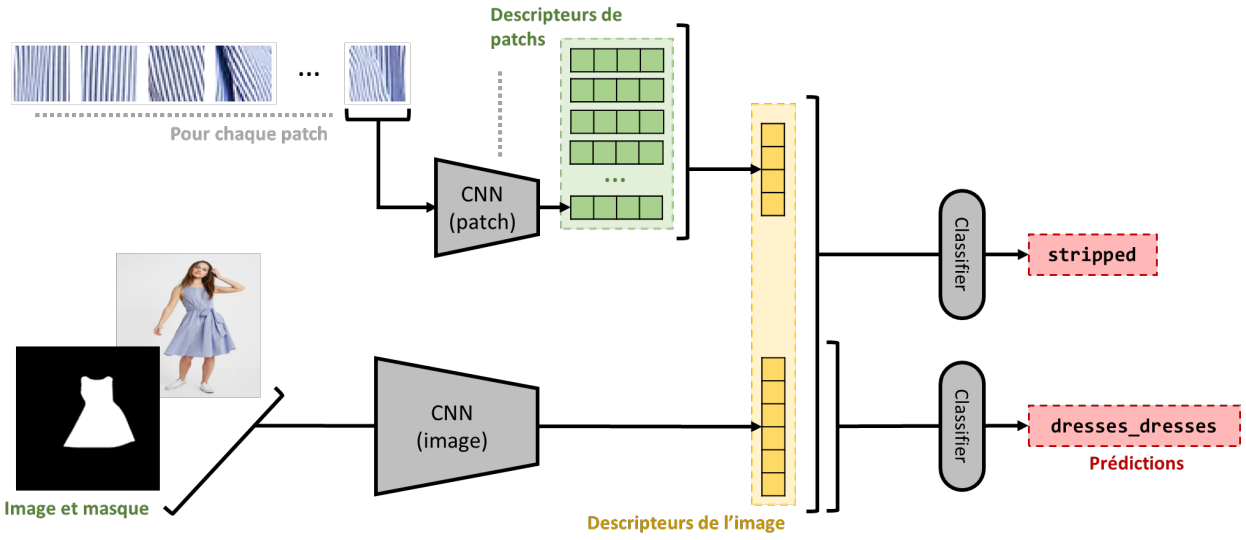


FIGURE 4.11 – Architecture de classification à partir d'une image associée à un masque et plusieurs patchs. Dans l'image d'exemple, le motif n'est plus visible à cause de la réduction de la résolution.

Afin d'améliorer les performances du modèle sur la caractérisation des motifs sur les tissus des vêtements, nous proposons une architecture qui prend en entrée plusieurs patchs à résolution native en plus de l'image et du masque (*fig.* 4.11). Nous conservons l'architecture précédente (*fig.* 4.8) mais ajoutons une branche qui extrait des descripteurs de patchs. Pour chaque patch pris sur l'image à résolution native, un CNN extrait un vecteur de descripteurs. Ces vecteurs

sont regroupés en faisant la moyenne par descripteur. Nous obtenons ainsi un unique vecteur regroupant tous les patchs qui est concaténé aux descripteurs de l'image.

Le classifieur de motifs peut alors les exploiter pour prédire une classe. Le classifieur de type, lui, ne s'appuie que sur les descripteurs de l'image. Cela permet à l'extracteur de descripteur de l'image de potentiellement être plus spécifique au type et les patchs spécifiques au motif.

Pour obtenir les patchs, nous utilisons le masque du vêtement. Pour chaque image, nous calculons une fenêtre circonscrite au masque. Puis dans cette fenêtre, nous tirons aléatoirement un patch. Si 80% des pixels du patch appartiennent au masque, le patch est conservé et il est rejeté sinon. Enfin, nous répétons l'opération jusqu'à obtenir le nombre de patchs souhaité.

Résultats généraux

Les résultats suivants ont été obtenus dans les mêmes conditions que ceux de la partie précédente. La branche d'extraction de descripteurs des patchs est constituée des cinq premiers blocs de l'architecture EfficientNet-B0 [Tan et Le, 2019]. Les deux extracteurs (image et patch) sont bien distincts. Cependant, les poids de la branche pour les patchs sont partagés entre tous les patchs. Cette branche rajoute environ 80 000 paramètres au modèle (environ 4 millions précédemment). Nous fournissons aux modèles 15 patchs de 64 pixels en hauteur et largeur par image. Ces modifications augmentent seulement d'environ 3 minutes le temps d'entraînement pour une epoch et n'a presque pas d'influence sur le temps d'inférence d'une image (31.4 images par seconde contre 31.9 images par seconde sans les patchs).

Avec cette méthode, les performances sur la classification du type de vêtement ne sont que faiblement augmentées (0.75 contre 0.74 en *accuracy*). Cependant, l'amélioration de la classification du motif est plus significative avec une *accuracy* de 0.92 contre 0.85 sans les patchs. L'amélioration est globale à toutes les étiquettes (*fig.* 4.12), elle est néanmoins plus nette sur certaines : `dotted`, `camouflage`, `animal`, `image`. Nous observons que le motif `dotted` avait plus tendance à disparaitre et nous supposons que : les gradients devenaient plus faibles pour le motif `camouflage`, le motif `animal` pouvait être confondu avec le motif `floral` avec la diminution de résolution de l'image. Ajouter des patchs à résolution native permet alors de corriger ces problèmes.

Toutefois, certaines erreurs sont toujours élevées, comme pour le motif `image`. Un vêtement ayant ce motif est globalement uni (`plain`) sauf certaines zones où se trouvent des images en impression. Ces vêtements peuvent donc induire le modèle en erreur. On note aussi que certains vêtements ont plusieurs motifs, n'en attribuer qu'un lors de l'annotation peut aussi introduire du bruit dans les données d'apprentissage.

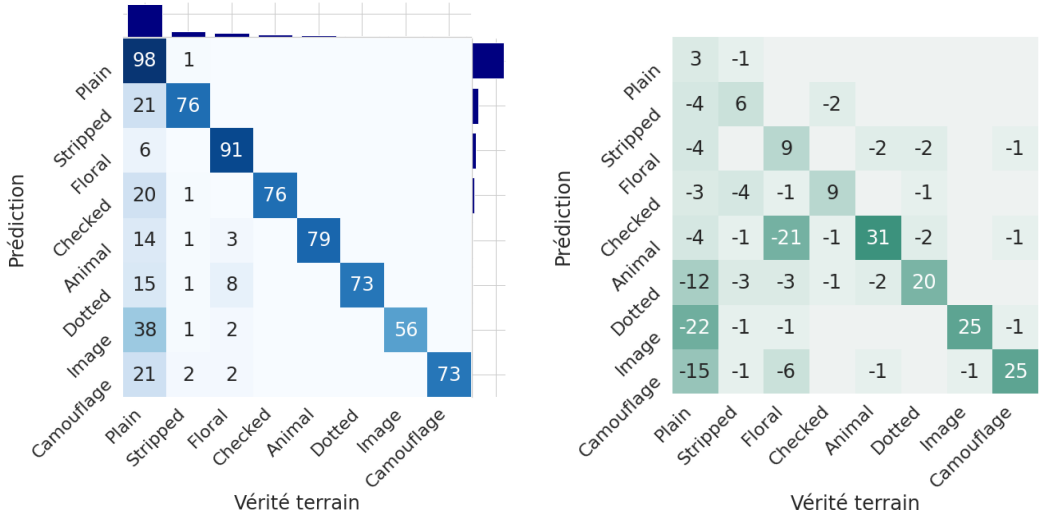


FIGURE 4.12 – Matrice de confusion (à gauche) des prédictions (lignes) du modèle exploitant des patchs par rapport à la vérité terrain (colonnes) du motif tissu du masque. Les valeurs sont exprimées en pourcentage de prédiction arrondi à l'entier inférieur. La matrice de droite représente l'écart avec la méthode sans patch. Plus le vert est sombre, plus l'amélioration est importante.

Résultats par article

Dans nos images, les vêtements sont capturés dans différentes conditions. Ils peuvent être portés ou non, de face, de dos, zoomés. Ces différentes variations peuvent rendre l'entraînement plus complexe et perturber les prédictions. Utiliser plusieurs images du même article lors de l'inférence pourrait permettre de limiter l'impact de ces variations.

Nous rappelons que certaines images de notre corpus contiennent le même vêtement, de 1 à 11 dans le jeu d'évaluation. Ces images peuvent alors être regroupées afin d'évaluer les modèles par article. Nous pouvons ainsi récupérer les prédictions par image et les agréger par article. En faisant la moyenne des scores de prédiction, nous obtenons une méthode simple de prédiction du type d'un vêtement et du motif tissu à partir de plusieurs images.

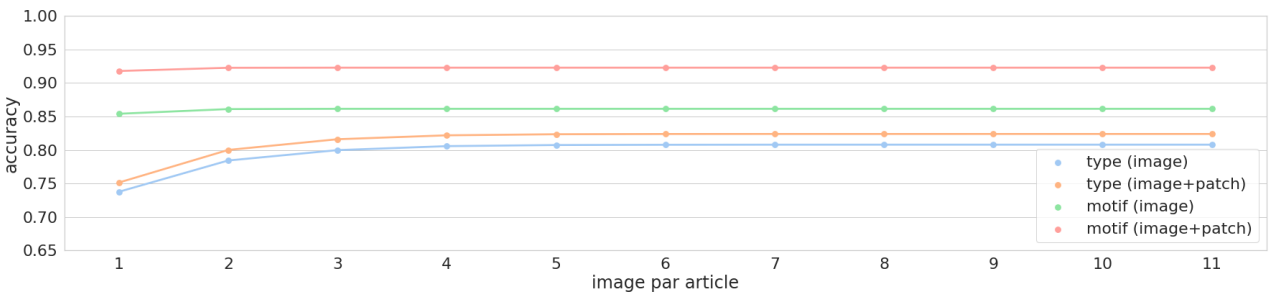


FIGURE 4.13 – Architecture de classification à partir d'une image associée à un masque et plusieurs patchs. Dans l'image d'exemple, le motif n'est plus visible dû à la résolution.

Afin d'évaluer l'influence du nombre d'images (avec un masque de vêtement par image) fourni lors de l'inférence pour caractériser un même article, nous constituons toutes les combinaisons possibles de n images, pour des n croissants à partir des ensemble d'images par articles. La figure 4.13 présente les résultats. Les performances de la classification du type de vêtements augmentent rapidement pour chaque image supplémentaire, puis se stabilisent. Cela nous informe que la prédiction de l'étiquette s'améliore lorsque l'on fournit plusieurs images, mais qu'il n'est pas nécessaire d'utiliser un nombre trop important d'images. En effet, à partir de quatre images, les résultats ne semblent plus s'améliorer. Regrouper les prédictions de quatre images par articles est alors une bonne stratégie pour prédire le type du vêtement.

Utiliser plusieurs images disponibles pour un même article améliore les prédictions de toutes les étiquettes de type (figure 4.14 et figure 4.15). Malgré cette amélioration, les prédictions de certaines étiquettes restent mauvaises (*ex.* `sleepwear_top`,`suits_pants`,*etc.*). Ces erreurs semblent principalement être liées aux plus faibles nombres d'exemples de certaines étiquettes (*ex.* 623 images pour l'entraînement et 364 pour la validation parmi les 336 195 images du corpus pour l'étiquette `sleepwear_top`) ou à des confusions entre étiquettes à la sémantique proche (*ex.* `suits_pants` avec `pants_pants`).

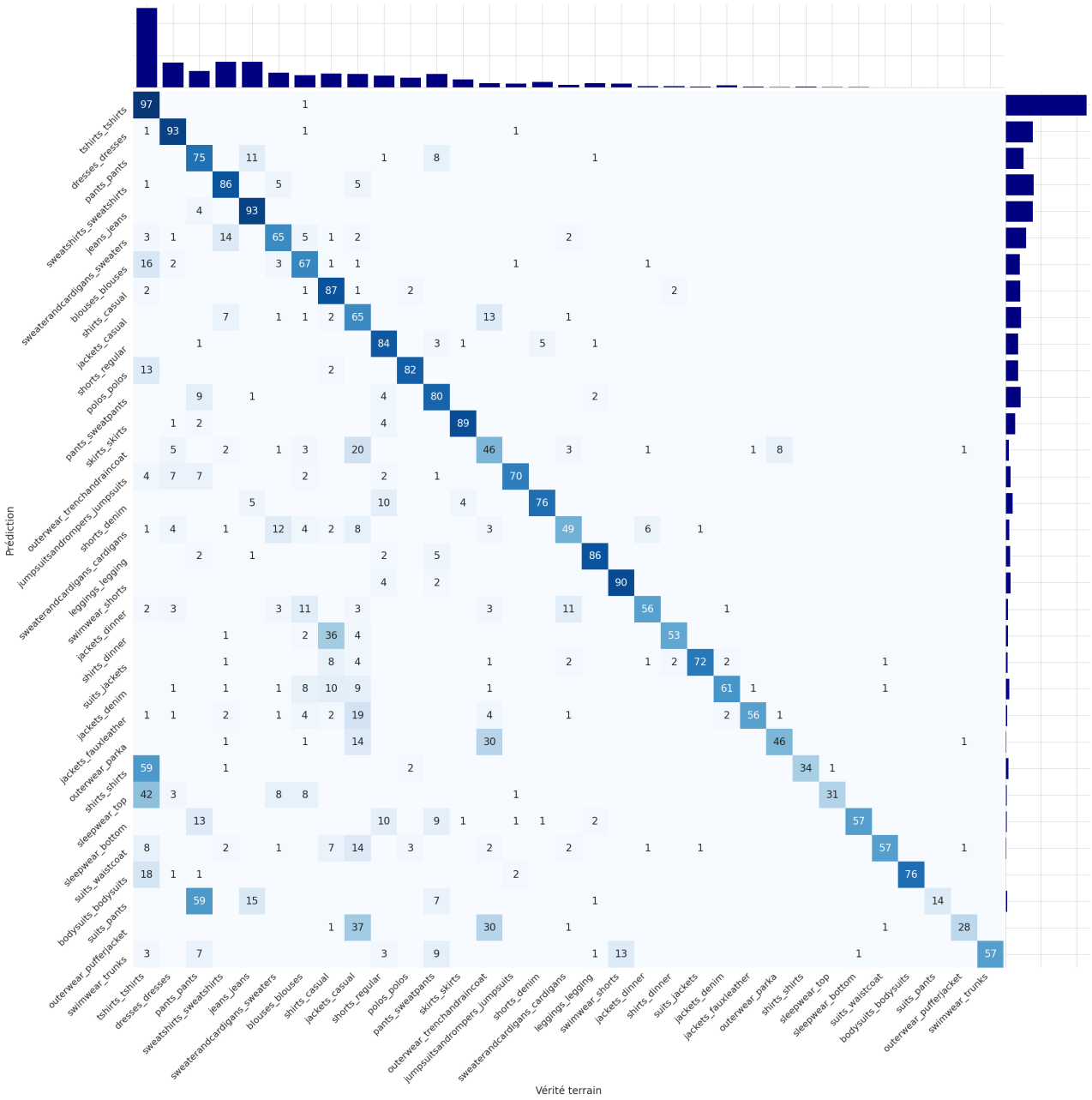


FIGURE 4.14 – Matrice de confusion des prédictions (lignes) agrégé par article du modèle exploitant des patchs par rapport à la vérité terrain (colonnes) du type de vêtement du masque. Les valeurs sont exprimées en pourcentage de prédiction arrondi à l'entier inférieur.

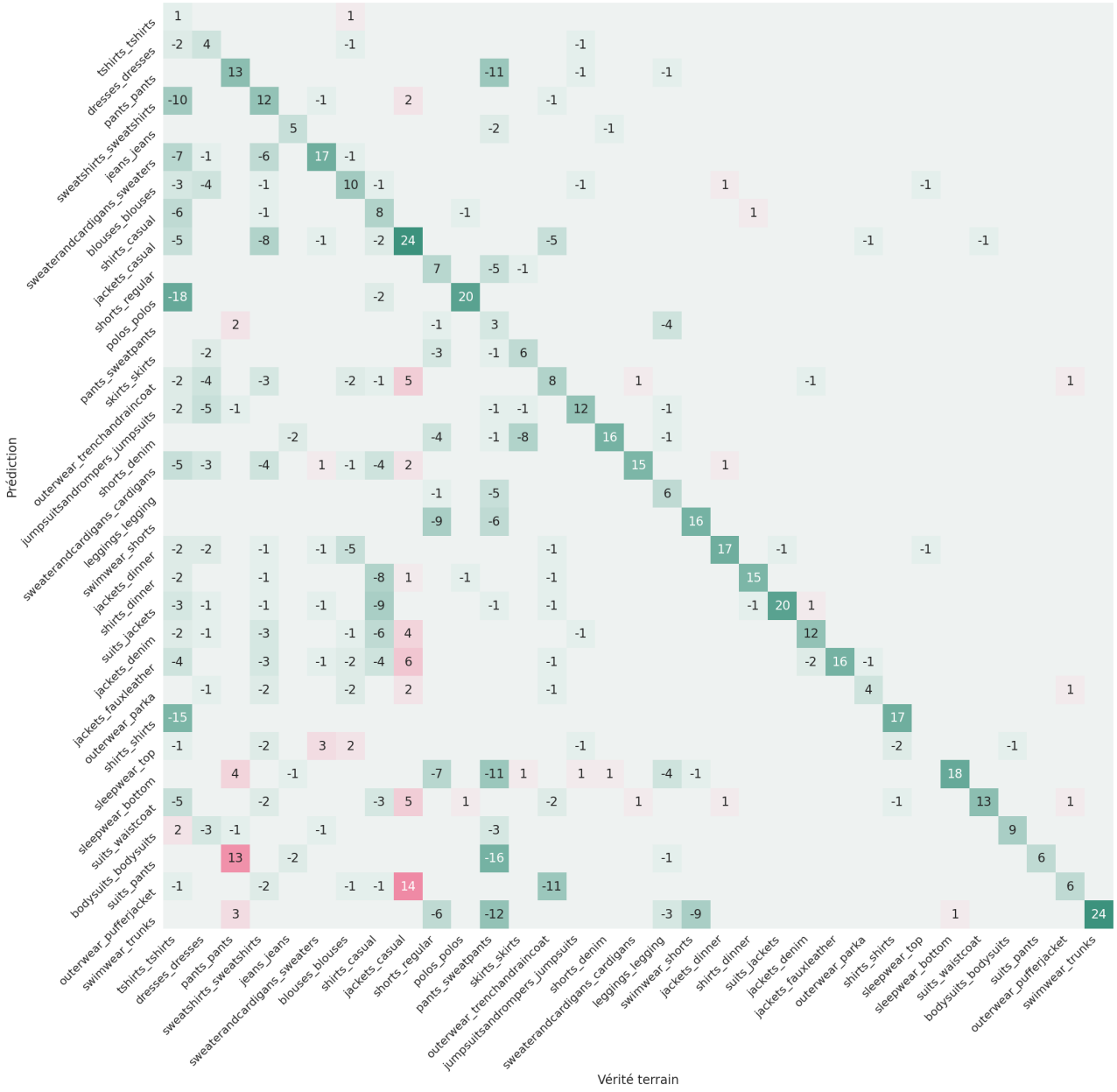


FIGURE 4.15 – Matrice de l'écart des prédictions par article du type de vêtement par rapport aux prédictions par images. Les cellules sont rouges pour les dégradations et vertes pour les améliorations. Plus le vert est sombre, plus l'amélioration est importante et plus le rouge est sombre, plus la dégradation est importante.

4.4 Extraction de la couleur dominante

Pour caractériser les couleurs des vêtements, nous avons deux objectifs. Le premier consiste à extraire la couleur dominante (*c.-à-d.* la couleur principale) d'un vêtement. Le deuxième est d'être capable de nommer cette couleur. Pour nommer les couleurs, nous utilisons une nomenclature. Nous considérons cette nomenclature comme une taxonomie qui nous fournit des références couleurs utilisées comme mots clefs intelligibles par un utilisateur.

La perception des couleurs varie en effet selon les individus. Afin de lever les ambiguïtés d'interprétation et d'associer des mots clefs à certaines couleurs, il est essentiel de pouvoir les nommer. Nommer les couleurs permet aussi d'obtenir une partition simplifiée de l'espace des couleurs [Regier et al., 2007]. Cependant, ce choix de nomenclature est intimement lié aux langues et aux cultures [BERLIN, 1969 ; Regier et al., 2007]. L'uniformisation des noms est une tâche non triviale qui facilite la cognition et l'utilisation de couleurs [Heer et Stone, 2012]. Dans un contexte commercial et industriel, les couleurs doivent être clairement identifiables pour un public international. Il existe différentes nomenclatures associées à des nuanciers (*c.-à-d.* des taxonomies souvent appelées bibliothèques de couleurs) qui permettent de s'abstraire des variations micro au niveau des individus et macro au niveau des sociétés. On peut citer par exemple le nuancier de Munsell, le *Colour Index International* (pour les colorants et teintures), les dictionnaires Pantone, et le *Natural Color System*.

Nommer les couleurs dominantes permet d'ajouter une information à la caractérisation des images et plus particulièrement des vêtements. Cette information est utile par exemple pour avoir des filtres supplémentaires dans un moteur de recherche et pour l'analyse à grande échelle des tendances et des styles vestimentaires [Al-Halah et Grauman, 2020a ; Al-Halah, Stiefelhagen et al., 2017 ; Hsiao et Grauman, 2021].

L'extraction des couleurs dominantes peut se faire par apprentissage supervisé : par régression [Ramé et al., 2022] ou par classification avec une nomenclature [Yazici et al., 2018]. Ces approches nécessitent une vérité terrain qui peut par exemple être une annotation des couleurs des vêtements réalisée par des experts métiers [Ramé et al., 2022]. Il est possible de se passer de cette annotation en utilisant des méthodes non supervisées. Les couleurs dominantes d'une image peuvent alors être obtenues sous la forme d'une palette ordonnée de couleurs, par exemple par découpage successif (*ex.* coupure médiane [Heckbert, 1982]) ou par partitionnement (*ex.* k-moyenne [Delon et al., 2005]) de l'espace des couleurs. L'obtention d'une palette de couleur est une étape importante de la quantification des couleurs d'une image. Heckbert en 1982 pose les étapes suivantes nécessaires à la quantification : 1) obtention de statistiques des couleurs dans l'image, 2) choisir une cartographie des couleurs, 3) associer les couleurs de l'image à la cartographie par voisin le plus proche, 4) quantifier et redessiner l'image.

Dans la suite, nous nous inspirons de ce cadre. Nous souhaitons obtenir la couleur principale d'un vêtement sous la forme d'un mot clef. Tout d'abord, nous projetons les pixels dans un espace de couleurs adapté à notre cas d'usage. Nous listons les espaces de couleurs utilisés pour les expérimentations dans la section 4.4.1. La première étape, que nous présentons dans la section 4.4.2, consiste alors à obtenir un histogramme local des couleurs du vêtement. Puis en deuxième étape, dans la section 4.4.3, nous étudions comment associer des couleurs à une nomenclature donnée. Enfin, dans la section 4.4.4, nous proposons une évaluation de la méthode sur nos données et détaillons notre sélection des meilleurs paramètres. Nous rappelons que cette approche empirique a été proposée pour répondre à un besoin industriel.

4.4.1 Projection dans un espace de couleurs

La couleur est une perception visuelle d'un phénomène physique (*c.-à-d.* la lumière). Dû à l'œil et au cortex visuel humain, la perception du spectre lumineux peut être émulée avec seulement trois longueurs d'onde. Concrètement, cela signifie qu'il est possible d'obtenir toutes les couleurs perceptibles à partir de la composition de trois couleurs : rouge, vert, bleu. Une image naturelle peut donc être obtenue par capture (*ex.* avec un appareil de photographie) en discrétisant spatialement la scène en cellule, et pour chaque cellule (*c.-à-d.* pixel) mesurer l'intensité lumineuse de chacune des trois couleurs. La résolution de l'image combinée aux valeurs des trois couleurs pour chaque pixel permet de simuler la perception d'objet par la vision dans des images.

L'ensemble des valeurs de couleurs possibles des pixels est contenu dans un espace en trois dimensions nommé espace de couleurs RGB (***R**ed **G**reen **B**lue*). Ces couleurs peuvent être représentées dans d'autres espaces adaptés à certains cas d'usage. La distance euclidienne utilisée dans l'espace de couleurs $l^\star a^\star b^\star$ (***l**uminance **a**lpha **b**eta*) [Robertson et al., 1977] permet par exemple de mieux prendre en compte l'écart entre deux couleurs en termes de perception humaine. L'espace de couleurs HSV (***H**ue **S**aturation **V**alue*) permet quant à lui de distinguer la teinte de la couleur en première composante.

Nous proposons alors d'étudier quel est l'espace de couleur le plus adapté à notre problème parmi l'espace de couleurs RGB (***R**ed **G**reen **B**lue*), l'espace de couleurs $l^\star a^\star b^\star$ (***l**uminance **a**lpha **b**eta*) et l'espace de couleurs HSV (***H**ue **S**aturation **V**alue*) (*fig.* 4.16). Les couleurs dans l'espace RGB ont des valeurs entières positives inférieures à 255. Dans l'espace $l^\star a^\star b^\star$, les couleurs ont une valeur de luminance réelle positive inférieure à 100 et deux composantes de chrominance α et β dont les valeurs sont comprises entre -100 et 100. Enfin, dans l'espace HSV, les couleurs peuvent avoir une valeur de teinte réelle positive inférieure à 360, une saturation et une valeur inférieures à 100.

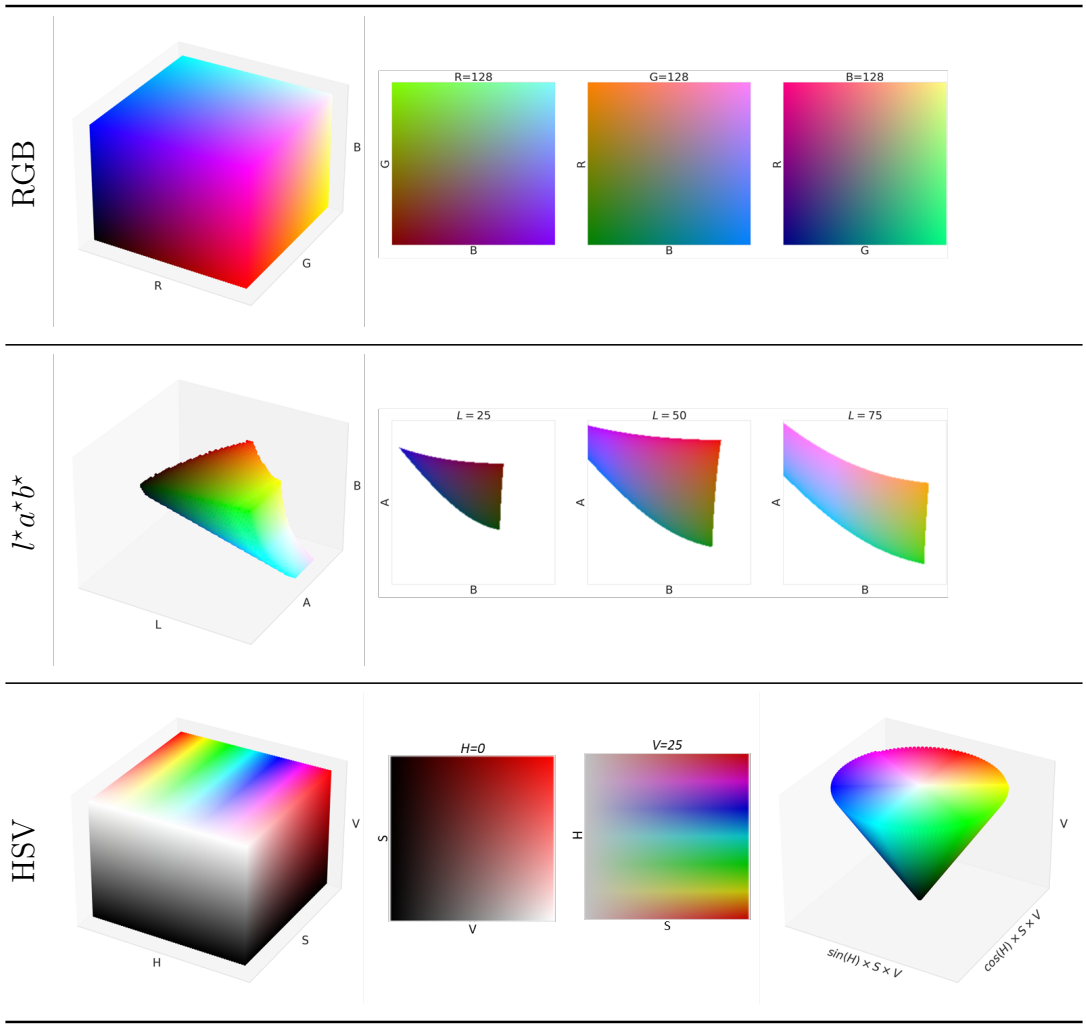


FIGURE 4.16 – Visualisation des espaces de couleurs en 3D. Pour chaque espace, les couleurs pour une composante fixe sont sous la forme d'une image. Pour HSV, nous faisons aussi figurer la projection dans le cône.

4.4.2 Mode principal d'histogrammes locaux

Afin de déterminer la couleur principale d'un vêtement, nous entendons sélectionner la couleur la plus fréquente en termes de perception humaine. Pour cela, on peut représenter les fréquences de couleurs par un histogramme 3D discret de l'espace de couleur, où chaque cellule de l'histogramme est communément appelée bin de couleur.

Étudier cet histogramme sur l'image entière ne nous permet pas d'obtenir une information pertinente. En effet, le mode principal de l'histogramme global de l'image, *c.-à-d.* le bin de couleur le plus représenté dans l'image, n'est pas contraint localement aux vêtements (couleur du fond dans la figure 4.17). Travailler avec des histogrammes locaux [ALAMDAR et KEYVANPOUR, 2011] peut donc s'avérer utile. Dans notre cas, nous disposons de la segmentation d'instances de vêtements. S'appuyer sur ces masques de vêtements nous permet donc d'obtenir les fréquences

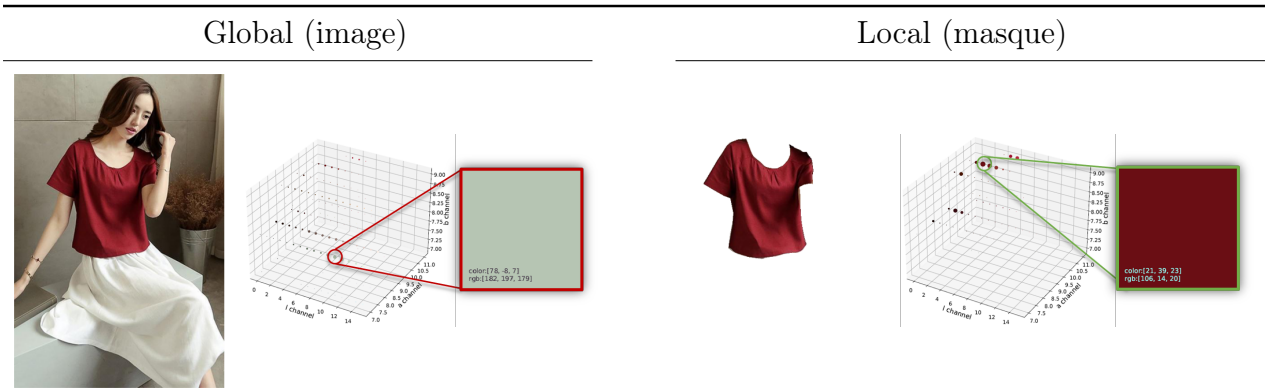


FIGURE 4.17 – Exemple de mode principal de l'histogramme pour l'image et localement au masque.

des couleurs spécifiques aux vêtements [AL-RAWI et BEEL, 2021]. Le mode principal de l'histogramme de couleur sur le masque est donc garanti d'être une couleur contenue dans le vêtement (*fig.* 4.17).

Toutefois, utiliser la couleur majoritaire ne garantit pas de fournir la couleur principale perceptuellement. En effet, les couleurs d'un vêtement peuvent être plus ou moins diffuses et précises en fonction de la présence de motifs sur le vêtement, ainsi que de l'éclairage de la scène. Cette problématique peut être adressée en considérant une discrétisation plus grossière des dimensions de l'espace de couleurs en réalisant un regroupement plus grand des couleurs. Ce regroupement en bin de grandes tailles gomme les faibles nuances de couleurs et permet de mieux faire ressortir la couleur principale.

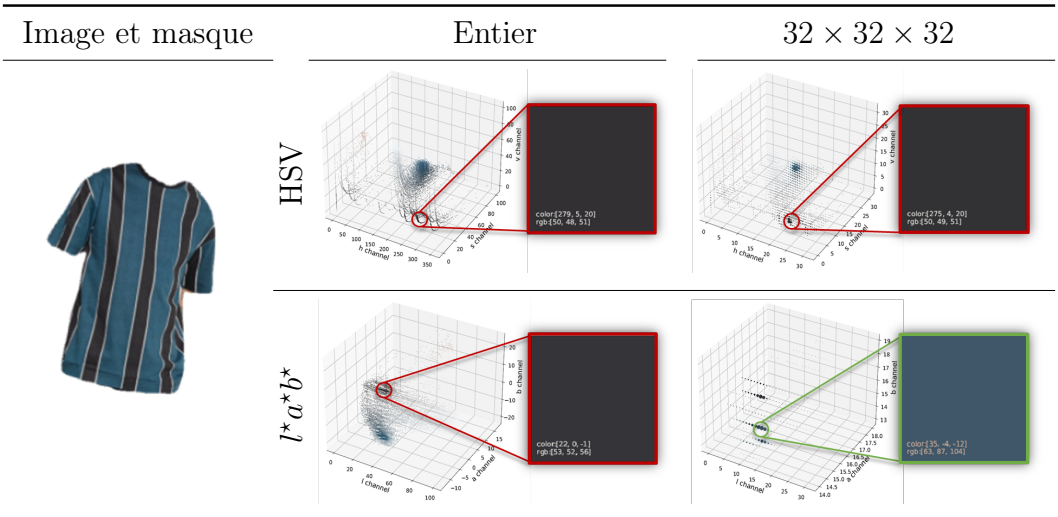


FIGURE 4.18 – Exemple de l'impact du choix de l'espace de couleurs (HSV en haut et $l^\star a^\star b^\star$ en bas) et de la discrétisation (en nombre entier à gauche et en 32 valeurs par composantes à droite) dans le bin mode.

Au sein d'un bin, il est crucial qu'une faible distance entre couleurs exprime une proximité en terme de perception humaine des couleurs. Les pixels peuvent alors être représentés dans des espaces de couleurs aux différentes propriétés (section 4.16 précédente). Une discrétisation uniforme dans ces différents espaces implique des répartitions différentes de couleurs au sein des bins respectifs (axe vertical de la figure 4.18).

Pour chaque espace, nous considérons des discrétisations uniformes de chaque dimension. Plus la discrétisation est grossière, plus les bins contiennent des couleurs différentes (axe horizontal de la figure 4.18). Il existe globalement un compromis entre un regroupement assez large pour faire ressortir une couleur dominante et assez fin pour conserver la cohérence des couleurs internes aux bins.

La méthode que nous proposons repose sur le choix d'un espace de couleur et d'un niveau de discrétisation de cet espace. La figure 4.18 montre l'impact de ces choix. Le t-shirt semble avoir comme couleur principale le bleu. Les nuances de bleu semblent plus dispersées et les couleurs sombres plus centrées. La discrétisation grossière permet alors de les regrouper. Néanmoins, le regroupement semble plus cohérent en $l^\star a^\star b^\star$. Il s'agit d'un exemple dans lequel l'espace et la discrétisation sont des paramètres de la méthode et leur sélection doit être confortée après évaluation.

4.4.3 Espace de couleur pour Pantone

Nous présentons ici notre méthode d'association de couleurs à des références dans une nomenclature donnée. Nous utiliserons la librairie Pantone « *Solid Uncoated V4* » à titre d'exemple. Cette bibliothèque contient 2 161 références de couleurs. Ces références sont des couleurs précises associées à des dénominations (*fig.* 4.21). Notre objectif est alors d'obtenir le partitionnement d'un espace de couleur fournissant une palette de couleurs à partir des dénominations Pantone. Nous souhaitons donc sélectionner l'espace de couleurs et la discrétisation la plus adaptée à la constitution de cette palette.

Ne souhaitant extraire que la couleur principale des vêtements, le mode principal de l'histogramme de couleur 3D nous suffit. Il ne nous est donc pas utile d'obtenir une partition de l'espace des couleurs spécifique à chaque vêtement. Nous pouvons alors simplifier l'attribution des noms de couleurs aux valeurs des pixels. Chaque pixel appartenant à un bin et la partition produite par la discrétisation étant globale et uniforme, nous pouvons directement attribuer un nom parmi la nomenclature à chaque bin. De plus, nous pouvons pré-calculer une table de correspondance globale associant ces intervalles aux noms de couleurs. Dans la suite, nous montrons l'impact du choix de l'espace et du niveau de discrétisation sur une nomenclature et

sur la création de la table de correspondance.

Afin d'associer une couleur pantone à chaque bin, nous devons être en mesure de calculer une distance entre couleurs. Pour l'espace RGB nous utilisons la distance euclidienne. La distance euclidienne sur des couleurs de l'espace $l^\star a^\star b^\star$ est une norme nommée *CIE 1976* posée par la commission internationale sur l'éclairage (CIE). Nous considérerons également la norme *CIE 2000* [Sharma et al., 2005] qui introduit des poids dans le calcul de la distance afin de mieux exploiter la similarité des couleurs dans l'espace $l^\star a^\star b^\star$. Cette distance peut de plus être paramétrée spécifiquement pour des textiles [H. Liu et al., 2013 ; Melgosa et al., 2004]. Enfin, pour l'espace HSV, nous projetons les couleurs sur le cône (*fig.* 4.16) avant d'utiliser la distance euclidienne.

Afin de sélectionner les meilleurs paramètres, nous calculons différentes tables de correspondance. Nous obtenons ainsi des tables de quatre tailles différentes, correspondant respectivement à une discrétisation des espaces de couleur en 128, 64, 32 et 16 valeurs sur chaque dimension. Nous notons qu'il pourrait être judicieux d'avoir un nombre de valeurs différent pour chaque composante (*c.-à-d.* dimension). En effet, les dimensions de chaque espace de couleurs peuvent porter plus ou moins d'information (*ex.* la teinte pour HSV).

Pour chaque niveau de discrétisation, nous considérons 4 différents espaces et distances associées : distance euclidienne dans l'espace RGB, *CIE 1976* et *CIE 2000* dans l'espace $l^\star a^\star b^\star$, et enfin distance euclidienne dans le cône HSV. Nous utilisons ces distances pour calculer la proximité de chaque pantone au centre de chaque bin. Le pantone le plus proche d'un bin lui est alors associé dans la table de correspondance.

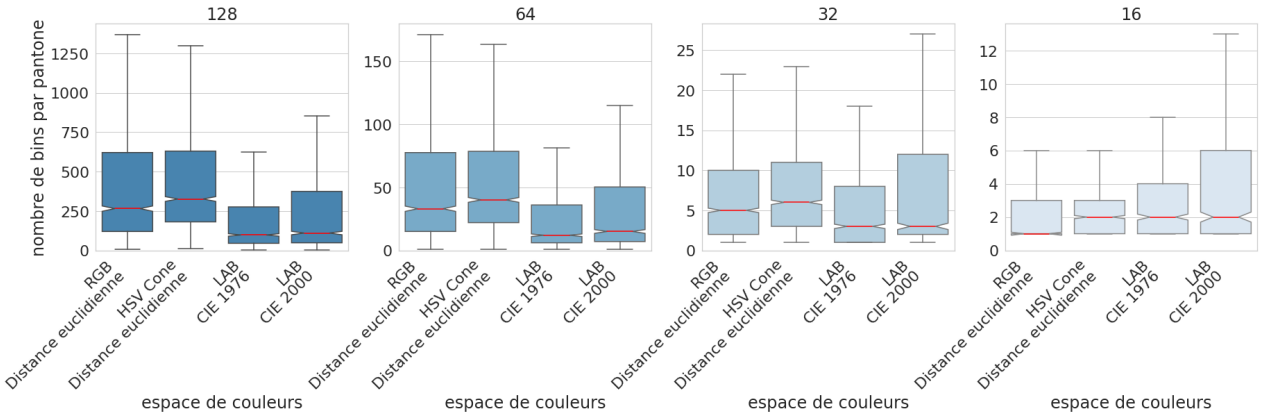


Figure 4.19 – Distribution du nombre de bin associé par pantone selon la discrétisation (128, 64, 32, 16) du bleu plus au moins foncé, et selon l'espace de couleurs et la distance.

La figure 4.19 présente la répartition des bins par pantone selon : l'espace de couleurs, la distance et la discrétisation. Ces résultats montrent alors que les tables de correspondance sont plus homogènes (*c.-à-d.* une meilleure répartition des pantones avec un nombre de bins plus

uniformes par pantone) dans l'espace $l^\star a^\star b^\star$. En effet, la médiane et le maximum sont plus faibles que pour les autres espaces de couleurs. Cependant, avec la distance *CIE 2000*, cette tendance s'inverse à partir d'une discrétisation en 32 valeurs par dimension. L'espace $l^\star a^\star b^\star$ avec une discrétisation en 128 ou 64 valeurs semble donc plus adapté à l'obtention de la table de correspondance.

La figure 4.20 précise alors le nombre de pantones non associés dans les tables de correspondances en fonction de la discrétisation. Nous observons que lorsque le nombre de valeurs de la discrétisation diminue, moins de références pantone sont associées dans les tables de correspondance. Pour des références proches en termes de distance, une discrétisation grossière ne permet pas de les distinguer. Plusieurs références devraient alors être associées aux mêmes bins. Comme nous ne conservons qu'une couleur pantone par bin, certaines couleurs de la nomenclature sont exclues.

Pour obtenir une table de correspondance (*c.-à-d.* en quelque sorte une palette de couleurs pantone) l'espace $l^\star a^\star b^\star$ discrétisé en 128 ou 64 valeurs et en utilisant semble les plus adaptés. Nous souhaitons alors sélectionner précisément les paramètres et conforter ce choix avec l'évaluation proposé dans la section suivante.

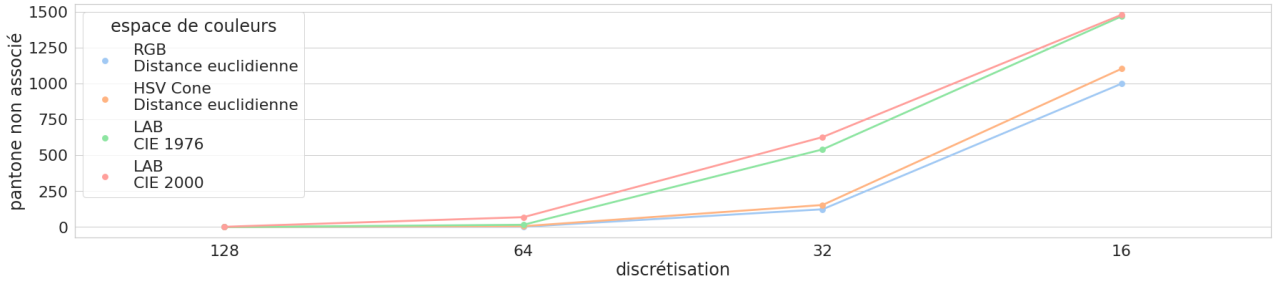


FIGURE 4.20 – Nombre de pantones non associés dans la table de correspondance en fonction de la discrétisation pour les différents espaces de couleurs et distances.

FIGURE 4.21 – Exemples de références pantone non associées (ligne du haut) dans la table de correspondance spécifique à l'espace $l^\star a^\star b^\star$ discrétisé en 64 valeurs et la distance *CIE 2000*. La ligne du bas montre les références pantone les plus proches et associées dans la table de correspondance.

En observant des exemples de références pantone non associées dans une table de corres-

pondance ($l^\star a^\star b^\star$, *CIE 2000*, discrétisé en 64 valeurs de la figure 4.21) nous nous apercevons qu'elles peuvent être très proches perceptuellement de références accessibles dans la table. Cette proximité est telle qu'il n'est pas souhaitable de forcer la sélection de paramètres de la méthode qui garantisse que toutes les références soient contenues dans la table de correspondance. De plus, être en mesure de nommer les couleurs avec la totalité des références pantone n'est pas réaliste au regard des différentes fluctuations introduites par le contexte de capture des images. Ce nombre de références non associées doit toutefois être maîtrisé.

4.4.4 Évaluation de la justesse à travers le lien article-image

Afin d'évaluer la méthode d'extraction, nous avons à notre disposition des images et des masques de vêtements. Toutefois, nous n'avons pas d'annotations de la couleur, que ce soit en termes de valeur précise ou de référence dans une nomenclature. Nous souhaitons néanmoins évaluer la qualité de la référence de couleur de vêtements obtenue par notre méthode.

Nous rappelons que pour chaque article (*c.-à-d.* vêtement) nous disposons de plusieurs images et masques. Au sein de ces images, la couleur extraite doit être la même. Nous proposons alors d'évaluer la consistance des extractions au sein des articles. Cette évaluation ne nous permet pas de déterminer si la référence associée est la bonne, mais elle informe sur l'homogénéité des extractions.

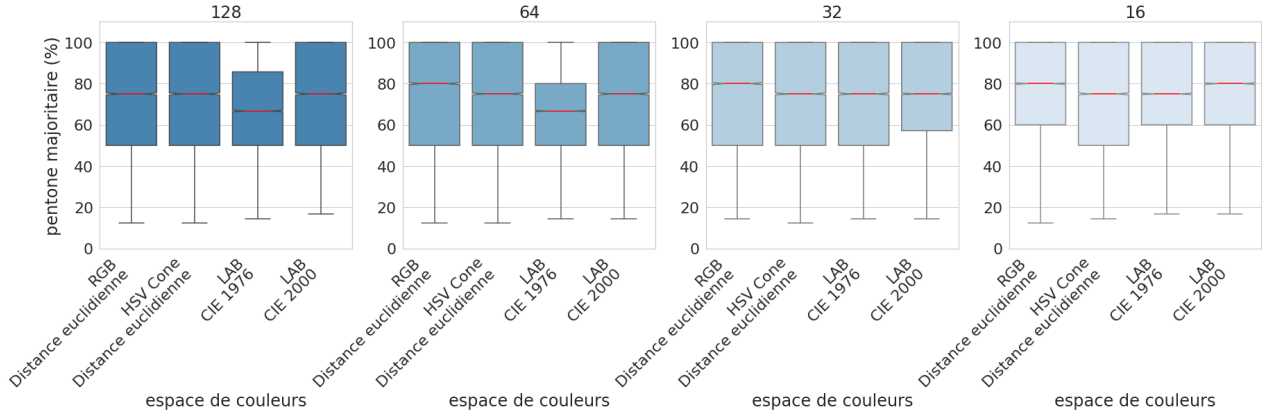


FIGURE 4.22 – Distributions des pourcentages des références pantone les plus attribuées aux images par articles selon l'espace de couleurs, la distance et la discrétisation.

En agrégeant les pantones attribués aux images par article, nous obtenons les résultats présentés dans la figure 4.22. La figure montre le pourcentage du pantone le plus attribué par article. Globalement, la moitié des articles ont au moins quatre cinquième de leurs images qui se sont vus attribuer la même référence couleur pantone. L'espace $l^\star a^\star b^\star$ avec la distance *CIE 1976* semble cependant fournir des attributions moins homogènes. À l'inverse, l'espace RGB

semble fournir de bons résultats. Cela montre que cette information seule n'est pas un bon indicateur pour sélectionner l'espace de projection des couleurs. En effet, cette évaluation ne nous garantit pas que le pantone attribué soit le bon.

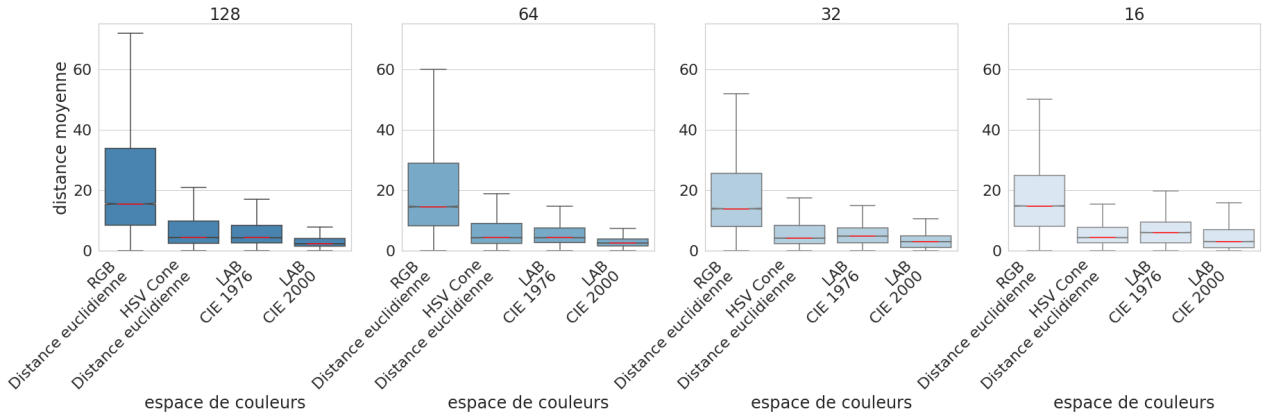


FIGURE 4.23 – Distributions des distances moyennes entre les couleurs pantone des images par articles selon l'espace de couleurs, la distance et la discrétisation.

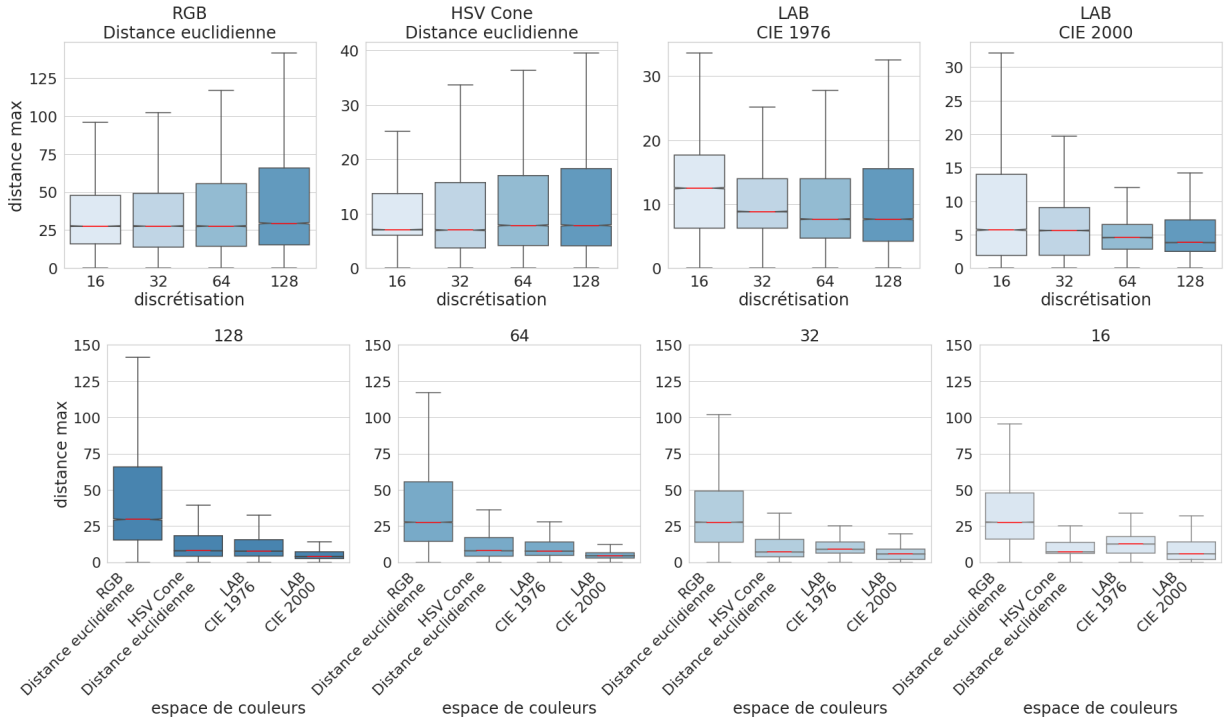


FIGURE 4.24 – Distributions de la plus grande distance entre deux couleurs pantone des images par article selon l'espace de couleurs, la distance et la discrétisation. Les deux lignes montrent la même information, dans la première une sous figure par espace de couleur, dans la deuxième une par discrétisation.

La figure 4.23 présente alors la distance moyenne des bins de couleur principale entre tous les couples d'images par article. On peut noter que pour un même article les couleurs des bins

extraits en RGB sont perceptuellement plus éloignés malgré leur attribution au même pantone (*fig.* 4.22). Les bins mode extraits dans l'espace $l^\star a^\star b^\star$ sont plus proches. Avec une discrétisation en 64 valeurs par composantes et la distance *CIE 2000*, les plus mauvaises extractions (*fig.* 4.24) sont les moins éloignées des autres bins de couleur principale extraits par article.

Sélection finale de la méthode Avec les différentes expérimentations présentées dans cette partie et les parties précédentes, nous sommes en mesure de sélectionner les meilleurs paramètres pour l'attribution d'une référence à partir d'une nomenclature. Les couleurs du vêtement projetées dans l'espace $l^\star a^\star b^\star$ permettent d'extraire des couleurs principales perceptuellement cohérentes (*fig.* 4.23, *fig.* 4.24 et section 4.4.2 *fig.* 4.18). Une discrétisation en 64 valeurs combinée avec la distance *CIE 2000* permet de pré-calculer une table de correspondance adaptée entre bins et références pantone. Ces paramètres, offrent un bon compromis entre uniformité de la table de correspondance (section 4.4.3 *fig.* 4.19) et références effectivement attribuables (section 4.4.3 *fig.* 4.20). La table permet de nommer les couleurs avec 2092 parmi les 2161 pantones. De plus, la méthode avec ces conditions attribue globalement les mêmes références pantone aux mêmes vêtements (*fig.* 4.22). Cette méthode, reversée à Retviews, permet l'analyse des vêtements à partir de la couleur standardisée avec une taxonomie déterminée. Précédemment, en s'appuyant sur le texte, la couleur pouvait ne pas être disponible, ou dans le meilleur des cas dans une nomenclature non choisie et variable.

4.5 Limite des méthodes et perspectives

Différents modèles de segmentation d'instances, de prédiction du type de vêtement, du motif tissu et la méthode d'extraction de couleur dominante des vêtements à partir d'une image ont été déployés en test à Lectra. Le processus global d'apposition de mots clefs a été intégré dans un service sous la forme d'une interface de programmation d'application (*API: **A**pplication **P**rogramming **I**nterface*) REST (***Re**presentational **S**tate **T**ransfer*). L'API est conteneurisé avec *Docker* et déployé avec *Kubernetes* managé par Azure pour permettre la montée en charge. Cette API prenant en requête des images, fournit en réponse les masques, les types, les motifs tissus, les couleurs dominantes (valeur RGB et référence pantone) de tous les vêtements contenus. Elle a ainsi permis d'effectuer de premiers tests d'utilisation.

Ces travaux ont par la suite été reversés à Retviews filiale de Lectra. La spécification des points critiques et la priorisation des améliorations sont en cours. Nous sommes actuellement en attente de retour sur les travaux. Nous soulevons à présent les limites des approches proposées et donnons des perspectives pour les résoudre.

4.5.1 Limites liées aux données

Dans nos expérimentations, nous avons considéré les données comme parfaites. Néanmoins, cela n'est pas réellement le cas. Nous notons alors deux sources de problèmes sur les données. La première concerne l'inexactitude des masques et les erreurs d'annotations. La deuxième se rapporte aux données incomplètes, au choix de la taxonomie et la non-exhaustivité des annotations.

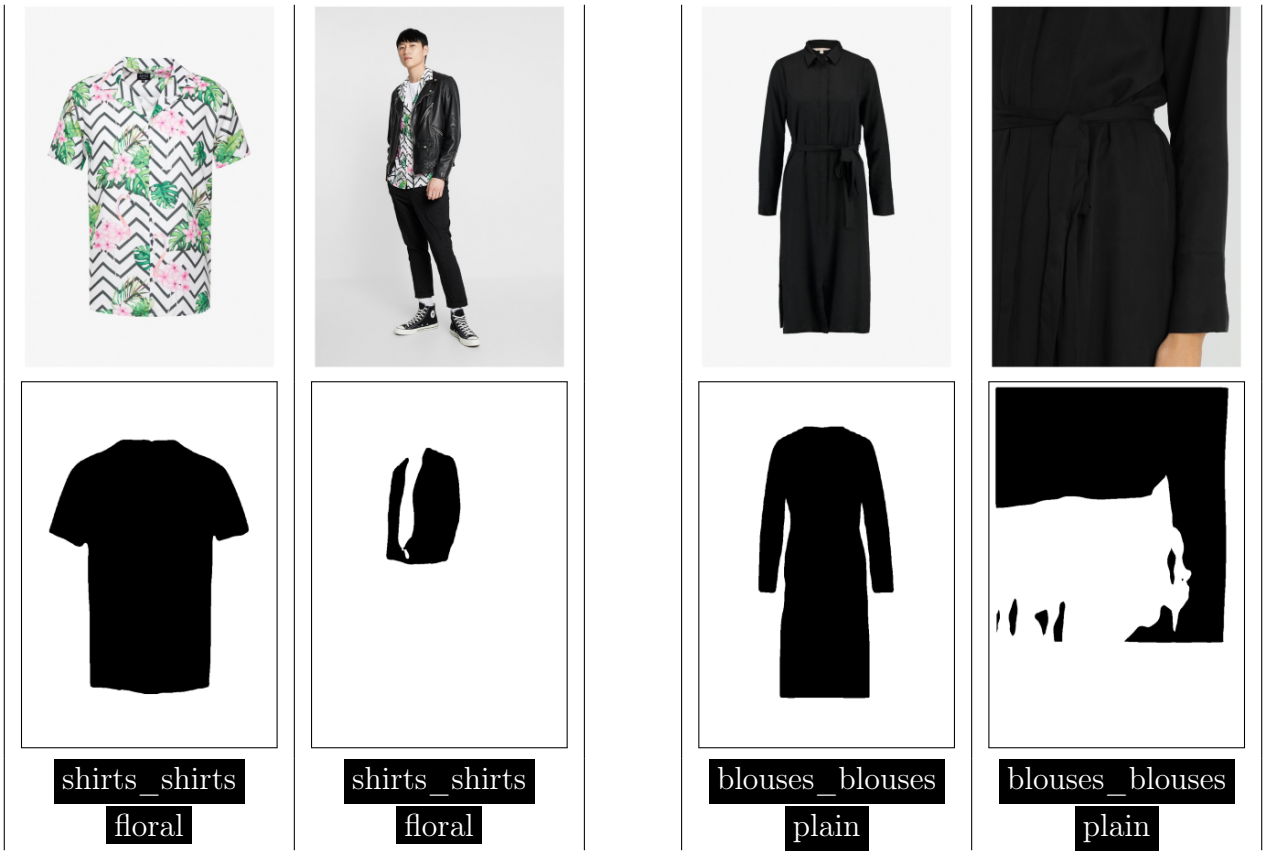


FIGURE 4.25 – Exemples d'erreur sur deux articles. Dans l'article de gauche : le masque sélectionné pour la deuxième image n'est pas le bon, l'étiquette de type pourrait aussi bien être `shirts_casual`, deux motifs sont présents. Pour l'article de droite : le masque de la deuxième image est de mauvaise qualité, l'étiquette de type semble plus être `dresses_dresses`

Nous rappelons ensuite que nous avons recours à une annotation automatique des images. Cette annotation s'appuie sur l'association de mots clefs à des étiquettes parmi une taxonomie donnée. Nous observons des erreurs liées aux descriptions des produits et nous ne pouvons donc pas garantir que les étiquettes associées à une image sont les bonnes (voir le deuxième article de la figure 4.25).

De plus, la sélection d'un masque pour une image est aussi automatique. Pour certaines images, le mauvais masque a été associé au vêtement (*ex.* masque de la deuxième image

figure 4.25). Ces masques proviennent de la prédiction fournie par le modèle de segmentation d'instances YOLACT. Aussi, il peut y avoir des masques de mauvaise qualité (*ex.* masque de la dernière image figure 4.25). Nous contrôlons cette qualité globalement, mais, ponctuellement, un masque prédit peut être inadapté pour une extraction de couleur ou une classification de motif ou de type de vêtement.

Enfin, le choix de la taxonomie pourrait être amélioré. Certaines étiquettes semblent être le nom d'un groupe de sous-étiquettes, *ex.* l'étiquette `shirts_shirts` *fig.* 4.25. Ces étiquettes pourraient alors être supprimées et remplacées par la sous-étiquette appropriée (*ex.* `shirts_casual`). Pour le motif, apposer une seule étiquette ne semble pas adapté. En effet, certains vêtements sont composés de plusieurs motifs (premier article de la figure 4.25). Une approche à étiquettes multiples, *c.-à-d.* plusieurs étiquettes de motifs pour un même vêtement, semble plus adéquate.

Les données peuvent alors être nettoyées, corrigées et annotées manuellement afin d'améliorer les résultats. Cette étape manuelle est au minimum requise sur un jeu de données de taille maîtrisable afin de permettre une meilleure évaluation. Nous disposons actuellement d'annotations d'un seul vêtement par image. Obtenir une annotation exhaustive permettrait de fournir des exemples différents pour une même image et augmenter le nombre de données pour l'entraînement.

Concernant les couleurs, il pourrait être question de sélectionner un sous ensemble de références couleurs pertinentes au lieu de travailler avec une nomenclature complète. En effet, comme nous l'avons observé, certaines références couleurs peuvent être proches perceptuellement. Il serait alors judicieux de sélectionner un nombre plus faible de noms de couleurs réellement utile et exploitable par le métier.

4.5.2 Uniformisation des vêtements

Il existe deux approches principales de présentation des données lors de l'apprentissage de modèle et de l'extraction d'information. La première que nous avons suivi consiste à s'appuyer sur la diversité des données. Cette diversité s'exprime sous différentes formes : diversité des mannequins, des poses, des prises de vue, de l'éclairage, des scènes. Une deuxième approche serait alors de contrôler ces aspects. Le but n'étant pas de les contrôler a priori lors de la capture de l'image, mais a posteriori en uniformisant les vêtements dans les images.

L'éclairage d'une scène a un impact sur les couleurs contenues dans une image. En connaissant l'éclairage d'une scène, les couleurs peuvent être normalisées. Cet éclairage peut être estimé, prédit et édité [Gardner et al., 2017 ; Song et Funkhouser, 2019 ; G. Wang et al., 2022]. L'éclairage produit aussi des ombres sur les vêtements. Ces ombres peuvent être supprimées [Zhihao Liu et al., 2021 ; Qu et al., 2017]. Enfin, il peut aussi y avoir des occlusions sur le vête-

ment, *c.-à-d.* des éléments masquant le vêtement comme les cheveux ou un bras. Ces occlusions peuvent aussi être corrigées et le vêtement reconstruit [Zhan et al., 2020].

Le mannequin, la posture et la prise de vue modifient la position du vêtement dans l'image. Ces éléments pourraient être uniformisés en alignant les vêtements à partir de leurs masques. Il existe de nombreuses méthodes d'alignement d'image [Szeliski et al., 2007]. L'alignement d'image permet d'améliorer la classification et la reconnaissance d'objets et éléments, par exemple pour la reconnaissance faciale [Schroff et al., 2015]. Appliquer à des vêtements, l'alignement peut être une étape importante de l'essayage virtuel [Han et al., 2018 ; Minar et al., 2020].

Ces différentes uniformisations peuvent garantir de présenter les vêtements dans les mêmes conditions lors de la caractérisation.

4.5.3 Amélioration des méthodes

L'architecture de classification du type de vêtement et du motif du tissu que nous avons présenté repose sur un CNN extracteur de descripteurs. Le modèle devant prendre plusieurs fois en entrée la même image (une fois par masque), nous avons sélectionné un CNN efficace parmi la littérature (EfficientNet-B0 [Tan et Le, 2019]). Cette efficacité nous permet d'obtenir un bon compromis entre résultats et temps d'inférence.

Pour améliorer ces résultats, des architectures plus conséquentes pourraient être utilisées. Maximisant la précision aux dépens de la complexité en temps, elles pourraient permettre d'améliorer les capacités de généralisation des modèles. De plus, les expérimentations pourraient se tourner vers le « *fine-tuning* » de modèles pré-entraînés sur le corpus Imagenet. En effet, une telle approche permet de s'appuyer sur des descripteurs haut niveaux déjà appris par les modèles.

L'obtention de la couleur dominante pourrait être ajoutée en troisième tâche du modèle. Cela nécessiterait soit une collecte de nouvelles données, soit une annotation de celles existantes par des experts métier.

Enfin, les deux étapes du processus (segmentation, caractérisation) sont obtenues séparément. Le processus pourrait être entraîné de bout en bout. Les deux parties devraient alors être regroupées au sein de la même architecture. Entraîner une telle architecture en multi-tâches est cependant difficilement réalisable dû à la complexité en espace, *c.-à-d.* en mémoire.

4.6 Conclusion

Dans ce chapitre, nous avons présenté des méthodes de caractérisation de vêtements à partir de leur segmentation. Le processus complet de segmentation et caractérisation a permis de réaliser des tests à Lectra et les travaux ont pu être reversés à Retviews.

Pour réaliser les expérimentations, nous avons dû constituer un corpus de données. Des images d'articles (*c.-à-d.* de vêtements) ont été collectées. Puis une taxonomie de caractérisation a été définie avec Lectra et Retviews ce qui a permis d'annoter les articles. Pour annoter les images, nous avons proposé une stratégie de sélection d'un masque et d'annotation automatique. Cette stratégie s'appuie sur des règles d'association entre étiquettes Deepfashion2 et notre taxonomie (pour le masque) et entre texte et notre taxonomie (pour le type et le motif).

Nous avons alors proposé une architecture de prédiction d'étiquettes, puis une méthode d'extraction de la couleur dominante. L'architecture, obtenue par apprentissage supervisé, s'appuie sur une image, un masque et un ensemble de patchs pour prédire le type de vêtement et le motif du tissu. L'extraction de la couleur du vêtement s'appuie quant à elle sur l'étude empirique d'une méthode de projection des couleurs dans l'espace de couleurs $l^\star a^\star b^\star$ (***l****uminance* ***a****lpha* ***b****eta*) discrétisé en 64 valeurs par composantes.

À la suite d'expérimentations concluantes d'un point de vue industriel, nous avons pu soulever plusieurs limites et proposer différentes pistes d'amélioration. Ces limites sont en cours d'examen et les améliorations à apporter sont en train d'être priorisées.

La limite principale de notre méthode de classification provient des données. Plusieurs erreurs sont introduites par l'utilisation de masques prédits, l'annotation automatique et le choix de la taxonomie des classes de types vêtements et de motifs sur les tissus. Ces erreurs pourraient être corrigées manuellement. Il serait toutefois préférable de pouvoir se passer d'une telle campagne d'annotation. Dans le chapitre suivant, nous étudions cette piste et proposons une approche faiblement supervisée.

Chapitre 5

Classification et localisation faiblement supervisées

Résumé : Dans ce chapitre, nous proposons une architecture reposant sur les patchs d'une image afin de répondre au problème de classification multi-étiquettes. Plus particulièrement nous nous intéressons au cas où une seule étiquette est disponible par image dans le corpus de données. Notre contribution concerne deux aspects. Premièrement, nous introduisons une architecture légère orientée patch s'appuyant sur le mécanisme d'attention. Deuxièmement, nous proposons une nouvelle stratégie tirant avantage de l'auto-similarité entre descripteurs pour estimer des exemples négatifs et traiter le problème de l'apprentissage positif et non étiqueté (*PU: **P**ositive and **U**nlabeled learning*). Les expériences conduites montrent que cette architecture peut être entraînée à partir de zéro, tandis que les méthodes connexes de l'état de l'art nécessitent un pré-entraînement. Ces travaux ont été acceptés pour publication et présentation à la conférence VISAPP 2023 [Jouanneau et al., 2023].

Jouanneau et al. 2023 : Warren Jouanneau, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2023). « A patch-based architecture for multi-label classification from single positive annotations ». In : *International Conference on Computer Vision Theory and Applications*

5.1 Introduction

Dans le cadre de la classification d'images, les méthodes de pointe reposent sur des ensembles toujours plus importants de données. L'annotation ou l'étiquetage de données est ainsi un enjeu majeur en apprentissage supervisé.

Les corpus d'apprentissage sont constitués de paires d'images et d'étiquettes obtenues par annotation manuelle coûteuse, par collecte automatique, ou par sélection et traitement de descriptions d'images préexistantes. Assembler un jeu de données est difficile, d'autant plus quand des experts doivent intervenir afin d'annoter les données, que des événements rares doivent être distingués, ou lors de la mise en commun de sources multiples aux taxonomies d'étiquettes inconsistantes.

La majorité des images naturelles requiert une classification avec de multiples étiquettes, afin d'être caractérisées précisément. Le contenu d'une image est en effet généralement riche en informations, car il comprend de multiples éléments structurés. Associer une seule étiquette à une image est donc un cadre trop restrictif. Pour être précis, il est ainsi préférable d'annoter une image avec différentes étiquettes non exclusives (*ex.* présence/absence de bois, de métal, tissu, *etc.*).

En apprentissage supervisé, le corpus doit contenir des exemples positifs et d'autres négatifs pour chaque étiquette. Les paires images-étiquettes doivent être le plus exhaustives possibles. Toute annotation manquante ou incorrecte d'une image I_n pour une étiquette l conduit à un exemple erroné pour cette étiquette. De telles erreurs peuvent avoir un impact négatif sur l'étiquetage complet de l'image I_n et sur la caractérisation de l'étiquette l pour les autres images du corpus. Naturellement, obtenir des annotations multi-étiquettes sans erreur rend la constitution d'un corpus d'autant plus complexe.

Afin de faciliter l'assemblage de jeu de données d'entraînement, les méthodes d'apprentissage faiblement supervisées sont conçues de manière à exploiter une annotation partielle des données. Ces méthodes combinent des approches allant de l'apprentissage complètement supervisé au non supervisé, *ex.* l'apprentissage où seul un faible nombre d'exemples est disponible pour chaque étiquette (*few shot learning*).

Un cas particulier d'apprentissage faiblement supervisé est l'apprentissage positif et non étiqueté (*PU: **P**ositive and **U**nlabeled learning*) [Bekker et Davis, 2020]. En apprentissage PU, seulement un étiquetage partiel positif est disponible. Dans le cas de l'apprentissage PU multi-étiquettes, seulement un sous ensemble des images est annoté pour chaque étiquette dans le jeu d'apprentissage. Pour le reste des images, nous ne disposons d'aucune information. Il est alors impossible de savoir si l'étiquette est présente ou non dans une image. De plus, plusieurs étiquettes peuvent devoir être identifiées dans une seule image.

Obtenir des annotations est grandement simplifié dans le cas de l'apprentissage PU multi-étiquettes. En effet, ce contexte est adapté à la collection automatique de données et la fusion de corpus. Pour chaque étiquette, des exemples positifs peuvent être collectés indépendamment de ce qu'ils représentent pour les autres étiquettes. Il n'est donc pas nécessaire de savoir si c'est un exemple positif ou négatif d'une autre étiquette. Plus globalement, ne nécessitant pas d'exemples négatifs et la complétude de l'étiquetage, cette forme d'apprentissage est parfaitement adaptée pour traiter des taxonomies d'étiquetages hétérogènes provenant de différents ensembles de données.

S'adapter à l'absence d'exemples négatifs est une des difficultés rencontrées par l'apprentissage PU. Cette difficulté est exacerbée dans le cadre d'étiquettes multiples, car une même image peut à la fois être positive pour certaines étiquettes et négative pour d'autres. Nous présentons cette problématique en détail dans la section 5.2.

Dans ces travaux, nous proposons d'aborder la problématique des exemples négatifs à l'aide d'une approche orientée patch. Nous soutenons que travailler au niveau patch est plus adapté que le niveau image pour une caractérisation multi-étiquettes. Si une image est un exemple positif d'une étiquette, alors certains de ses patchs peuvent être considérés positifs et d'autres négatifs. Cependant, pour une image positive, la relation patch-étiquette n'est pas connue. Cela rend la problématique de l'apprentissage PU multi-étiquettes difficile. Nous présentons les méthodes de l'état de l'art traitant des patchs dans la section 5.3.

Notre principale contribution, présentée en sections 5.4 et 5.5, est une architecture légère orientée patch et spécialisée au cas de l'apprentissage PU multi-étiquettes. Premièrement, une image est considérée comme un ensemble de patchs, puis des représentations d'images conditionnées aux étiquettes sont obtenues grâce à un mécanisme d'attention. Dans un second temps, en supposant qu'un patch contient au plus une seule étiquette, nous proposons d'estimer les exemples négatifs en tirant avantage de l'auto-similarité entre les représentations d'images construites précédemment à partir des patchs.

Dans la section 5.6, les résultats d'expérimentations démontrent que notre approche orientée patchs est adaptée à la classification multi-étiquettes à partir d'une seule annotation positive par image tout en localisant explicitement les étiquettes. Quand les modèles sont entraînés à partir de zéro, notre proposition généralise le problème plus vite et mieux qu'un Resnet-50. De plus, elle est significativement plus légère (le nombre de paramètres est réduit d'un facteur 100).

5.2 Problématique : Apprentissage positif et non étiqueté pour la classification multi-étiquettes

La classification multi-étiquettes d'images peut être scindée en de multiples tâches de classification à étiquette unique [Read et al., 2009], où chaque classifieur doit prédire spécifiquement une étiquette. Cependant, Wei et al. en 2014 démontrent l'intérêt de traiter la problématique multi-étiquettes de façon conjointe. Dans les images, il existe une corrélation entre localisation et étiquettes. Ainsi, les méthodes de détection ont pour rôle non seulement d'inférer les étiquettes sur une image, mais aussi d'estimer leur localisation par le biais de boîtes englobantes [Redmon et al., 2016 ; Ren et al., 2015] ou de masques de segmentation [He, Gkioxari et al., 2017] (chapitre 2).

L'apprentissage positif et non étiqueté (*PU:* ***P****ositive and* ***U****nlabeled learning*) est un problème de classification faiblement supervisé où seulement des exemples positifs sont disponibles. Comme étudié dans l'article de synthèse de Bekker et Davis en 2020, de nombreuses méthodes ont abordé ce problème dans le cadre d'une unique étiquette.

Cependant, il n'existe que peu de méthodes dédiées à l'apprentissage PU pour les problèmes de classification à étiquettes multiples. Comme détaillé par Cole et al. en 2021, la plupart des travaux considèrent les problèmes dans lesquels, soit plusieurs étiquettes positives sont connues par image [Kanehira et Harada, 2016], soit une positive et une négative sont connues [X. Huang et Yan, 2018], soit il existe des exemples positifs et négatifs pour chaque étiquette [Ishida et al., 2017].

Dans ce chapitre, nous nous concentrons à l'application complexe de la classification à étiquettes multiples à partir d'une unique étiquette positive connue par image du jeu de données d'entraînement. Pour Mac Aodha et al. en 2019, les étiquettes inconnues (*c.-à-d.* toutes les étiquettes qui ne sont pas des positives connues pour cette image) sont définies comme vérités terrain négatives dans la fonction de coût optimisée pendant l'entraînement. Cela correspond à une pénalisation uniforme de la présence des étiquettes non connues. Cette pénalisation peut être enrichie d'une fonction dédiée à la cohérence spatiale [Verelst et al., 2022]. Cole et al. en 2021 proposent d'améliorer ce type de méthode avec une stratégie d'estimation régularisée des étiquettes en ligne (*ROLE:* ***R****egularized* ***O****nline* ***L****abel* ***E****stimation*). La modélisation ROLE améliore la fonction de coût avec un terme pénalisant la distance entre le nombre d'étiquettes positives prédites pour une image et un hyper-paramètre. Cet hyper-paramètre correspond au nombre d'étiquettes positives moyen par image dans le jeu d'entraînement. Il est à noter que cette valeur est inconnue pour des cas d'applications réels. Puis, une estimation en ligne des étiquettes est mise en place pour les étiquettes non observées. Cela consiste en l'apprentissage

de paramètres qui doivent correspondre à la vérité terrain des étiquettes inconnues (positives ou négatives). Ces estimations sont obtenues dans une branche séparée du modèle. Dans un procédé itératif, elles sont ensuite comparées à la prédiction de classification courante dans une fonction de coût dédiée.

Le modèle ROLE fait une proposition intéressante avec l'estimation d'exemples négatifs, néanmoins il ne s'appuie pas sur le contenu des données pour répondre à l'aspect "multiples étiquettes" du problème. La stratégie en ligne stabilise l'apprentissage à travers la mémorisation des précédentes prédictions. Mais en pratique, cela renforce les tendances du modèle courant en rapprochant les poids de prédiction des étiquettes de plus en plus vers 0 ou 1.

Dans la suite du chapitre, nous proposons une approche basée sur les patchs et montrons que cette stratégie est appropriée pour estimer des exemples négatifs d'étiquettes dans le contexte de l'apprentissage PU à étiquettes multiples.

Définition formelle Étant donné une taxonomie L, c'est-à-dire un ensemble d'étiquettes $l \in L$, l'objectif est de déterminer $\mathbb{P}(I_n|l)$, la probabilité de présence de l'étiquette l dans l'image I_n. Nous désignons par $\boldsymbol{y}_n = \{y_{n,l}\}_{l \in L}$ la vérité terrain qui indique si une étiquette $l \in L$ est présente ($y_{n,l} = 1$) ou non ($y_{n,l} = 0$) dans une image I_n. Le problème multi-étiquettes peut alors être formulé comme l'estimation d'un score d'étiquetage $\hat{\boldsymbol{y}}_n = \{\hat{y}_{n,l}\}_{l \in L}$, où $\hat{y}_{n,l} \in [0, 1]$ indique la présence ($\hat{y}_{n,l} \to 1$) ou l'absence ($\hat{y}_{n,l} \to 0$) du label l dans l'image I_n.

L'apprentissage d'un modèle de classification à étiquettes multiples de façon entièrement supervisée nécessite des données complètement annotées, dont la collecte est très coûteuse. Dans notre cas, ces annotations exhaustives ne sont pas disponibles. Nous disposons d'une vérité terrain partielle des images du corpus. Cette vérité partielle contient une information partielle sur la nature positive $\boldsymbol{z}_n^+$ et négative $\boldsymbol{z}_n^-$ des étiquettes pour l'image I_n. Nous définissons alors les variables $z_{n,l}^+$ et $z_{n,l}^-$ pour représenter respectivement la présence et l'absence de l'étiquette l dans l'image I_n du corpus d'entraînement. Si $z_{n,l}^+ = z_{n,l}^- = 0$ alors nous ne disposons d'aucune information sur l'étiquette l. Enfin, les deux annotations sont supposées compatibles, c'est-à-dire que $z_{n,l}^+ = z_{n,l}^- = 1$ est impossible.

Dans le contexte de l'apprentissage PU traité ici, aucun négatif n'est connu ($\forall_{l \in L}\, z_{n,l}^- = 0$) et les positifs ne sont que partiellement connu ($\{l|l \in L, z_{n,l}^+ = 1\} \subset \{l|l \in L, y_{n,l} = 1\}$). Dans notre application, nous considérons le cas où un seul positif est connu ($\sum_{l \in L} z_{n,l}^+ = 1$) par image I_n.

5.3 Utilisation des patchs pour la classification multi-étiquettes

Pour résoudre le problème de l'apprentissage PU à étiquettes multiples, nous proposons une méthode faiblement supervisée capable d'exploiter un étiquetage partiel. Notre méthode s'appuie sur l'utilisation de patchs d'images. Elle est construite sur l'hypothèse qu'un patch contient majoritairement un seul objet, i.e. une seule étiquette. Si d'autres étiquettes sont présentes dans le patch, elles ne le sont que de manière minoritaire. Ces autres étiquettes sont vraisemblablement majoritaires dans les patchs voisins.

5.3.1 Utilisation des patchs en traitement d'images et classification

Si les approches basées sur les patchs ont d'abord été introduites pour des problématiques de synthèse de texture [Efros et Leung, 1999], leur intérêt pour différentes tâches de traitement et d'analyse d'images ont aussi été démontrées par la suite. Grâce à la capacité des patchs à exploiter les auto-similarités présentes dans les images naturelles, des résultats de référence ont été obtenus pour par exemple le débruitage [Buades et al., 2005], la super-résolution [Freeman et al., 2002], la segmentation, l'étiquetage [Coupé et al., 2011], la classification [Varma et Zisserman, 2008], *etc.*. La représentation d'images à partir de patchs a fait l'objet d'études approfondies et nous renvoyons vers L. Liu, J. Chen et al. en 2019 pour une revue des méthodes existantes, du cadre posé par les sacs de mots jusqu'aux récents modèles d'apprentissage profond.

Dans ce contexte de l'apprentissage profond, l'architecture Transformer [Vaswani et al., 2017] s'appuie également sur les auto-similarités contenues dans les données traitées. Cette architecture, originellement proposée par la communauté du traitement du langage naturel, a été transposée au cas de l'image. Pour cela, la méthode ViT [Dosovitskiy et al., 2020] considère une image comme un ensemble ou une séquence (si la position des patchs est encodée) de patchs. ViT atteint des résultats impressionnants en classification sans aucune couche de convolution.

De nombreuses extensions des Transformers appliquées aux images se concentrent sur la pertinence des éléments soumis au réseau. Ainsi, CrossViT [C.-F. R. Chen et al., 2021] s'appuie sur des patchs de différentes tailles pour apporter une information multi-échelle. Un réseau de neurones à couche de convolution (*CNN: **C**onvolutional **N**eural **N**etwork*) peut aussi être utilisé en amont du Transformer, pour encoder l'image [Tete Xiao et al., 2021] ou la représenter par une pyramide de descripteurs [D. Zhang et al., 2020].

Avec les méthodes d'apprentissage moderne, des ensembles de représentations de patchs peuvent être utilisés efficacement pour la classification d'images. Les travaux sur ConvMixer [Trockman

et Kolter, 2022] ont notamment montré que la performance de l'approche Transformer est due à la modélisation de l'image sous forme d'un ensemble de patchs, plutôt que liée à l'architecture à proprement parler. Lorsque l'on aborde un problème de classification d'image, il est alors nécessaire de combiner les informations disponibles au niveau des patchs pour prendre une décision au niveau de l'image.

5.3.2 Utilisation des patchs dans le mécanisme d'attention

Nous présentons ici les méthodes permettant d'obtenir des représentations d'images pertinentes à partir de descripteurs de ses patchs. Une représentation pertinente des patchs doit permettre la classification à étiquettes multiples et traiter la présence de plusieurs instances de ces étiquettes dans l'image. Pour ce faire, l'information contenue par l'ensemble de ces descripteurs de patchs doit être agrégée. Dans la littérature, l'agrégation des éléments d'un ensemble est principalement réalisée par des opérateurs de regroupement (*pooling*) comme la moyenne, le maximum, le minimum, la somme des représentations des éléments [Zaheer et al., 2017]. Ainsi, par exemple le regroupement des cartes de descripteurs des champs réceptif d'un CNN est généralement réalisé avec une moyenne globale [M. Lin et al., 2013].

La popularité du mécanisme d'attention, qui réalise une somme pondérée des représentations des éléments, a conduit à de nouvelles formes de méthodes de regroupement [Ilse et al., 2018a]. Les Transformers avec attention à têtes multiples [Vaswani et al., 2017] peuvent aussi être considérés pour réaliser un regroupement d'éléments. Les blocs décodant des Transformers utilisent une attention croisée avec des poids obtenus à partir du score de similarité entre la représentation des éléments et des "requêtes". Dans le cas des patchs, ces "requêtes" peuvent être vues comme un livre-code (ou codebook, i.e. une bibliothèque de textons) [Zhao et al., 2022]. Ces requêtes peuvent être conçues au préalable ou être des paramètres appris par le modèle. Une des limitations principales des Transformers est leur complexité quadratique par rapport aux dimensions des données d'entrée. Afin de réduire la charge de calcul, le Perceiver [Jaegle et al., 2021] utilise alors des requêtes apprises afin de construire des représentations latentes intermédiaires de plus petite dimension. De plus, il est possible de conditionner les requêtes pour qu'elles soient dépendantes d'une tâche en particulier et ainsi les utiliser pour réaliser un regroupement d'éléments. Les requêtes peuvent donc être définies comme des représentations d'étiquettes [Lanchantin et al., 2021], permettant de réaliser un regroupement indépendant pour chaque étiquette.

Les regroupements d'éléments, tels que réalisés par le mécanisme d'attention, permettent de traiter naturellement les multiples instances d'une étiquette dans une image. Dans le domaine de la vision par ordinateur, l'apprentissage d'instances multiples (*MIL: **M**ultiple **I**nstance*

***L**earning*) consiste à la fois à détecter la présence d'une étiquette et à en localiser les instances correspondantes dans l'image [Carbonneau et al., 2018]. Le mécanisme d'attention apporte une solution à ces problèmes [Ilse et al., 2018b]. La détection d'une étiquette est en effet donnée par le regroupement des représentations de patchs effectué avec la requête correspondante. Le score de similarité entre un patch et une requête indique le degré de participation du patch dans la prise de décision pour une étiquette. Si un patch a un poids important dans le regroupement conditionné à une étiquette prédite comme positive, alors ce patch doit contenir des informations pertinentes relatives à celle-ci. Par conséquent, les scores de similarité peuvent aider à localiser des sous-zones de l'image correspondant à une étiquette.

Dans un contexte supervisé, le mécanisme d'attention sur les patchs est alors adapté à la fois aux cas des étiquettes multiples et des instances multiples. Cependant, son application dans le contexte faiblement supervisé de l'apprentissage PU au niveau image doit être adapté et requiert le développement de nouvelles méthodes.

5.4 Architecture multi-étiquettes orientée patch

Nous fournissons à présent une description détaillée de notre architecture orientée patchs. Comme illustré dans la figure 5.1, l'architecture comporte cinq blocs basés sur la méthodologie des "sacs de mots" [L. Liu, J. Chen et al., 2019]. Un sous-ensemble de patchs est tout d'abord sélectionné. Les patchs sont ensuite individuellement soumis à un CNN commun afin d'obtenir des représentations (ou embeddings) de patchs. Dans l'espace de ces représentations, les descriptions des étiquettes du *codebook* (les représentations des étiquettes) coexistent. Un regroupement des représentations de patch pour chaque étiquette est réalisé en utilisant le mécanisme d'attention. Enfin, la classification de l'image est effectuée à partir des représentations d'images conditionnées aux étiquettes.

Extraction de patchs (Bloc 1 de la figure 5.1). Dans notre architecture, une image est tout d'abord convertie en un ensemble de patchs extraits à différentes résolutions. Chaque image $I_n \in \mathbb{R}^{H \times W \times C}$ de hauteur H, de largeur W et de C canaux est redimensionnée R fois avec un ratio constant d, en utilisant une interpolation bilinéaire.

Pour chaque image résultante, les patchs P_n sont extraits grâce à une fenêtre glissante de taille $h \times w$ parcourant l'image horizontalement et verticalement avec un pas de déplacement (stride). Nous désignons un patch par $p \in \mathbb{R}^{h \times w \times C}$. La taille des patchs est la même à travers toutes les résolutions. Les patchs contiennent alors une information fine pour le plus haut niveau et de plus en plus grossière lorsque la résolution décroit.

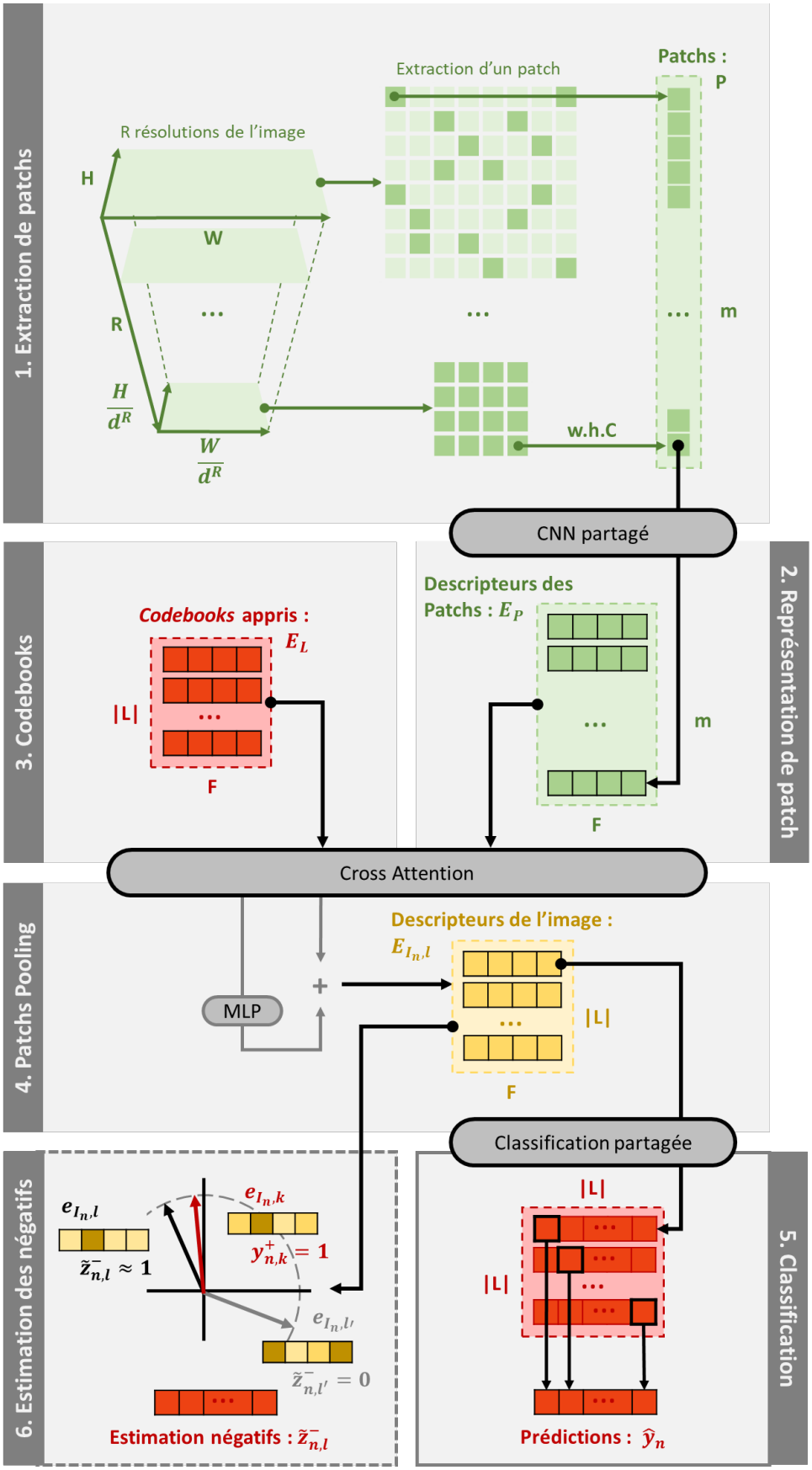


FIGURE 5.1 – Proposition d'une architecture multi-résolution basée patchs

Dans ces travaux, nous considérons des patchs extraits de manière régulière dans l'image avec une fenêtre glissante. Les patchs forment alors une grille parfaite de l'image. Cependant, quand l'image est trop grande, un sous échantillonnage à chaque résolution peut être réalisé afin de limiter la charge de calcul.

Représentation de patch (Bloc 2 de la figure 5.1). Chaque patch est donné en entrée d'un CNN. Il s'agit du même modèle pour tous les patchs, les poids sont donc partagés. Avec ce modèle sous-jacent, tous les patchs sont projetés dans le même espace latent de dimension F, $E_{P_n} = \{e_p \in \mathbb{R}^F, p \in P_n\}$ donnant ainsi un vecteur de descripteurs $e_p \in E_{P_n}$ tenant le rôle de représentation de patch. Le CNN est composé de plusieurs blocs EfficientNet [Tan et Le, 2019].

Représentation de *codebook* (Bloc 3 de la figure 5.1). Chaque étiquette est associée à son propre patch représentatif, e_l, que nous nommons *codebook*. Nous supposons que chaque représentation e_l contient les descripteurs que devrait contenir un patch pour être discriminé comme positif relativement à l'étiquette l. L'ensemble des *codebooks*, noté $E_L = \{e_l \in \mathbb{R}^F, l \in L\}$, est obtenu par entraînement.

Représentation d'image par regroupement (Bloc 4 de la figure 5.1). Une représentation de l'image est obtenue à partir du regroupement (*pooling*) des descripteurs des patchs. À cette fin, nous proposons de considérer le regroupement par mécanisme d'attention. Le regroupement est alors réalisé en utilisant l'attention croisée entre l'ensemble E_{P_n} des descripteurs de patchs et les *codebooks* appris E_L.

L'attention peut être vue comme un mécanisme en deux étapes. La première consiste à évaluer la pertinence de chaque patch sélectionné relativement à chaque étiquette possible. Pour ce faire, nous considérons le produit scalaire entre les vecteurs pour définir une matrice de score A aux poids $\alpha_{l,p}$ entre les représentations e_p d'un patch p et e_l d'un label l :

$$\alpha_{l,p} = \frac{\exp(e_l \,.\, e_p)}{\sum_{q \in P_n} \exp(e_l \,.\, e_q)}. \tag{5.1}$$

La deuxième étape calcule alors une somme pondérée des représentations de patch avec le produit AE_{P_n}. Inspiré de Vaswani et al. en 2017, nous définissons enfin les représentations de l'image comme :

$$E_{I_n} = f(AE_{P_n}) + AE_{P_n}, \tag{5.2}$$

où, f est une projection linéaire. Avec cette forme d'attention, nous obtenons plusieurs représentations d'image $E_{I_n} = \{e_{I_n,l} | e_{I_n,l} \in \mathbb{R}^F, l \in L\}$ conditionnées aux étiquettes L. Ainsi, une

représentation d'image $e_{I_n,l}$ regroupe des descripteurs d'un sous-ensemble de patchs correspondant à la représentation globale d'un patch d'étiquette $l \in L$.

Classifieur (Bloc 5 de la figure 5.1). Pour réaliser la classification multi-étiquettes, nous proposons d'utiliser plusieurs classifieurs dont les poids sont partagés. Étant donné une représentation d'image $e_{I_n,l}$, un classifieur fourni une prédiction $\hat{y}_l$ relative à la présence de l'étiquette l dans l'image I_n à partir d'un vecteur de poids appris w_l. En pratique, un classifieur fournit toutes les prédictions $\hat{\boldsymbol{y}}$ à partir de la matrice des poids W et d'une seule représentation d'image $e_{I_n,l}$. Les poids W sont en effet partagés entre tous les classifieurs. Cependant, pour chaque classifieur associé à une représentation d'image $e_{I_n,l}$ donnée, seule la prédiction pour l'étiquette l est conservée après l'utilisation de la fonction softmax :

$$\hat{y}_{n,l} = \frac{\exp(w_l \,.\, e_{I_n,l})}{\sum_{k \in L} \exp(w_k \,.\, e_{I_n,l}))}. \tag{5.3}$$

Avec le partage des poids et l'utilisation de l'opérateur softmax, l'ensemble des classifieurs sont incités à réaliser un partitionnement de l'espace latent. Notre objectif avec une telle modélisation de la classification est d'obtenir une spécialisation de l'espace des descripteurs selon les étiquettes. Autrement dit, cette architecture de classification encourage le *codebook* e_l à être le centroïde des représentations des patchs e_p contenant supposément l'étiquette l.

Nous suggérons, qu'avec le regroupement de patchs par mécanisme d'attention et les classifieurs, notre architecture renforce la cohérence intra-classe et la dissimilarité inter-classe. En effet, alors que le produit scalaire (5.1) entre les représentations de patchs relatives à la même étiquette est maximisé, le bloc de classification vise implicitement à minimiser le produit scalaire entre les représentations d'images conditionnées aux étiquettes apparaissant dans la relation (5.3).

5.5 Stratégie d'entraînement avec estimation d'exemples négatifs

Nous présentons maintenant la fonction de coût multi-étiquettes utilisée pour obtenir une prédiction $\hat{\boldsymbol{y}}_n$ de la vérité terrain $\boldsymbol{y}_n$ à partir d'exemples positifs $\boldsymbol{z}_n^+$ et négatifs $\boldsymbol{z}_n^-$. Tout d'abord, pour introduire les différentes fonctions de coût et éviter les confusions, nous examinons les différences entre les problèmes multi-classes et multi-étiquettes.

5.5.1 Fonctions de coût

Apprentissage supervisé multi-classes. Pour la classification multi-classes, les classes sont mutuellement exclusives : pour chaque image, il n'existe qu'un unique label l tel que $y_{n,l} = 1$ dans la vérité terrain, et $y_{n,k} = 0$ pour tous les autres $k \neq l$. Les prédictions $\hat{y}$, sont alors contraintes d'appartenir au simplexe (*c.-à-d.* $\forall_l\, \hat{y}_{n,l} \geq 0$ et $\|\hat{\boldsymbol{y}}_n\|_1 = 1$). Dans ce cas, la fonction de coût classique est l'entropie croisée (*CE: **C**ross **E**ntropy*) $\mathcal{L}_{CE}$ entre les exemples positifs $\boldsymbol{z}_n^+$ et les prédictions normalisées $\hat{\boldsymbol{y}}_n$:

$$\mathcal{L}_{CE}(\boldsymbol{z}_n^+, \hat{\boldsymbol{y}}_n) = -\sum_{l \in L} z_{n,l}^+ \log(\hat{y}_{n,l}). \tag{5.4}$$

Apprentissage supervisé multi-étiquettes. Dans le cas de la classification multi-étiquettes, plusieurs vérités terrain positives sont possibles pour une même image ($\sum_{l \in L} y_{n,l} \geq 1$). La contrainte du simplexe n'a alors plus de raison d'être, et chaque prédiction d'étiquette est un score indépendant $\hat{y}_{n,l} \in [0, 1]$. Pour traiter ce problème de manière supervisée, il est nécessaire de disposer de l'ensemble des exemples négatifs, contenus dans $\boldsymbol{z}_n^-$. Nous pouvons alors utiliser l'entropie croisée (*BCE: **B**inary **C**ross **E**ntropy*) $\mathcal{L}_{BCE}$, qui est une fonction de coût standard en classification à étiquettes multiples. Pour chaque étiquette $l \in L$, la BCE est la somme de deux termes de CE (5.4) entre les positifs $z_{n,l}^+$ (*resp.* negatifs $z_{n,l}^-$) des observations de la vérité terrain et les positifs $\hat{y}_{n,l}$ (*resp.* negatifs $1 - \hat{y}_{n,l}$) des prédictions :

$$\mathcal{L}_{BCE}(\hat{\boldsymbol{y}}_n) = \mathcal{L}_{CE}(\boldsymbol{z}_n^+, \hat{\boldsymbol{y}}_n) + \mathcal{L}_{CE}(\boldsymbol{z}_n^-, 1 - \hat{\boldsymbol{y}}_n). \tag{5.5}$$

Apprentissage positif et non étiqueté (PU). Dans le contexte de la classification multi-étiquette, l'apprentissage positif et non étiqueté (*PU: **P**ositive and **U**nlabeled learning*) soulève deux difficultés principales : (1) la vérité terrain est partiellement observée (donc les positifs ne sont que partiellement étiquetés) et (2) nous avons accès seulement à des exemples positifs ($\forall_{l \in L}\, z_l^- = 0$). Pour aborder le problème de l'étiquetage partiel des exemples positifs, nous supposons qu'en utilisant la BCE les descripteurs appris pour une étiquette sur une image se transposent globalement pour cette même étiquette aux autres images.

Afin de s'adapter à l'absence d'exemples négatifs, on distingue deux possibilités. La première consiste à n'utiliser que les exemples positifs et à ne pas traiter les négatifs avec la fonction de coût $\mathcal{L}_{CE}(\boldsymbol{z}_n^+, \hat{\boldsymbol{y}}_n)$ lors de l'entraînement du modèle. Cependant, cette fonction est globalement minimisée avec la solution triviale prédisant toutes les étiquettes comme positives pour toutes les images. La deuxième option est de considérer toutes les étiquettes, sauf celles observées comme positives, comme négatives, c'est-à-dire d'entraîner le modèle avec la fonction de coût

$\mathcal{L}_{CE}(\boldsymbol{z}_n^+, \hat{\boldsymbol{y}}_n) + \lambda \mathcal{L}_{CE}(1 - \boldsymbol{z}_n^+, 1 - \hat{\boldsymbol{y}}_n)$. Le paramètre de pénalisation $\lambda \geq 0$ est en général difficile à fixer. Le modèle réalise alors une pénalisation aveugle et homogène, indépendamment du contenu de l'image, ce qui encourage la prédiction d'une seule étiquette positive $\sum_{l \in L} \hat{y}_{n,l} \approx 1$.

Nous proposons de faire un compromis entre ces deux approches avec la fonction de coût faiblement négative (*WN:* ***W****eak* ***N****egative*)

$$\mathcal{L}_{WN} = \mathcal{L}_{CE}(\boldsymbol{z}_n^+, \hat{\boldsymbol{y}}_n) + \mathcal{L}_{CE}(\tilde{\boldsymbol{z}}_n^-, 1 - \hat{\boldsymbol{y}}_n), \tag{5.6}$$

où $\tilde{\boldsymbol{z}}_n^-$ est une estimation de la vérité terrain négative inconnue. Nous soulignons une différence principale entre notre modélisation et celle introduite par Cole et al. en 2021. Dans [Cole et al., 2021], toutes les étiquettes non observées sont apprises, c'est-à-dire que les valeurs inconnues de la vérité terrain $\boldsymbol{y}_n$ sont estimées en ligne pour tout $z_{n,l}^+ = 0$. *A contrario*, notre but est d'estimer seulement les exemples négatifs les plus vraisemblables $\tilde{z}_{n,l}^- = 1$, ce qui correspond à un sous ensemble d'étiquettes $y_{n,l} = 0$ de la vérité terrain inconnue. Dans la suite, nous détaillons comment tirer profit de nos représentations d'images pour obtenir ces exemples négatifs.

5.5.2 Estimation des négatifs à partir de l'auto-similarité de représentations

Afin d'aborder le problème de l'apprentissage positif et non étiqueté (*PU:* ***P****ositive and* ***U****nlabeled learning*), nous proposons d'estimer les étiquettes négatives $\tilde{\boldsymbol{z}}_n = \{\tilde{z}_{n,l}\}_{l \in L}$ pour chaque image. Cette estimation est faite à partir de l'information contenue dans les représentations d'images E_{I_n} conjointe à notre hypothèse de départ : dans un patch, une étiquette est majoritairement observée.

Auto-similarité au sein des représentations d'image. Tout d'abord, nous montrons que deux représentations d'image, $e_{I_n,l}$ et $e_{I_n,k}$ pour les étiquettes l et k respectivement, sont similaires si elles proviennent de patchs similaires. Pour mesurer la proximité et la ressemblance des patchs, nous considérons la similarité cosinus $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u} \,.\, \boldsymbol{v}}{\|\boldsymbol{u}\|_2 \times \|\boldsymbol{v}\|_2}$ entre deux vecteurs $\boldsymbol{u}$ and $\boldsymbol{v}$. Comme définit dans (5.2), les représentations d'images $e_{I_n,l} \in E_{I_n,l}$ sont conçues pour sélectionner les descripteurs de patchs $e_p \in E_{P_n}$ qui sont corrélés avec les descripteurs d'étiquettes $e_l \in E_L$.

Nous rappelons également qu'avec notre bloc de classification (voir le bloc 5 de la figure 5.1 et la discussion page 97), les descripteurs d'étiquettes E_L sont encouragés à former un partitionnement de l'espace latent. En conséquence, si la similarité cosinus $\text{sim}(e_{I_n,l}, e_{I_n,k})$ entre deux représentations d'images pour deux étiquettes est grande, alors ces deux représentations

proviennent sûrement de sous-ensembles similaires de descripteurs de patchs e_p et par extension des mêmes patchs, ou de patchs similaires.

Ensuite, nous rappelons que nous supposons qu'une seule étiquette est majoritaire par patch. Ainsi, si un ensemble de patchs est réellement représentatif d'une étiquette l^*, cet ensemble ne peut être pertinent pour la caractérisation d'une autre étiquette $k \neq l^*$. Le modèle ne doit donc pas retourner un score de classification positif pour une étiquette k différente de l^* en se basant sur le même ensemble de patchs.

En combinant ces observations, nous concluons que lorsque la similarité cosinus $\text{sim}(e_{I_n,l}, e_{I_n,k})$ est grande, les représentations de l'image $e_{I_n,l}$ et $e_{I_n,k}$ sont obtenues à partir des mêmes sous-ensembles de patchs. Au moins une des prédictions pour les étiquettes l et k doit donc être négative.

Estimer les étiquettes négatives. En s'appuyant sur l'analyse précédente, nous proposons de nous servir de la similarité cosinus entre les représentations d'image pour estimer les exemples négatifs $\tilde{\boldsymbol{z}}_n^-$. Nous rappelons qu'une étiquette, disons z_{n,l^*}, est observée sur chaque image de l'ensemble d'entrainement. Ainsi, nous utilisons la valeur de similarité cosinus avec les étiquettes observées, $\text{sim}(e_{I_n,l^*}, e_{I_n,k}) \in [-1, 1]$, pour déterminer si les étiquettes non observées k peuvent être considérées comme négatives. À cette fin, nous définissons d'abord les poids

$$\beta_{l,k} = \varphi(\text{sim}(e_{I_n,l}, e_{I_n,k}), \theta), \tag{5.7}$$

où $\varphi(x, \theta) = \mathbb{1}_{[x>\theta]}x$ est l'opérateur d'unité linéaire rectifiée (*ReLU: **Re**ctified **L**inear **U**nit*) seuillé avec le paramètre θ. Ce paramètre $\theta \in [-1, 1]$ représente la valeur seuil pour laquelle la similarité est considérée comme assez faible pour que deux représentations puissent être considérées comme différentes. Choisir $\theta \geq 0$ garantit $\varphi(\text{sim}(e_{I_n,l}, e_{I_n,k}), \theta) \in [0, 1]$.

Enfin, comme illustré dans le bloc 6 de la figure 5.1, nous estimons pour chaque image un score d'exemple négatif $\tilde{z}_{n,l}^-$ pour toutes les étiquettes non observées $l \in L$ (*c.-à-d.* quand $z_{n,l}^+ = 0$) :

$$\tilde{z}_{n,l}^- = \max_{\substack{k \in L \\ z_{n,k}^+ = 1}} \beta_{l,k}. \tag{5.8}$$

Cela implique que l'étiquette l peut être considérée comme (faiblement) négative si la représentation de l'image lui correspondant est assez similaire à la représentation d'une des étiquettes observées $z_{n,k}^+ = 1$. Le score des négatifs $\tilde{\boldsymbol{z}}_n^-$ prend des valeurs continues dans $[0, 1]$ donnant ainsi des scores $0 < \tilde{z}_{n,l}^- < 1$ pour les exemples négatifs d'étiquette. Le score est de 0 si la similarité cosinus est inférieure au seuil θ et grandit avec la cette similiarité ($\tilde{z}_{n,l'}^-$ bloc 6 de la figure 5.1).

En considérant toutes les prédictions $\hat{\boldsymbol{y}}_n$ à partir des n images I_n du corpus et en rappelant

la définition de l'entropie croisée (*CE: **C**ross **E**ntropy*) (5.4), notre fonction de coût faiblement négative (*WN: **W**eak **N**egative*) $\mathcal{L}_{WN}$ (5.6) s'écrit finalement :

$$\mathcal{L}_{WN} = \sum_n \mathcal{L}_{\text{CE}}(\mathbf{z}_n^+, \hat{\mathbf{y}}_n) + \mathcal{L}_{\text{CE}}(\tilde{\mathbf{z}}_n^-, 1 - \hat{\mathbf{y}}_n) \quad = -\sum_n \sum_{l \in L} z_{n,l}^+ \log(\hat{\mathbf{y}}_{n,l}) - \sum_n \sum_{l \in L} \tilde{z}_{n,l}^- \log(1 - \hat{\mathbf{y}}_{n,l}). \tag{5.9}$$

Avec cette fonction de coût, la mise à jour des paramètres du réseau se fait par rétropropagation de l'erreur à la fois sur les variables prédites $\hat{\mathbf{y}}_{n,l}$ et les exemples négatifs estimés $\tilde{z}_{n,l}^-$. Cette modélisation fournit les scores des exemples négatifs $z_{n,k}^-$ avec une valeur dépendant de la similarité entre descripteurs des étiquettes observées $z_{n,l}^+ = 1$ et des non observées $z_{n,k}^+ = 0$.

5.6 Expérience

Dans cette section, premièrement, nous décrivons le cadre expérimental dans la section 5.6.1. Dans la section 5.6.2, nous proposons une validation empirique de la performance de notre architecture et de l'estimation des négatifs à partir des auto-similarités de représentations d'images conditionnées aux étiquettes. Enfin, nous réalisons des comparaisons par rapport aux autres méthodes de l'état de l'art et discutons des résultats obtenus dans la partie 5.6.3.

5.6.1 Conditions expérimentales

Corpus de données. Seul un faible nombre de jeux de données comporte des images annotées avec de multiples étiquettes. Cependant, les corpus traditionnellement utilisés pour la détection ou la segmentation peuvent être adaptés pour répondre au problème d'apprentissage multi-étiquettes. Avec de tels corpus, les boîtes englobantes et les masques de vérité terrain sont mis de côté et seule l'étiquette qui leur était associée est conservée. La distinction des instances est donc perdue, puisqu'une seule étiquette persiste dans l'annotation finale. Ainsi, $y_{n,l} = 1$ dans la vérité terrain signifie qu'au moins une instance de l'étiquette l est observée dans l'image I_n. Dans nos travaux, l'unique étiquette positive par image est obtenue par une sélection aléatoire uniforme parmi toutes les étiquettes de vérité terrain disponibles.

Nous considérons trois corpus de données : COCO (*Microsoft **C**ommon **O**bject in **CO**ntext*) [T.-Y. Lin, Maire et al., 2014], VOC (*Pascal **V**isual **O**bject **C**lasses*) [Everingham et al., 2010] et Deepfashion2 [Ge et al., 2019]. Ces corpus ont été présentés dans la section 2.3 du chapitre 2. Nous tenons à rappeler les éléments suivants. La version de 2014 de COCO contient 82 081 images pour l'entraînement des modèles et 40 137 images pour la validation. Le corpus est annoté avec 80 étiquettes différentes et les images en couleur ont une résolution pouvant aller

jusqu'à 500×500. La version de 2012 de VOC contient 5 717 images pour l'entraînement des modèles et 5 823 pour la validation. Les objets peuvent être de 20 classes différentes. Les images sont en couleur et peuvent atteindre une résolution de 640×640. Deepfashion2 contient 191 961 images pour l'entraînement et 32 153 images pour la validation. Les images sont annotées avec 13 étiquettes différentes et les images en couleur ont une résolution de 600×750 en moyenne.

Architecture. L'extraction des patchs a été effectuée à partir de $R = 3$ différents niveaux de résolution avec un facteur $d = 2$ entre chaque niveau. Les patchs sont des sous-images carrées de taille $w = h = 64$, et sont extraits avec un pas de déplacement de 64. Pour COCO,VOC et deepfashion2 cela correspond approximativement à 130 patchs par image.

Toutes les représentations e_p, e_l, $e_{I_n,l}$ sont définies dans un espace de dimension $F = 256$. Les patchs de taille 64×64 sont traités par un sous-réseau CNN. Cet extracteur de descripteurs est composé des cinq premiers blocs d'EfficientNet [Tan et Le, 2019] avec les mêmes hyper-paramètres que ceux utilisés par Tan et Le en 2019. Les deux dernières couches de notre réseau sont une couche de regroupement par moyenne (*average pooling*) et une couche entièrement connectée (*fully connected*) de taille $F = 256$. Cette dernière nous permet d'obtenir les représentations de patchs $e_p \in E_{P_n}$.

Les vecteurs de descripteurs des étiquettes de taille $|L| \times F$ sont appris par le modèle. Le nombre d'étiquettes est de $|L| = 80$ pour COCO, $|L| = 20$ pour VOC et $|L| = 13$ pour Deepfashion2. Le regroupement par attention est réalisé avec une attention croisée classique et un MLP à deux couches avec 256 neurones. Cette étape de regroupement fourni $|L|$ représentation d'image $e_{I_x,l}$ de taille $F = 256$. Tous les poids du modèle sont initialisés avec la méthode de mise à l'échelle par variance unitaire (*unit variance scaling*), les fonctions d'activation utilisées sont GELU pour les couches entièrement connectées et Swish pour les couches de convolution.

Les expérimentations ont été conduites sur une machine virtuelle de la plate-forme applicative en nuage de Microsoft Azure dotée d'une carte graphique Tesla P40. Notre modèle a été entraîné pendant 25 epochs avec des lots (*batch*) de 16 images. Le taux d'apprentissage est initialisé à $\text{lr} = 10^{-3}$ et est programmé pour décroitre avec un facteur 0.5 toutes les 5 epochs. Nous avons utilisé l'optimiseur LAMB [You et al., 2019] avec une déflation des poids avec un taux de 10^{-4}. Pour l'estimation des exemples négatifs d'étiquettes, nous fixons $\theta = 0$ pour obtenir les poids β dans l'équation (5.7), ce qui revient à une ReLU standard pour la normalisation des similarités.

Méthodes comparées. Nous utilisons comme référence le modèle entraîné en supervision complète (*c.-à-d.* toutes les étiquettes négatives et positives sont connues) avec la fonction $\mathcal{L}_{BCE}$ définie dans (5.5). Nous rappelons que tous les autres modèles sont entraînés en n'utilisant

qu'une unique étiquette positive par image.

Nous comparons notre modèle avec les rares méthodes de l'état de l'art qui abordent aussi le problème de la classification multi-étiquettes à partir d'un seul exemple positif connu. Ces méthodes ont été introduites par COLE et al. en 2021 et par VERELST et al. en 2022. L'évaluation des résultats est réalisée avec la moyenne des précisions moyennes (*mAP : **m**ean **A**verage **P**recision*), sur les mêmes versions des jeux de données de validation. Les modèles ont été entraînés avec les mêmes jeux de données d'apprentissage issus des corpus.

Nous soulignons que COLE et al. en 2021 présentent des résultats obtenus après un pré-apprentissage d'un ResNet-50 sur ImageNet [DENG et al., 2009] suivi d'un affinage (*fine tuning*) sur le problème d'apprentissage PU. Notre architecture étant entraînée de zéro, nous considérons le cadre expérimental de [VERELST et al., 2022] similaire au nôtre. Ainsi, pour nous comparer à [COLE et al., 2021], nous présentons les résultats publiés dans [VERELST et al., 2022], où un ResNet-50 a été entraîné avec différentes fonctions de coût pendant 100 epochs.

En pratique, nous fournissons une comparaison avec les modèles obtenus après 100 epochs avec les fonctions de coût $\mathcal{L}_{AN}$, $\mathcal{L}_{EPR}$, $\mathcal{L}_{ROLE}$ proposées par COLE et al. en 2021 et $\mathcal{L}_{EN+CL}$, $\mathcal{L}_{EN+SCL}$ proposées par VERELST et al. en 2022. $\mathcal{L}_{AN}$ est la fonction de coût des inconnus supposés négatifs (Assumed Negative), qui considère toutes les étiquettes non observées comme négatives. $\mathcal{L}_{EPR}$ est une régularisation sur les positifs attendus. Cette fonction applique une pénalisation pour que la somme des scores de prédiction soit proche de la moyenne des étiquettes positives par image sur la base d'entraînement. $\mathcal{L}_{ROLE}$ est une régularisation en ligne de l'estimation des étiquettes inconnues, qui repose sur la fonction $\mathcal{L}_{EPR}$ et estime en plus la vérité terrain des étiquettes non observées pour toutes les images comme des paramètres du modèle. $\mathcal{L}_{EN+CL}$ est une fonction de coût sur les négatifs attendus qui inclut une notion de cohérence spatiale et utilise des versions augmentées des images. Enfin, $\mathcal{L}_{EN+SCL}$ est une fonction de coût sur les négatifs attendus qui prédit des scores spatiaux de classification sur une carte de descripteurs d'image augmentée.

Lors de l'entraînement de notre architecture avec la fonction $\mathcal{L}_{EPR}$, la moyenne des étiquettes par image est définie à $k = 2.92$ pour COCO, $k = 1.38$ pour VOC et $k = 1.61$ pour Deepfashion2. Pour la fonction $\mathcal{L}_{ROLE}$, lr est multipliée par 10 pour la branche estimant la valeur des étiquettes non observées, comme recommandé dans [COLE et al., 2021].

5.6.2 Résultats et comparaisons

Nous présentons maintenant les expériences conduites pour valider l'architecture et la méthode d'estimation des exemples négatifs. Dans le tableau 5.1, nous fournissons les résultats obtenus avec notre architecture (partie haute du tableau) et un ResNet-50 (partie basse du

tableau). Pour les comparer équitablement, les deux architectures ont été entraînées à partir de zéro avec la fonction de coût $\mathcal{L}_{BCE}$. Cela fournit une borne supérieure aux résultats pouvant être obtenus par l'architecture. En effet, cette configuration correspond au cas « idéal » où tous les positifs et négatifs sont connus.

Modèle	Fonction de coût	COCO-14	VOC2012	Deepfashion2
architecture basée sur les patchs (notre proposition)	$\mathcal{L}_{\text{BCE}}$ (complètement annoté)	65.8	61.6	62.8
	$\mathcal{L}_{\text{AN}}$ ♣	62.6	60.0	**61.7**
	$\mathcal{L}_{\text{EPR}}$ ♣	61.4	58.8	57.9
	$\mathcal{L}_{\text{ROLE}}$ ♣	33.3	49.7	55.6
	$\mathcal{L}_{\text{WN}}$ (notre proposition)	**63.2**	**60.4**	61.1
Resnet-50 (rapporté par ♠)	$\mathcal{L}_{\text{BCE}}$ (complètement annoté)	64.8	53.4	-
	$\mathcal{L}_{\text{AN}}$ ♣	50.2	45.7	-
	$\mathcal{L}_{\text{ROLE}}$ ♣	51.9	45.0	-
	$\mathcal{L}_{\text{EN+CL}}$ ♠	54.3	47.0	-
	$\mathcal{L}_{\text{EN+SCL}}$ ♠	54.0	50.4	-

TABLE 5.1 – Résultats de mAP pour notre architecture entraînée pendant 25 epochs avec différentes fonctions de coût (partie haute) et comparaison (partie basse) avec les résultats rapportés par VERELST et al. en 2022 après entraînement d'un ResNet-50 pendant 100 epochs avec les fonctions de coût proposées dans ♣ [COLE et al., 2021] et ♠ [VERELST et al., 2022]. Les meilleurs résultats sont en gras.

Notre architecture est adaptée au problème d'apprentissage multi-étiquettes. Dans un contexte complètement supervisé, notre architecture atteint une mAP de 65.8 sur COCO, ce qui est supérieur au score de 64.8 rapporté par VERELST et al. en 2022 pour le ResNet-50. Sur la base VOC la différence en faveur de notre modélisation orientée patch est significative (61.6 contre 53.4).

Pour valider notre approche d'estimation des négatifs, nous présentons les résultats obtenus dans le contexte où une unique étiquette positive est connue par image. Nous rappelons que notre stratégie consiste à utiliser les auto-similarités entre les représentations d'image pour estimer des exemples négatifs qui sont fournis à la fonction de coût $\mathcal{L}_{WN}$. Comme illustré dans la tableau 5.1, pour le même nombre d'epochs, la mAP est proche de celle obtenue dans le contexte complètement supervisé (63.2 contre 65.8 pour COCO et 60.4 contre 61.6 pour VOC).

Nous avons aussi entraîné notre architecture basée sur les patchs avec les fonctions de coût concurrentes $\mathcal{L}_{AN}$, $\mathcal{L}_{EPR}$, $\mathcal{L}_{ROLE}$. Notre architecture entraînée avec notre proposition d'estimation des négatifs donne les meilleurs résultats sur les bases COCO et VOC. Avec la

fonction $\mathcal{L}_{AN}$, la somme des prédictions est toujours proche de 1, ce qui correspond davantage au cas du corpus VOC et Deepfashion2 (la moyenne des étiquettes positives par image étant $k = 1.38$ et $k = 1.61$ respectivement), qu'au corpus COCO ($k = 2.92$). Il est à noter que $\mathcal{L}_{ROLE}$ est moins performante avec les conditions d'entraînement considérées. Nous suggérons que cela est dû à la complexité du modèle, qui vise à estimer toutes les étiquettes non observées avec une branche dédiée dans la fonction de perte.

Pour être complet, nous reproduisons dans le tableau 5.1 les résultats en mAP rapportés par Verelst et al. en 2022, en entraînant un ResNet-50 à partir de zéro avec les fonctions de coût concurrentes. Contrairement à notre architecture, la baisse de la performance observée avec la mAP est significative entre le cas complètement supervisé et le cas où une seule étiquette positive est considérée.

L'ensemble de ces résultats indique que nos propositions sont adaptées au cas complexe d'apprentissage PU multi-étiquettes à partir d'une seule étiquette positive.

5.6.3 Discussion et conclusion

Charge de calcul. La charge de calcul nécessaire pour entraîner notre architecture est moindre par rapport à un ResNet-50 utilisé par Cole et al. en 2021 et Verelst et al. en 2022. Premièrement, la taille du modèle est réduite d'un facteur 100. Notre modèle contient approximativement $2,5.10^5$ paramètres à apprendre, là où un ResNet-50 est composé de $2,3.10^7$ paramètres. Avec notre architecture plus légère, de meilleurs résultats sont obtenus en seulement 25 epochs, au lieu de 100 pour le ResNet-50. Cela suggère que notre architecture généralise mieux le problème.

Hyper-paramètre. Notre fonction de coût d'estimation des négatifs n'inclut pas d'hyper-paramètres. À l'exception de ceux de l'architecture, le seul hyper-paramètre de notre modélisation globale est le seuil de la similarité cosinus (5.7) que nous fixons simplement à $\theta = 0$. D'autre part, les fonctions de coût $\mathcal{L}_{EPR}$ et $\mathcal{L}_{ROLE}$ pénalisent la somme des prédictions par rapport à la moyenne des étiquettes par image. Il s'agit d'une hypothèse forte, car la variance des étiquettes positives sur l'ensemble des données peut être importante. De plus, la connaissance préalable du nombre moyen d'étiquettes n'est souvent pas disponible dans les cas d'utilisations réelles. La fonction de coût $\mathcal{L}_{ROLE}$ repose également sur une estimation en ligne de toutes les annotations de vérité terrain non observées à partir des prédictions actuelles. Ce modèle est donc fortement influencé par l'initialisation et les premières epochs, tout en augmentant les besoins en mémoire.

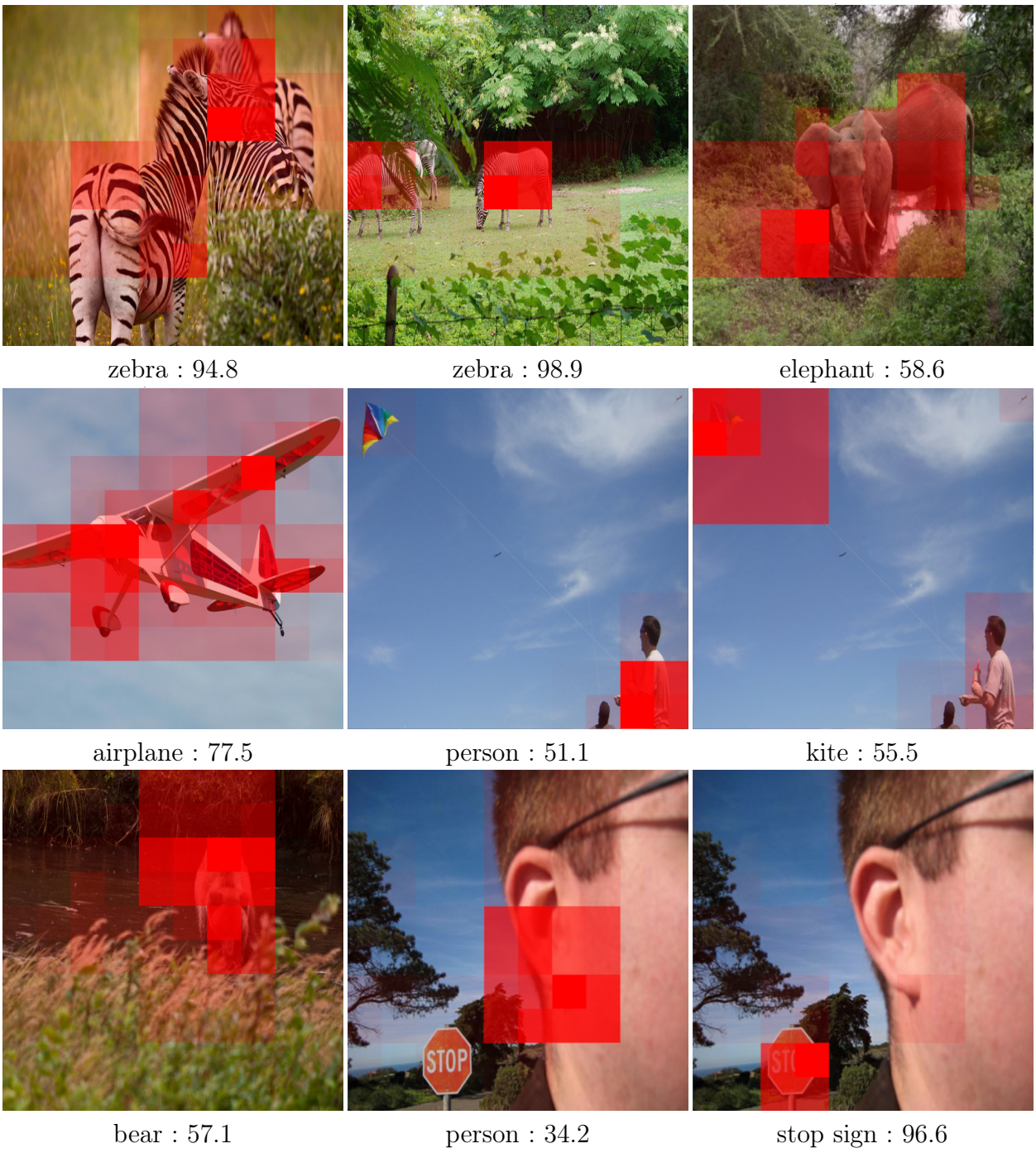


FIGURE 5.2 – Exemples de scores d'attention pour des étiquettes positives sur des images du corpus COCO. Les patchs sont remplis avec leurs valeurs de score d'attention, qui varient de 0 (transparent) à 1 (rouge). La deuxième ligne présente le score de prédiction pour les étiquettes données.

Localisation des étiquettes. Notre architecture a le potentiel de localiser les exemples de patchs correspondant à une étiquette détectée (Fig. 5.2). En effet, les scores d'attention calculés dans le regroupement des patchs par attention (bloc 4 de la *fig.* 5.1) permettent de déterminer

le niveau de participation des patchs à la décision de classification. Notre modèle basé sur les patchs offre donc un mécanisme intégré pour interpréter les résultats obtenus, sans s'appuyer sur des mécanismes avancés de rétropropagation de gradient [Selvaraju et al., 2017].

Pré-entraînement et raffinement. Il est important de souligner la limitation actuelle de notre approche par rapport aux architectures ResNet-50. Les méthodes proposées par Cole et al. en 2021 et par Verelst et al. en 2022 fournissent de bien meilleurs résultats en mAP (72 pour COCO et même 88 pour VOC), en considérant un ResNet-50 pré-entraîné sur Imagenet [Deng et al., 2009], avec des raffinements potentiels. Nous postulons que nos résultats pourraient également être améliorés en pré-entraînant soit notre architecture complète basée sur les patchs sur Imagenet, soit l'encodeur des patchs sur les boîtes englobantes d'un corpus de détection. Les performances pourraient également être améliorées en effectuant une recherche approfondie d'hyper-paramètres (dimension de l'espace de représentation F, nombre de couches d'attention, *etc.*) et en optimisant les paramètres de l'entraînement.

Figure 5.3 – Exemples de scores d'attention pour des images de t-shirts et de pantalons du corpus Deepfashion2.

Application aux vêtements. Notre architecture permet la classification multi-étiquettes de vêtement. Cependant, le problème semble un peu plus difficile pour notre stratégie d'entraînement (*tab.* 5.1). En effet, certains types de vêtement sont potentiellement trop similaires, ce qui complexifie l'estimation de négatifs, par exemple `short sleeved top` et `long sleeved top` où seules les manches diffèrent. Nous observons sur la première ligne d'images de la figure 5.3 que l'attention se porte principalement sur les manches et cols pour les distinguer. Une amélioration possible serait alors de chercher les similarités entre représentations au niveau lot (*batch*) lors de l'entraînement. Une représentation d'image conditionnée à une étiquette positive permettrait ainsi d'estimer les étiquettes positives et négatives dans les autres images du lot. Enfin, les scores d'attention ne fournissent pas directement les masques. Pour le cas des vêtements, les patchs ayant un haut score semblent corrélés à l'information nécessaire pour distinguer les étiquettes, par exemple les t-shirt cités plus haut, ou les pantalons avec des patchs contenant les deux jambes en même temps de la figure 5.3. Une méthode doit donc être conçue pour affiner la localisation et exploiter les scores d'attention.

Globalement, notre architecture et notre stratégie d'estimation d'exemples négatifs permettent de répondre à un problème complexe issu du domaine de l'apprentissage faiblement supervisé. Notre proposition est une preuve de concept de classification multi-étiquettes à partir d'une seule étiquette positive par image. C'est alors la contribution principale de cette thèse en apprentissage profond et en vision assistée par ordinateur.

Chapitre 6

Conclusion

Les travaux réalisés lors de cette thèse s'inscrivent dans un contexte industriel lié à la production de vêtements. Les différents acteurs de l'industrie textile et de la mode manipulent des images tout au long de la conception, du développement et de la commercialisation des vêtements. Décrire les vêtements contenus dans ces images automatiquement pourrait permettre : d'analyser le marché avec plus de sources et plus d'indicateurs, d'indexer les images dans des moteurs de recherche et de compléter l'information manquante dans des fiches produit.

Nous nous sommes appuyés sur des méthodes d'apprentissage issues de l'état de l'art et nous avons développé nos propres architectures de réseau de neurones à couche de convolution pour réaliser cette automatisation. Notre processus produisant des mots clefs décrivant des vêtements s'articule en deux étapes. La première localise chacun des vêtements d'une image. La deuxième prédit et extrait des caractéristiques plus fines à partir de ces localisations. Ce processus est évolutif en permettant l'ajout progressif de méthodes de caractérisation.

Pour localiser les vêtements, nous avons posé l'état de l'art des méthodes de localisation et retenons l'approche par segmentation d'instances. Nous avons sélectionné plusieurs modèles que nous appliquons au contexte de segmentation des vêtements [Jouanneau et al., 2020] en utilisant le corpus Deepfashion2 [Ge et al., 2019].

Ces expérimentations nous ont permis de soulever différentes limites de l'évaluation de segmentation d'instances avec la moyenne des précisions moyennes. Pour y répondre, nous avons proposé un protocole d'évaluation en trois axes [Jouanneau et al., 2021] : global, contour, contenu. Pour chacun de ces axes. Nous avons sélectionné ou proposé une métrique par axe : l'intersection sur l'union, la *boundary jaccard*, et une mesure de similarité obtenue à partir de la distance de transport optimal respectivement. Ces métriques nous ont servi à valider la qualité de masques prédits par différentes méthodes et ont pu être intégrées au calcul de la moyenne des précisions moyennes. Ce protocole nous a permis de retenir YOLACT [Bolya et al., 2019]

comme méthode de segmentation d'instances de vêtement.

Ce modèle nous permet d'obtenir des masques de vêtement. Nous pouvons alors utiliser ces masques pour caractériser finement les vêtements. Nous avons constitué un corpus à partir d'images et de texte issus de fiches produit et défini une taxonomie des étiquettes que nous souhaitions être en capacité de prédire. Ces étiquettes et les masques ont été attribués par une annotation automatique des images. Ce corpus nous a permis de développer deux méthodes pour prédire les étiquettes : du type de vêtement, du motif et de la couleur dominante du tissu. La première méthode repose sur une architecture de classification à partir d'une image, d'un masque et d'un ensemble de patchs. La deuxième se base sur l'extraction de la couleur et sa dénomination à partir du masque et d'une bibliothèque de références couleurs. Le processus global a pu être reversé à Lectra et il est en cours de test.

Enfin, ces travaux nous ont permis d'observer différentes problématiques liées à la constitution et l'annotation de corpus. Pour y répondre, nous avons proposé une approche faiblement supervisée qui est la contribution principale de cette thèse à la communauté scientifique [Jouanneau et al., 2023]. Elle repose sur une architecture légère de classification multi-étiquettes à partir de patchs. L'architecture produit différentes représentations de l'image selon l'étiquette. Ces représentations sont obtenues en combinant l'information de patchs par mécanisme d'attention. De plus, nous avons montré que les scores de contribution des patchs (*c.-à-d.* les poids de la combinaison linéaires dans l'attention) peuvent également être utilisés pour localiser les éléments au sein de l'image.

En complément de cette architecture, nous avons proposé une stratégie d'estimation des étiquettes négatives à partir de la similarité entre ces représentations d'images intégrée dans une fonction de coût. En se comparant à l'état de l'art, nous avons montré que nos propositions en terme d'architecture et de fonction de coût permettent, sur les 3 jeux de données testés, d'obtenir des résultats pertinents tout en réduisant la taille du nombre de paramètres d'un facteur 100. Enfin, notre proposition facilite l'étape d'annotation en ne nécessitant qu'une seule étiquette positive par image pour l'entraînement, permettant ainsi de s'abstraire de nombreuses contraintes liées aux données.

Futurs travaux

Dans cette section, nous listons les futurs travaux envisageables à l'issue de cette thèse.

Notre approche faiblement supervisée de classification multi-étiquettes est d'ores et déjà adaptable au contexte industriel d'identification et d'apposition de mots clefs décrivant des vêtements dans des images. Cette approche pourrait être intégrée aux processus global de différentes manières. Elle pourrait naturellement complémenter la deuxième étape de caractérisation

du contenu et permettre d'exploiter des données incomplètes (*ex.* multiples motifs non renseignés), de sources diverses (*ex.* réconciliation des taxonomies de Deepfashion2 et de Retviews), à la taxonomie plus complexe (*ex.* différentes granularités des types de vêtements dans la taxonomie de Retviews). Elle pourrait aussi complémenter ou remplacer l'étape de localisation. En effet, dans notre approche, les scores d'attention peuvent être interprétés comme un indicateur de présence d'un élément dans un patch. Cependant, une méthode efficace pour localiser les vêtements à partir de ces scores d'attention sera nécessaire. Ces localisations devront alors être validées par évaluation.

Dans les travaux présentés ici, nous nous sommes concentrés sur trois caractéristiques des vêtements : le type, le motif et la couleur dominante. Notre processus pourrait alors être enrichi avec d'autres types d'étiquettes. Ces étiquettes pourraient décrire d'autres caractéristiques comme le style (*ex.* Bohème-chic, *Sportswear*, *etc.*), la coupe (*ex. slim*, *regular*, *etc.*), l'occasion (*ex.* travail, *casual*, *etc.*) ou des sous éléments comme le type de col (*ex.* rond, en V,*etc.*), de manche (*ex.* bouffante, raglan,*etc.*), de poches (*ex.* passepoilée, à rabat,*etc.*). Nous pourrions aussi décrire les compositions de vêtement (*c.-à-d.* les tenues) sans doute plus appropriées pour le style. Les étiquettes pourront dépasser le cadre du vêtement en décrivant les modèles (*c.-à-d.* les porteurs de vêtements), les actions [H.-B. Zhang et al., 2019], différents éléments de la scène et constituer des graphes de scène [Chang et al., 2021]. Cela nous permettrait par exemple de préciser le contexte (défilé, conception, *in the wild*, *etc.*) et d'améliorer l'analyse et la recherche d'images.

Industriellement, utiliser des mots clefs est une approche très répandue [1] [2] [3] [4] à des fins d'analyse, d'indexation de contenu, de complétion d'information. Comme nous l'avons vu, ce problème se traduit scientifiquement en des problématiques de localisation et de classification. Cependant, il serait envisageable de se tourner vers des approches récentes qui exploitent directement les représentations des images et des éléments obtenus par réseau de neurones profond. Il serait alors possible d'associer textes et images avec des vecteurs de descripteurs (*embeddings)* contenus dans le même espace latent [Radford et al., 2021]. Dans cet espace latent, les vecteurs de texte et les vecteurs d'image seraient proches selon leurs sémantiques (*visual semantic embeddings* [Frome et al., 2013]).

Ainsi, la recherche peut se faire sur ces vecteurs de descripteurs sans indexation du contenu par mots clefs. Une recherche s'effectue en projetant une requête dans l'espace latent ce qui fourni les données plus proche voisin en résultats. L'approche par vecteur de descripteurs permet

1. heuritech.com/heuritech-suite
2. intelistyle.com/product-attribute-tagging
3. visenze.com/discovery-suite/modules/smart-tagging
4. wideeyes.ai/auto-tagging

d'adresser les problématiques suivantes : la recherche d'image similaire [Ji et al., 2020b], la recherche par dessin [L. Liu, Shen et al., 2017], les systèmes de recommandation [Y.-L. Lin et al., 2020], recherche à partir de texte [Cao et al., 2022], recherche itérative [H. Wu et al., 2021], la description automatique d'image [Karpathy et Fei-Fei, 2015]. De plus, analyser l'espace latent peut permettre par exemple de déterminer les modes et tendances et les suivre dans le temps de façon non supervisée [Al-Halah et Grauman, 2020b].

Enfin, pour les besoins de notre processus en deux étapes, nous avons obtenu un modèle de segmentation d'instances. Ce modèle fourni des masques de vêtement qui pourraient être affichés aux utilisateurs pour améliorer la compréhension et l'interprétation des mots clefs qui ont été retrouvés. Cet affichage permettrait potentiellement de mettre en place un protocole de corrections et d'enrichissement des données en s'appuyant sur les retours utilisateurs. Ces masques peuvent aussi servir à d'autres utilisations, par exemple pour transformer et générer des images. Les travaux envisageables seraient : de générer des *packshots* aux conditions standardisées contenant seulement un vêtement, de produire un patron simplifié d'un vêtement, de développer une solution d'essayage virtuel [Han et al., 2018] et d'éditer les vêtements et la tenue dans des images [A. Cui et al., 2021].

Bibliographie

Alamdar et Keyvanpour 2011 : Alamdar, Fatemeh et MohammadReza Keyvanpour (2011). « A new color feature extraction method based on QuadHistogram ». *Procedia Environmental Sciences* 10, p. 777-783 (cf. p. 75).

Arbelaez et al. 2010 : Arbelaez, Pablo, Michael Maire, Charless Fowlkes et Jitendra Malik (2010). « Contour detection and hierarchical image segmentation ». *IEEE transactions on pattern analysis and machine intelligence* 33.5, p. 898-916 (cf. p. 41, 42).

Bastan et al. 2020 : Bastan, Muhammet, Arnau Ramisa et Mehmet Tek (2020). « Cross-Modal Fashion Product Search with Transformer-Based Embeddings ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (cf. p. 32).

Bay et al. 2006 : Bay, Herbert, Tinne Tuytelaars et Luc Van Gool (2006). « Surf : Speeded up robust features ». In : *European conference on computer vision*. Springer, p. 404-417 (cf. p. 16).

Bekker et Davis 2020 : Bekker, Jessa et Jesse Davis (2020). « Learning from positive and unlabeled data : A survey ». *Machine Learning* 109.4, p. 719-760 (cf. p. 88, 90).

BERLIN 1969 : BERLIN, B (1969). « Basic Color Terms ». *Their Universality and Evolution* (cf. p. 73).

Bhattacharyya et Nag 2020 : Bhattacharyya, Mayukh et Sayan Nag (2020). « Hybrid Style Siamese Network : Incorporating style loss in complementary apparels retrieval ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (cf. p. 32).

Bianco et al. 2018 : Bianco, Simone, Remi Cadene, Luigi Celona et Paolo Napoletano (2018). « Benchmark analysis of representative deep neural network architectures ». *IEEE access* 6, p. 64270-64277 (cf. p. 63).

Bolya et al. 2019 : Bolya, Daniel, Chong Zhou, Fanyi Xiao et Yong Jae Lee (2019). « Yolact : Real-time instance segmentation ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 9157-9166 (cf. p. 20, 47, 60, 109).

Bossard et al. 2012 : Bossard, Lukas, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack et Luc Van Gool (2012). « Apparel classification with style ». In : *Asian Conference on Computer Vision*. Springer, p. 321-335 (cf. p. 57).

Buades et al. 2005 : Buades, Antoni, Bartomeu Coll et J-M Morel (2005). « A non-local algorithm for image denoising ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. T. 2, 60-65 vol. 2 (cf. p. 92).

Cai et Vasconcelos 2018 : Cai, Zhaowei et Nuno Vasconcelos (2018). « Cascade r-cnn : Delving into high quality object detection ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 6154-6162 (cf. p. 20, 27, 47).

Canny 1986 : Canny, John (1986). « A computational approach to edge detection ». *IEEE Transactions on pattern analysis and machine intelligence* 6, p. 679-698 (cf. p. 15).

Cao et al. 2022 : Cao, Min, Shiping Li, Juntao Li, Liqiang Nie et Min Zhang (2022). « Image-text Retrieval : A Survey on Recent Research and Development ». *arXiv preprint arXiv :2203.14713* (cf. p. 112).

Carbonneau et al. 2018 : Carbonneau, Marc-André, Veronika Cheplygina, Eric Granger et Ghyslain Gagnon (2018). « Multiple instance learning : A survey of problem characteristics and applications ». *Pattern Recognition* 77, p. 329-353 (cf. p. 94).

Chan et Vese 2001 : Chan, Tony F et Luminita A Vese (2001). « Active contours without edges ». *IEEE Transactions on image processing* 10.2, p. 266-277 (cf. p. 16).

Chang et al. 2021 : Chang, Xiaojun, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen et Alex Hauptmann (2021). « Scene graphs : A survey of generations and applications ». *arXiv preprint arXiv :2104.01111* (cf. p. 111).

C.-F. R. Chen et al. 2021 : Chen, Chun-Fu Richard, Quanfu Fan et Rameswar Panda (2021). « Crossvit : Cross-attention multi-scale vision transformer for image classification ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 357-366 (cf. p. 92).

H. Chen et al. 2020 : Chen, Hao, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang et Youliang Yan (2020). « Blendmask : Top-down meets bottom-up for instance segmentation ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 8573-8581 (cf. p. 20).

K. Chen, Pang et al. 2019 : Chen, Kai, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang et al. (2019). « Hybrid task cascade for instance segmentation ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 4974-4983 (cf. p. 20, 27, 47).

K. Chen, J. Wang et al. 2019 : Chen, Kai, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy et

Dahua Lin (2019). « MMDetection : Open MMLab Detection Toolbox and Benchmark ». *arXiv preprint arXiv :1906.07155* (cf. p. 27).

Y. Chen et al. 2018 : Chen, Yangyang, Dongping Ming, Lu Zhao, Beiru Lv, Keqi Zhou et Yuanzhao Qing (2018). « Review on high spatial resolution remote sensing image segmentation evaluation ». *Photogrammetric Engineering & Remote Sensing* 84.10, p. 629-646 (cf. p. 34).

B. Cheng et al. 2021 : Cheng, Bowen, Ross Girshick, Piotr Dollár, Alexander C Berg et Alexander Kirillov (2021). « Boundary IoU : Improving object-centric image segmentation evaluation ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 15334-15342 (cf. p. 38, 41, 42).

W.-H. Cheng et al. 2021 : Cheng, Wen-Huang, Sijie Song, Chieh-Yun Chen, Shintami Chusnul Hidayati et Jiaying Liu (2021). « Fashion meets computer vision : A survey ». *ACM Computing Surveys (CSUR)* 54.4, p. 1-41 (cf. p. 10).

Ciresan et al. 2012 : Ciresan, Dan, Alessandro Giusti, Luca Gambardella et Jürgen Schmidhuber (2012). « Deep neural networks segment neuronal membranes in electron microscopy images ». *Advances in Neural Information Processing Systems* 25 (cf. p. 16).

Cole et al. 2021 : Cole, Elijah, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris et Nebojsa Jojic (2021). « Multi-Label Learning from Single Positive Labels ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 933-942 (cf. p. 90, 99, 103-105, 107).

Comaniciu et Meer 2002 : Comaniciu, Dorin et Peter Meer (2002). « Mean shift : A robust approach toward feature space analysis ». *IEEE Transactions on pattern analysis and machine intelligence* 24.5, p. 603-619 (cf. p. 15).

Cordts et al. 2016 : Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth et Bernt Schiele (2016). « The cityscapes dataset for semantic urban scene understanding ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3213-3223 (cf. p. 21).

Coupé et al. 2011 : Coupé, Pierrick, José V. Manjón, Vladimir Fonov, Jens Pruessner, Montserrat Robles et D. Louis Collins (2011). « Patch-based segmentation using expert priors : Application to hippocampus and ventricle segmentation ». *NeuroImage* 54.2, p. 940-954 (cf. p. 92).

Csurka et al. 2013 : Csurka, Gabriela, Diane Larlus, Florent Perronnin et France Meylan (2013). « What is a good evaluation measure for semantic segmentation ? » In : *The British Machine Vision Conference*. T. 27 (cf. p. 37, 41, 42).

A. Cui et al. 2021 : Cui, Aiyu, Daniel McKee et Svetlana Lazebnik (2021). « Dressing in order : Recurrent person image generation for pose transfer, virtual try-on and outfit editing ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 14638-14647 (cf. p. 112).

Y. Cui et al. 2019 : Cui, Yin, Menglin Jia, Tsung-Yi Lin, Yang Song et Serge Belongie (2019). « Class-balanced loss based on effective number of samples ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 9268-9277 (cf. p. 64).

Dalal et Triggs 2005 : Dalal, Navneet et Bill Triggs (2005). « Histograms of oriented gradients for human detection ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. T. 1. Ieee, p. 886-893 (cf. p. 16, 17).

Delon et al. 2005 : Delon, Julie, Agnes Desolneux, Jose Luis Lisani et Ana Belen Petro (2005). « Automatic color palette ». In : *IEEE international conference on image processing 2005*. T. 2. IEEE, p. II-706 (cf. p. 73).

Deng et al. 2009 : Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li et Li Fei-Fei (2009). « Imagenet : A large-scale hierarchical image database ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. Ieee, p. 248-255 (cf. p. 10, 21, 65, 103, 107).

Dice 1945 : Dice, Lee R (1945). « Measures of the amount of ecologic association between species ». *Ecology* 26.3, p. 297-302 (cf. p. 37).

Dosovitskiy et al. 2020 : Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly et al. (2020). « An image is worth 16x16 words : Transformers for image recognition at scale ». *arXiv preprint arXiv :2010.11929* (cf. p. 92).

Efros et Leung 1999 : Efros, Alexei A et Thomas K Leung (1999). « Texture synthesis by non-parametric sampling ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*. T. 2, p. 1033-1038 (cf. p. 92).

Ester et al. 1996 : Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu et al. (1996). « A density-based algorithm for discovering clusters in large spatial databases with noise. » In : *kdd*. T. 96. 34, p. 226-231 (cf. p. 16).

Everingham et al. 2010 : Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn et Andrew Zisserman (2010). « The pascal visual object classes (voc) challenge ». *International journal of computer vision* 88.2, p. 303-338 (cf. p. 21, 101).

Fernandez-Moral et al. 2018 : Fernandez-Moral, Eduardo, Renato Martins, Denis Wolf et Patrick Rives (2018). « A new metric for evaluating semantic segmentation :

leveraging global and contour accuracy ». In : *Intelligent Vehicles Symposium* (cf. p. 38, 42).

Freeman et al. 2002 : Freeman, William T, Thouis R Jones et Egon C Pasztor (2002). « Example-based super-resolution ». *IEEE Computer Graphics and Applications* 22.2, p. 56-65 (cf. p. 92).

Frome et al. 2013 : Frome, Andrea, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato et Tomas Mikolov (2013). « Devise : A deep visual-semantic embedding model ». *Advances in Neural Information Processing Systems* 26 (cf. p. 111).

Fukunaga et Hostetler 1975 : Fukunaga, Keinosuke et Larry Hostetler (1975). « The estimation of the gradient of a density function, with applications in pattern recognition ». *IEEE Transactions on information theory* 21.1, p. 32-40 (cf. p. 15).

Fukushima 1975 : Fukushima, Kunihiko (1975). « Cognitron : A self-organizing multilayered neural network ». *Biological cybernetics* 20.3, p. 121-136 (cf. p. 10).

Gardner et al. 2017 : Gardner, Marc-André, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné et Jean-François Lalonde (2017). « Learning to predict indoor illumination from a single image ». *arXiv preprint arXiv :1704.00090* (cf. p. 84).

Ge et al. 2019 : Ge, Yuying, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang et Ping Luo (2019). « Deepfashion2 : A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 5337-5345 (cf. p. 24, 26, 27, 47, 101, 109).

Girshick 2015 : Girshick, Ross (2015). « Fast r-cnn ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 1440-1448 (cf. p. 18).

Girshick et al. 2014 : Girshick, Ross, Jeff Donahue, Trevor Darrell et Jitendra Malik (2014). « Rich feature hierarchies for accurate object detection and semantic segmentation ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 580-587 (cf. p. 18).

Goree et Crandall 2020 : Goree, Sam et David Crandall (2020). « Studying Empirical Color Harmony in Design ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (cf. p. 32).

Guo et al. 2019 : Guo, Sheng, Weilin Huang, Xiao Zhang, Prasanna Srikhanta, Yin Cui, Yuan Li, Hartwig Adam, Matthew R Scott et Serge Belongie (2019). « The imaterialist fashion attribute dataset ». In : *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (cf. p. 57).

Hadi Kiapour et al. 2015 : Hadi Kiapour, M, Xufeng Han, Svetlana Lazebnik, Alexander C Berg et Tamara L Berg (2015). « Where to buy it : Matching street clothing photos in online shops ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 3343-3351 (cf. p. 57, 59).

Al-Halah et Grauman 2020a : Al-Halah, Ziad et Kristen Grauman (2020a). « From paris to berlin : Discovering fashion style influences around the world ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 10136-10145 (cf. p. 73).

Al-Halah et Grauman 2020b : Al-Halah, Ziad et Kristen Grauman (2020b). « From paris to berlin : Discovering fashion style influences around the world ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 10136-10145 (cf. p. 112).

Al-Halah, Stiefelhagen et al. 2017 : Al-Halah, Ziad, Rainer Stiefelhagen et Kristen Grauman (2017). « Fashion forward : Forecasting visual style in fashion ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 388-397 (cf. p. 73).

Han et al. 2018 : Han, Xintong, Zuxuan Wu, Zhe Wu, Ruichi Yu et Larry S Davis (2018). « Viton : An image-based virtual try-on network ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 7543-7552 (cf. p. 85, 112).

He, Gkioxari et al. 2017 : He, Kaiming, Georgia Gkioxari, Piotr Dollár et Ross Girshick (2017). « Mask r-cnn ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 2961-2969 (cf. p. 19, 27, 47, 90).

He, X. Zhang et al. 2016 : He, Kaiming, Xiangyu Zhang, Shaoqing Ren et Jian Sun (2016). « Deep residual learning for image recognition ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 770-778 (cf. p. 27).

Heckbert 1982 : Heckbert, Paul (1982). « Color image quantization for frame buffer display ». *ACM Siggraph Computer Graphics* 16.3, p. 297-307 (cf. p. 15, 73).

Heer et Stone 2012 : Heer, Jeffrey et Maureen Stone (2012). « Color naming models for color selection, image editing and palette design ». In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 1007-1016 (cf. p. 73).

Hsiao et Grauman 2021 : Hsiao, Wei-Lin et Kristen Grauman (2021). « From culture to clothing : Discovering the world events behind a century of fashion images ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 1066-1075 (cf. p. 73).

X. Huang et Yan 2018 : Huang, Xiaolin et Ming Yan (2018). « Nonconvex penalties with analytical solutions for one-bit compressive sensing ». *Signal Processing* 144, p. 341-351 (cf. p. 90).

Z. Huang et al. 2019 : Huang, Zhaojin, Lichao Huang, Yongchao Gong, Chang Huang et Xinggang Wang (2019). « Mask scoring r-cnn ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 6409-6418 (cf. p. 19, 27, 47).

Hubert et Arabie 1985 : Hubert, Lawrence et Phipps Arabie (1985). « Comparing partitions ». *Journal of classification* 2.1, p. 193-218 (cf. p. 36).

Ilse et al. 2018a : Ilse, Maximilian, Jakub Tomczak et Max Welling (2018a). « Attention-based deep multiple instance learning ». In : *International Conference on Machine Learning*. PMLR, p. 2127-2136 (cf. p. 93).

Ilse et al. 2018b : Ilse, Maximilian, Jakub Tomczak et Max Welling (2018b). « Attention-based deep multiple instance learning ». In : *International Conference on Machine Learning*. PMLR, p. 2127-2136 (cf. p. 94).

Ishida et al. 2017 : Ishida, Takashi, Gang Niu, Weihua Hu et Masashi Sugiyama (2017). « Learning from complementary labels ». *Advances in neural information processing systems* 30 (cf. p. 90).

Jaccard 1912 : Jaccard, Paul (1912). « The distribution of the flora in the alpine zone. 1 ». *New phytologist* 11.2, p. 37-50 (cf. p. 37).

Jaegle et al. 2021 : Jaegle, Andrew, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman et Joao Carreira (2021). « Perceiver : General perception with iterative attention ». In : *International Conference on Machine Learning*. PMLR, p. 4651-4664 (cf. p. 93).

Ji et al. 2020a : Ji, Yang-Ho, HeeJae Jun, Insik Kim, Jongtack Kim, Youngjoon Kim, Byungsoo Ko, Hyong-Keun Kook, Jingeun Lee, Sangwon Lee et Sanghyuk Park (2020a). « An Effective Pipeline for a Real-world Clothes Retrieval System ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (cf. p. 32).

Ji et al. 2020b : Ji, Yang-Ho, HeeJae Jun, Insik Kim, Jongtack Kim, Youngjoon Kim, Byungsoo Ko, Hyong-Keun Kook, Jingeun Lee, Sangwon Lee et Sanghyuk Park (2020b). « An effective pipeline for a real-world clothes retrieval system ». *arXiv preprint arXiv :2005.12739* (cf. p. 112).

Jia et al. 2020 : Jia, Menglin, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam et Serge Belongie (2020). « Fashionpedia : Ontology, segmentation, and an attribute localization dataset ». In : *European Conf. on Computer Vision*. Springer, p. 316-332 (cf. p. 24).

Jouanneau et al. 2020 : Jouanneau, Warren, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2020). « Étude comparative de méthodologies issues de

Mask R-CNN : Application au Corpus DeepFashion2 ». In : *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP'20)*, p. 1-3 (cf. p. 11, 13, 30, 109).

Jouanneau et al. 2021 : Jouanneau, Warren, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2021). « Where are my clothes ? A multi-level approach for evaluating deep instance segmentation architectures on fashion images ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3951-3955 (cf. p. 12, 31, 52, 109).

Jouanneau et al. 2023 : Jouanneau, Warren, Aurélie Bugeau, Marc Palyart, Nicolas Papadakis et Laurent Vézard (2023). « A patch-based architecture for multi-label classification from single positive annotations ». In : *International Conference on Computer Vision Theory and Applications* (cf. p. 12, 87, 110).

Kanehira et Harada 2016 : Kanehira, Atsushi et Tatsuya Harada (2016). « Multi-label ranking from positive and unlabeled data ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 5138-5146 (cf. p. 90).

Karpathy et Fei-Fei 2015 : Karpathy, Andrej et Li Fei-Fei (2015). « Deep visual-semantic alignments for generating image descriptions ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3128-3137 (cf. p. 112).

Kass et al. 1988 : Kass, Michael, Andrew Witkin et Demetri Terzopoulos (1988). « Snakes : Active contour models ». *International journal of computer vision* 1.4, p. 321-331 (cf. p. 16).

Kirillov et al. 2019 : Kirillov, Alexander, Kaiming He, Ross Girshick, Carsten Rother et Piotr Dollár (2019). « Panoptic segmentation ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 9404-9413 (cf. p. 14).

Krizhevsky et al. 2017 : Krizhevsky, Alex, Ilya Sutskever et Geoffrey E Hinton (2017). « Imagenet classification with deep convolutional neural networks ». *Communications of the ACM* 60.6, p. 84-90 (cf. p. 10, 63).

Lanchantin et al. 2021 : Lanchantin, Jack, Tianlu Wang, Vicente Ordonez et Yanjun Qi (2021). « General multi-label image classification with transformers ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 16478-16488 (cf. p. 93).

LeCun et al. 1998 : LeCun, Yann, Léon Bottou, Yoshua Bengio et Patrick Haffner (1998). « Gradient-based learning applied to document recognition ». *Proceedings of the IEEE* 86.11, p. 2278-2324 (cf. p. 10, 63).

Lee et Park 2020 : Lee, Youngwan et Jongyoul Park (2020). « Centermask : Real-time anchor-free instance segmentation ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 13906-13915 (cf. p. 20).

M. Lin et al. 2013 : Lin, Min, Qiang Chen et Shuicheng Yan (2013). « Network in network ». *arXiv preprint arXiv :1312.4400* (cf. p. 93).

T.-Y. Lin, Dollár et al. 2017 : Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan et Serge Belongie (2017). « Feature pyramid networks for object detection ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 2117-2125 (cf. p. 27).

T.-Y. Lin, Maire et al. 2014 : Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár et C Lawrence Zitnick (2014). « Microsoft coco : Common objects in context ». In : *European Conf. on Computer Vision*. Springer, p. 740-755 (cf. p. 21, 22, 26, 39, 63, 101).

Y.-L. Lin et al. 2020 : Lin, Yen-Liang, Son Tran et Larry S Davis (2020). « Fashion outfit complementary item retrieval ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3311-3319 (cf. p. 112).

H. Liu et al. 2013 : Liu, Haoxue, Min Huang, Guihua Cui, M Ronnier Luo et Manuel Melgosa (2013). « Color-difference evaluation for digital images using a categorical judgment method ». *JOSA A* 30.4, p. 616-626 (cf. p. 78).

L. Liu, J. Chen et al. 2019 : Liu, Li, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa et Matti Pietikäinen (2019). « From BoW to CNN : Two decades of texture representation for texture classification ». *International Journal of Computer Vision* 127.1, p. 74-109 (cf. p. 92, 94).

L. Liu, Shen et al. 2017 : Liu, Li, Fumin Shen, Yuming Shen, Xianglong Liu et Ling Shao (2017). « Deep sketch hashing : Fast free-hand sketch-based image retrieval ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 2862-2871 (cf. p. 112).

W. Liu et al. 2016 : Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu et Alexander C Berg (2016). « Ssd : Single shot multibox detector ». In : *European Conf. on Computer Vision*. Springer, p. 21-37 (cf. p. 18).

Zhihao Liu et al. 2021 : Liu, Zhihao, Hui Yin, Xinyi Wu, Zhenyao Wu, Yang Mi et Song Wang (2021). « From shadow generation to shadow removal ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 4927-4936 (cf. p. 84).

Ziwei Liu et al. 2016 : Liu, Ziwei, Ping Luo, Shi Qiu, Xiaogang Wang et Xiaoou Tang (2016). « Deepfashion : Powering robust clothes recognition and retrieval with rich annotations ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 1096-1104 (cf. p. 24).

Lloyd 1982 : Lloyd, Stuart (1982). « Least squares quantization in PCM ». *IEEE transactions on information theory* 28.2, p. 129-137 (cf. p. 15).

Lowe 1999 : Lowe, David G (1999). « Object recognition from local scale-invariant features ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*. T. 2. Ieee, p. 1150-1157 (cf. p. 16, 17).

Mac Aodha et al. 2019 : Mac Aodha, Oisin, Elijah Cole et Pietro Perona (2019). « Presence-only geographical priors for fine-grained image classification ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 9596-9606 (cf. p. 90).

Malik et al. 2001 : Malik, Jitendra, Serge Belongie, Thomas Leung et Jianbo Shi (2001). « Contour and texture analysis for image segmentation ». *International journal of computer vision* 43.1, p. 7-27 (cf. p. 16).

Martin et al. 2004 : Martin, David R, Charless C Fowlkes et Jitendra Malik (2004). « Learning to detect natural image boundaries using local brightness, color, and texture cues ». *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26.5, p. 530-549 (cf. p. 37).

Melgosa et al. 2004 : Melgosa, Manuel, Rafael Huertas et Roy S Berns (2004). « Relative significance of the terms in the CIEDE2000 and CIE94 color-difference formulas ». *JOSA A* 21.12, p. 2269-2275 (cf. p. 78).

Minar et al. 2020 : Minar, MR, TT Tuan, H Ahn, P Rosin et YK Lai (2020). « Cp-vton+ : Clothing shape and texture preserving image-based virtual try-on ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (cf. p. 32, 85).

Otsu 1979 : Otsu, Nobuyuki (1979). « A threshold selection method from gray-level histograms ». *IEEE transactions on systems, man, and cybernetics* 9.1, p. 62-66 (cf. p. 15).

Papadakis et Rabin 2017 : Papadakis, Nicolas et Julien Rabin (2017). « Convex histogram-based joint image segmentation with regularized optimal transport cost ». *Journal of Mathematical Imaging and Vision* 59.2, p. 161-186 (cf. p. 45).

Pele et Werman 2009 : Pele, O. et M. Werman (2009). « Fast and robust Earth Mover's Distances ». In : *Proc. of the IEEE Int. Conf. on Computer Vision* (cf. p. 45).

Pitié et al. 2007 : Pitié, F., A. C. Kokaram et R. Dahyot (2007). « Automated colour grading using colour distribution transfer ». *Computer Vision and Image Understanding* 107.1-2, p. 123-137 (cf. p. 45).

Puzicha et al. 1999 : Puzicha, Jan, Joachim M Buhmann, Yossi Rubner et Carlo Tomasi (1999). « Empirical evaluation of dissimilarity measures for color and texture ». In : *Proc. of the IEEE Int. Conf. on Computer Vision* (cf. p. 45).

Qin, Dai et al. 2022 : Qin, Xuebin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao et al. (2022). « Highly Accurate Dichotomous Image Segmentation ». *arXiv preprint arXiv :2203.03041* (cf. p. 17).

Qin, Z. Zhang et al. 2020 : Qin, Xuebin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane et Martin Jagersand (2020). « U2-Net : Going deeper with nested U-structure for salient object detection ». *Pattern recognition* 106, p. 107404 (cf. p. 17).

Qu et al. 2017 : Qu, Liangqiong, Jiandong Tian, Shengfeng He, Yandong Tang et Rynson WH Lau (2017). « Deshadownet : A multi-context embedding deep network for shadow removal ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 4067-4075 (cf. p. 84).

Radford et al. 2021 : Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark et al. (2021). « Learning transferable visual models from natural language supervision ». In : *International Conference on Machine Learning*. PMLR, p. 8748-8763 (cf. p. 111).

Ramé et al. 2022 : Ramé, Alexandre, Arthur Douillard et Charles Ollion (2022). « CoRe : Color Regression for Multicolor Fashion Garments ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 2252-2257 (cf. p. 73).

Rand 1971 : Rand, William M (1971). « Objective criteria for the evaluation of clustering methods ». *Journal of the American Statistical association* 66.336, p. 846-850 (cf. p. 36).

Al-Rawi et Beel 2021 : Al-Rawi, Mohammed et Joeran Beel (2021). « Probabilistic Color Modelling of Clothing Items ». In : *Recommender Systems in Fashion and Retail*. Springer, p. 21-40 (cf. p. 76).

Read et al. 2009 : Read, Jesse, Bernhard Pfahringer, Geoff Holmes et Eibe Frank (2009). « Classifier chains for multi-label classification ». In : *Joint European conference on machine learning and knowledge discovery in databases*. Springer, p. 254-269 (cf. p. 90).

Reddi et al. 2020 : Reddi, Vijay Janapa, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou et al. (2020). « Mlperf inference benchmark ». In : *The International Symposium on Computer Architecture*. IEEE, p. 446-459 (cf. p. 63).

Redmon et al. 2016 : Redmon, Joseph, Santosh Divvala, Ross Girshick et Ali Farhadi (2016). « You only look once : Unified, real-time object detection ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 779-788 (cf. p. 18, 90).

Regier et al. 2007 : Regier, Terry, Paul Kay et Naveen Khetarpal (2007). « Color naming reflects optimal partitions of color space ». *Proceedings of the National Academy of Sciences* 104.4, p. 1436-1441 (cf. p. 73).

Reinke et al. 2021 : Reinke, Annika, Matthias Eisenmann, Minu D Tizabi, Carole H Sudre, Tim Rädsch, Michela Antonelli, Tal Arbel, Spyridon Bakas, M Jorge Cardoso, Veronika Cheplygina et al. (2021). « Common limitations of image processing metrics : A picture story ». *arXiv preprint arXiv :2104.05642* (cf. p. 40, 41).

Ren et al. 2015 : Ren, Shaoqing, Kaiming He, Ross Girshick et Jian Sun (2015). « Faster r-cnn : Towards real-time object detection with region proposal networks ». *Advances in neural information processing systems* 28 (cf. p. 18, 19, 90).

Robertson et al. 1977 : Robertson, AR, RD Lozano, DH Alman, SE Orchard, JA Keitch, R Connely, LA Graham, WL Acree, RS John, RF Hoban et al. (1977). « CIE recommendations on uniform color spaces, color-difference equations, and metric color terms ». *Color Res. Appl* 2, p. 5-6 (cf. p. 45, 74).

Ronneberger et al. 2015 : Ronneberger, Olaf, Philipp Fischer et Thomas Brox (2015). « U-net : Convolutional networks for biomedical image segmentation ». In : *International Conference on Medical image computing and computer-assisted intervention*. Springer, p. 234-241 (cf. p. 16).

Rubner et al. 1998 : Rubner, Y., C. Tomasi et L.J. Guibas (1998). « A metric for distributions with applications to image databases ». In : *Proc. of the IEEE Int. Conf. on Computer Vision* (cf. p. 45).

Russakoff et al. 2004 : Russakoff, Daniel B, Carlo Tomasi, Torsten Rohlfing et Calvin R Maurer (2004). « Image similarity using mutual information of regions ». In : *European Conf. on Computer Vision*. Springer, p. 596-607 (cf. p. 36).

Schroff et al. 2015 : Schroff, Florian, Dmitry Kalenichenko et James Philbin (2015). « Facenet : A unified embedding for face recognition and clustering ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 815-823 (cf. p. 85).

Selvaraju et al. 2017 : Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh et Dhruv Batra (2017). « Grad-cam : Visual explanations from deep networks via gradient-based localization ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 618-626 (cf. p. 107).

Sharma et al. 2005 : Sharma, Gaurav, Wencheng Wu et Edul N Dalal (2005). « The CIEDE2000 color-difference formula : Implementation notes, supplementary test data, and mathematical observations ». *Color Research & Application : Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science*

Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 30.1, p. 21-30 (cf. p. 78).

Jianbo Shi et Malik 2000 : Shi, Jianbo et Jitendra Malik (2000). « Normalized cuts and image segmentation ». *IEEE Transactions on pattern analysis and machine intelligence* 22.8, p. 888-905 (cf. p. 16).

Song et Funkhouser 2019 : Song, Shuran et Thomas Funkhouser (2019). « Neural illumination : Lighting prediction for indoor environments ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 6918-6926 (cf. p. 84).

Szeliski et al. 2007 : Szeliski, Richard et al. (2007). « Image alignment and stitching : A tutorial ». *Foundations and Trends® in Computer Graphics and Vision* 2.1, p. 1-104 (cf. p. 85).

Taha et Hanbury 2015 : Taha, Abdel Aziz et Allan Hanbury (2015). « Metrics for evaluating 3D medical image segmentation : analysis, selection, and tool ». *BMC medical imaging* 15.1, p. 1-28 (cf. p. 36, 42).

Tan et Le 2019 : Tan, Mingxing et Quoc Le (2019). « Efficientnet : Rethinking model scaling for convolutional neural networks ». In : *International Conference on Machine Learning*. PMLR, p. 6105-6114 (cf. p. 63, 65, 68, 85, 96, 102).

Trockman et Kolter 2022 : Trockman, Asher et J Zico Kolter (2022). « Patches Are All You Need ? » *arXiv preprint arXiv :2201.09792* (cf. p. 92).

Tversky 1977 : Tversky, Amos (1977). « Features of similarity. » *Psychological review* 84.4, p. 327 (cf. p. 42).

Unnikrishnan et Hebert 2005 : Unnikrishnan, Ranjith et Martial Hebert (2005). « Measures of similarity ». In : *Winter Conference on Applications of Computer Vision*. T. 1. IEEE, p. 394-394 (cf. p. 36).

Varma et Zisserman 2008 : Varma, Manik et Andrew Zisserman (2008). « A statistical approach to material classification using image patch exemplars ». *IEEE Trans. on Pattern Analysis and Machine Intelligence* 31.11, p. 2032-2047 (cf. p. 92).

Vaswani et al. 2017 : Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser et Illia Polosukhin (2017). « Attention is all you need ». *Advances in Neural Information Processing Systems* 30 (cf. p. 92, 93, 96).

Verelst et al. 2022 : Verelst, Thomas, Paul K Rubenstein, Marcin Eichner, Tinne Tuytelaars et Maxim Berman (2022). « Spatial Consistency Loss for Training Multi-Label Classifiers from Single-Label Annotations ». *arXiv preprint arXiv :2203.06127* (cf. p. 90, 103-105, 107).

VIOLA et JONES 2001 : VIOLA, Paul et Michael JONES (2001). « Rapid object detection using a boosted cascade of simple features ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. T. 1. Ieee, p. I-I (cf. p. 17).

VIOLA et WELLS III 1997 : VIOLA, Paul et William M WELLS III (1997). « Alignment by maximization of mutual information ». *International journal of computer vision* 24.2, p. 137-154 (cf. p. 36).

C.-Y. WANG et al. 2022 : WANG, Chien-Yao, Alexey BOCHKOVSKIY et Hong-Yuan Mark LIAO (2022). « YOLOv7 : Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors ». *arXiv preprint arXiv :2207.02696* (cf. p. 19).

G. WANG et al. 2022 : WANG, Guangcong, Yinuo YANG, Chen Change LOY et Ziwei LIU (2022). « StyleLight : HDR Panorama Generation for Lighting Estimation and Editing ». In : *European Conf. on Computer Vision*. Springer, p. 477-492 (cf. p. 84).

Q. WANG et Z. WANG 2009 : WANG, Qi et Zengfu WANG (2009). « A subjective method for image segmentation evaluation ». In : *Asian Conference on Computer Vision*. Springer, p. 53-64 (cf. p. 34).

X. WANG et al. 2020 : WANG, Xinlong, Tao KONG, Chunhua SHEN, Yuning JIANG et Lei LI (2020). « Solo : Segmenting objects by locations ». In : *European Conf. on Computer Vision*. Springer, p. 649-665 (cf. p. 20).

WEI et al. 2014 : WEI, Yunchao, Wei XIA, Junshi HUANG, Bingbing NI, Jian DONG, Yao ZHAO et Shuicheng YAN (2014). « CNN : Single-label to multi-label ». *arXiv preprint arXiv :1406.5726* (cf. p. 90).

H. WU et al. 2021 : WU, Hui, Yupeng GAO, Xiaoxiao GUO, Ziad AL-HALAH, Steven RENNIE, Kristen GRAUMAN et Rogerio FERIS (2021). « Fashion iq : A new dataset towards retrieving images by natural language feedback ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 11307-11317 (cf. p. 112).

XIA et al. 2018 : XIA, Gui-Song, Xiang BAI, Jian DING, Zhen ZHU, Serge BELONGIE, Jiebo LUO, Mihai DATCU, Marcello PELILLO et Liangpei ZHANG (2018). « DOTA : A large-scale dataset for object detection in aerial images ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3974-3983 (cf. p. 21).

Tete XIAO et al. 2021 : XIAO, Tete, Mannat SINGH, Eric MINTUN, Trevor DARRELL, Piotr DOLLÁR et Ross GIRSHICK (2021). « Early convolutions help transformers see better ». *Advances in Neural Information Processing Systems* 34, p. 30392-30400 (cf. p. 92).

Tong XIAO et al. 2015 : XIAO, Tong, Tian XIA, Yi YANG, Chang HUANG et Xiaogang WANG (2015). « Learning from massive noisy labeled data for image classification ». In : *Proc. of*

the IEEE/CVF Conf. on Computer Vision and Pattern Recognition, p. 2691-2699 (cf. p. 57, 59).

Yamaguchi et al. 2013 : Yamaguchi, Kota, M Hadi Kiapour et Tamara L Berg (2013). « Paper doll parsing : Retrieving similar styles to parse clothing items ». In : *Proc. of the IEEE Int. Conf. on Computer Vision*, p. 3519-3526 (cf. p. 24).

Yazici et al. 2018 : Yazici, Vacit Oguz, Joost van de Weijer et Arnau Ramisa (2018). « Color naming for multi-color fashion items ». In : *World Conference on Information Systems and Technologies*. Springer, p. 64-73 (cf. p. 73).

Ye et al. 2003 : Ye, Qixiang, Wen Gao et Wei Zeng (2003). « Color image segmentation using density-based clustering ». In : *International Conference on Multimedia and Exposition*. T. 2. IEEE, p. II-401 (cf. p. 16).

You et al. 2019 : You, Yang, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer et Cho-Jui Hsieh (2019). « Large batch optimization for deep learning : Training bert in 76 minutes ». *arXiv preprint arXiv :1904.00962* (cf. p. 102).

Yu et al. 2020 : Yu, Fisher, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan et Trevor Darrell (2020). « Bdd100k : A diverse driving dataset for heterogeneous multitask learning ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 2636-2645 (cf. p. 21).

Zaheer et al. 2017 : Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov et Alexander J Smola (2017). « Deep sets ». *Advances in Neural Information Processing Systems* 30 (cf. p. 93).

Zeiler et Fergus 2014 : Zeiler, Matthew D et Rob Fergus (2014). « Visualizing and understanding convolutional networks ». In : *European conference on computer vision*. Springer, p. 818-833 (cf. p. 16).

Zhan et al. 2020 : Zhan, Xiaohang, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin et Chen Change Loy (2020). « Self-supervised scene de-occlusion ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 3784-3792 (cf. p. 85).

D. Zhang et al. 2020 : Zhang, Dong, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua et Qianru Sun (2020). « Feature pyramid transformer ». In : *European Conf. on Computer Vision*. Springer, p. 323-339 (cf. p. 92).

H.-B. Zhang et al. 2019 : Zhang, Hong-Bo, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du et Duan-Sheng Chen (2019). « A comprehensive survey of vision-based human action recognition methods ». *Sensors* 19.5, p. 1005 (cf. p. 111).

H. Zhang et al. 2008 : Zhang, Hui, Jason E Fritts et Sally A Goldman (2008). « Image segmentation evaluation : A survey of unsupervised methods ». *computer vision and image understanding* 110.2, p. 260-280 (cf. p. 33, 35, 36).

Y. J. Zhang 1996 : Zhang, Yu Jin (1996). « A survey on evaluation methods for image segmentation ». *Pattern recognition* 29.8, p. 1335-1346 (cf. p. 33, 35).

Zhao et al. 2022 : Zhao, Tianchen, Niansong Zhang, Xuefei Ning, He Wang, Li Yi et Yu Wang (2022). « CodedVTR : Codebook-based Sparse Voxel Transformer with Geometric Guidance ». In : *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, p. 1435-1444 (cf. p. 93).

Zheng et al. 2018 : Zheng, Shuai, Fan Yang, M Hadi Kiapour et Robinson Piramuthu (2018). « Modanet : A large-scale street fashion dataset with polygon annotations ». In : *ACM International Conference. on Multimedia*, p. 1670-1678 (cf. p. 24).

Zhou et al. 2018 : Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh et Jianming Liang (2018). « Unet++ : A nested u-net architecture for medical image segmentation ». In : *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, p. 3-11 (cf. p. 17).