Fine-grained action detection and classification from videos with spatio-temporal convolutional neural networks : Application to Table Tennis.

Pierre-Etienne Martin

Thèse

Fine-Grained Action Detection and Classification from Videos with Spatio-Temporal Convolutional Neural Networks. Application to Table Tennis.

présentée à

l'Université de Bordeaux

École Doctorale de Mathématiques et d'Informatique

par

Pierre-Etienne MARTIN

pour obtenir le grade de

DOCTEUR

Spécialité : Informatique

Date de soutenance : 18 décembre 2020

Devant la commission d'examen composée de :

Petia	Radeva	Pr.	Universitat de Barcelona	Rapporteure
Klaus	Schöffmann	Dr. HDR	Klagenfurt University	Rapporteur
Nicolas	Thome	Pr.	CNAM Paris	Rapporteur
Martha	Larson	Pr.	Radboud University	Examinatrice
Pascal	Desbarats	Pr.	LaBRI, Université de Bordeaux	Examinateur
Jenny	Benois-Pineau	Pr.	LaBRI, Université de Bordeaux	Co-directeur de thèse
Renaud	Péteri	MCF-HDR	MIA, La Rochelle Université	Co-directeur de thèse
Julien	Morlier	MCF-HDR	IMS, Université de Bordeaux	Invité
Stefan	Posch	Pr.	University Halle Wittenberg	Invité

Le jury est présidé par : Pr .Pascal Desbarats

– 2020 –

Acknowledgements

I would like to thank my supervisors, Jenny Benois-Pineau and Renaud Péteri, who have worked on my side all along and provided quality advises, encouragements and support during those three years.

Special thanks to Julien Morlier, director of the Sports Faculty - STAPS - of the University of Bordeaux. Without him this project could not have been brought into being; and to Alain Coupet, his students and contributors in genereal: Daniel Canelle, Anthony Espadinha, Pablo Cluzaud, Céline Castanier, Bastien Levesque, Gaétan Cany, Mathieu Ballon, Xalbat Dirassar, Gabin Baracand, Guillaume Dogon, Benjamin Horreau, Jérémie Ruelle, Maxime Regodesebes, Alexandre Cifuentes, Axelle Perraud, Claire Maugein, Dylan Pereira, Fabien Lacotte, Ivan Perromat, Louis Pouthier, Matthieu Dubos, Maxence Schirrecker, Mehdi, Thomas Nouhaud, Remi Betelu, Nicolas CASCAILH, Mano Bayart, Clement Vedel, Jiangzhou XIA; for spending so much free time and energy to constitute the `TTStroke-21` database.

Thank you to the rapporteurs and reviewers constituting my jury: Pr. Petia Radeva, Assoc. Prof. Dr. Klaus Schöffmann, Pr. Nicolas Thome, Pr. Martha Larson and Pr. Pascal Desbarats; who agreed to spend time and energy to evaluate this work.

Thanks also to my colleagues and friends from the AIV team: Boris Mansencal, Abraham Montoya Obeso, Karim Aderghal, Attila Fejer, Thinhinane Yebda who have been present on my daily routine in LaBRI and provided help, advises and mature reflections when needed.

I would like to acknowledge Kazi Asif Ahmed Fuad who did his IPCV Master internship with me and gave a deeper dimension to this project.

Great thanks to the organization team of the sport task in MediaEval since 2019: Jenny Benois-Pineau, Renaud Péteri, Boris Mansencal, Jordan Calandre, Julien Morlier, Laurent Mascarilla.

Thank you also to Pascal Desbarats, Laura Tejada Pascual and Fanny Garat for their work and their help on the development of the Alumni network of the IPCV master; to Aurélie Bugeau for her dedication to the EcoLabri commission; to Akka Zemmari for his advises on the deep learning in computer vision topic and Andreas Hartmann and Sylvaine Granier from the mathematics and computer science doctoral school EDMI who followed me up ever since I started my PhD.

I also thank Pr. Stefan Posch to have welcomed me in Halle, Germany, to conduct research within his institute which gave me the opportunity to widen my scientific network while learning a new language.

Big thanks to my friends, colleagues, co-workers from the LaBRI with whom I work with, drink coffee with, present with and made my PhD experience rewarding.

Special thanks to the LaBRI direction and administration who provide a qualitative work place to perform research and were always here to answer queries and provide help: Jean Philippe Domenger, Xavier Blanc, Cathy Roubineau, Isabelle Garcia, Sylvie Le Laurain, Maité Labrousse, Auriane Dantes, Emmanuelle Lesage,

Laura Lorto, Magali Hinnenberger, Elia Meyre, Francine Krief, Cyril Gavoille, Mohamed Mosbah, Marc Zeitoun, Katel Guerin and Pascal Ung.

I also would like to thank all members and friends from the AFoDIB association, with whom we organized countless events in the LaBRI or Bordeaux for better life experience when being a Computer Science PhD student.

I shall not forget to thank the members of my family and all my friends not from the computer science field who have, for three years, be patient with me, gave me reassurance and always faked to understand the details of my topic.

Grateful tanks to Florian Delaplace, my mother and my sister for proofreading my thesis and correcting the many orthographic errors.

At last but not least, great thanks to Julia Jacob who always had an encouraging smile, a sympathetic ear, a generous heart and a trained eye to spot my spelling mistakes.

Title

Fine-Grained Action Detection and Classification from Videos with Spatio-Temporal Convolutional Neural Networks. Application to Table Tennis.

Abstract

Action recognition in videos is one of the key problems in visual data interpretation. Despite intensive research, differencing and recognizing similar actions remains a challenge. This thesis deals with fine-grained classification of sport gestures from videos, with an application to table tennis. In this manuscript, we propose a method based on deep learning for automatically segmenting and classifying table tennis strokes in videos. Our aim is to design a smart system for students and teachers for analyzing their performances. By profiling the players, a teacher can therefore tailor the training sessions more efficiently in order to improve their skills. Players can also have an instant feedback on their performances.

For developing such a system with fine-grained classification, a very specific dataset is needed to supervise the learning process. To that aim, we built the "`TTStroke-21`" dataset, which is composed of 20 stroke classes plus a rejection class. The `TTStroke-21` dataset comprises video clips of recorded table tennis exercises performed by students at the sport faculty of the University of Bordeaux - STAPS. These recorded sessions were annotated by professional players or teachers using a crowdsourced annotation platform. The annotations consist in a description of the handedness of the player and information for each stroke performed (starting and ending frames, class of the stroke). Fine-grained action recognition has some notable differences with coarse-grained action recognition. In general, datasets used for coarse-grained action recognition, the background context often provides discriminative information that methods can use to classify the action, rather than focusing on the action itself. In fine-grained classification, where the inter-class similarity is high, discriminative visual features are harder to extract and the motion plays a key role for characterizing an action.

In this thesis, we introduce a Twin Spatio-Temporal Convolutional Neural Network. This deep learning network takes as inputs an RGB image sequence and its computed Optical Flow. The RGB image sequence allows our model to capture appearance features while the optical flow captures motion features. Those two streams are processed in parallel using 3D convolutions, and fused at the last stage of the network. Spatio-temporal features extracted in the network allow efficient classification of video clips from `TTStroke-21`. Our method gets an average classification performance of 87.3% with a best run of 93.2% accuracy on the test set. When applied on joint detection and classification task, the proposed method reaches an accuracy of 82.6%.

A systematic study of the influence of each stream and fusion types on classification accuracy has been performed, giving clues on how to obtain the best performances. A comparison of different optical flow methods and the role of their

normalization on the classification score is also done. The extracted features are also analyzed by back-tracing strong features from the last convolutional layer to understand the decision path of the trained model. Finally, we introduce an attention mechanism to help the model focusing on particular characteristic features and also to speed up the training process. For comparison purposes, we provide performances of other methods on `TTStroke-21` and test our model on other datasets. We notice that models performing well on coarse-grained action datasets do not always perform well on our fine-grained action dataset.

The research presented in this manuscript was validated with publications in one international journal, five international conference papers, two international workshop papers and a reconductible task in MediaEval workshop in which participants can apply their action recognition methods to `TTStroke-21`. Two additional international workshop papers are in process along with one book chapter.

Keywords

Deep Learning, Action classification, Spatio-temporal convolution, Table tennis, Optical Flow, Computer Vision, Video indexing

Location

University of Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
University of La Rochelle, Mathématiques, Image et Applications - MIA, La Rochelle, France

Titre

Détection et classification fines d'actions à partir de vidéos par réseaux de neurones à convolutions spatio-temporelles. Application au tennis de table.

Résumé

La reconnaissance des actions à partir de vidéos est l'un des principaux problèmes de vision par ordinateur. Malgré des recherches intensives, la différenciation et la reconnaissance d'actions similaires restent un défi. Cette thèse porte sur la classification des gestes sportifs à partir de vidéos, avec comme cadre applicatif le tennis de table.

Nous proposons une méthode d'apprentissage profond pour segmenter et classifier automatiquement les différents coup de Tennis de Table. Notre objectif est de concevoir un système intelligent permettant d'analyser les performances des élèves pongistes, et de donner la possibilité à l'entraîneur d'adapter ses séances d'entraînement pour améliorer leurs performances.

Dans ce but, nous avons élaboré la base de données "`TTStroke-21`", constituée de clips vidéo d'exercices de tennis de table, enregistrés par les étudiants de la faculté de sport de l'Université de Bordeaux – STAPS. Cette base de données a ensuite été annotée par des professionnels du domaine à l'aide d'une plateforme crowdsourcing. Les annotations consistent en une description des coups effectués (début, fin et type de coup). Au total, 20 différents coups de tennis de table sont considérés plus une classe de rejet.

La reconnaissance des actions similaires présente des différences avec la reconnaissance d'actions classique. En effet, dans les bases de données classiques, le contexte de l'arrière plan fournit souvent des informations discriminantes que les méthodes peuvent utiliser pour classer l'action plutôt que de se concentrer sur l'action elle-même. Dans notre cas, la similarité entre classes est élevée, les caractéristiques visuelles discriminantes sont donc plus difficiles à extraire et le mouvement joue un rôle clef dans la caractérisation de l'action.

Dans cette thèse, nous introduisons un réseau de neurones spatio-temporel convolutif avec une architecture Jumelle. Ce réseau d'apprentissage profond prend comme entrées une séquence d'images RVB et son flot optique estimé. Les données RVB permettent à notre modèle de capturer les caractéristiques d'apparence tandis que le flot optique capture les caractéristiques de mouvement. Ces deux flux sont traités en parallèle à l'aide de convolutions 3D, et sont fusionnés à la dernière étape du réseau. Les caractéristiques spatio-temporelles extraites dans le réseau permettent une classification efficace des clips vidéo de `TTStroke-21`. Notre méthode obtient une performance de classification de 93.2% sur l'ensemble des données tests. Appliquée à la tâche jointe de détection et de classification, notre méthode atteint une précision de 82.6%.

Nous étudions les performances en fonction des types de données utilisés en entrée et la manière de les fusionner. Différents estimateurs de flot optique ainsi que leur

normalisation sont testés afin d'améliorer la précision. Les caractéristiques de chaque branche de notre architecture sont également analysées afin de comprendre le chemin de décision de notre modèle. Enfin, nous introduisons un mécanisme d'attention pour aider le modèle à se concentrer sur des caractéristiques discriminantes et aussi pour accélérer le processus d'entraînement. Nous comparons notre modèle avec d'autres méthodes sur `TTStroke-21` et le testons sur d'autres ensembles de données. Nous constatons que les modèles fonctionnant bien sur des bases de données d'actions classiques ne fonctionnent pas toujours aussi bien sur notre base de données d'actions similaires.

Les travaux présentés dans cette thèse ont été validés par des publications dans une revue internationale, cinq papiers de conférences internationales, deux papiers d'un workshop international et une tâche reconductible dans le workshop MediaEval où les participants peuvent appliquer leurs méthodes de reconnaissance d'actions à notre base de données `TTStroke-21`. Deux autres papiers de workshop internationaux sont en cours de préparation, ainsi qu'un chapitre de livre.

Mots-clés

Apprentissage profond, Classification d'actions, Tennis de table, Convolutions Spatio-temporelles, Indexation vidéo, Flot optique, Vision par ordinateur

Adresse

Université de Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
Université de La Rochelle, MIA, La Rochelle, France

Synthèse des travaux en français

Résumé

Ces travaux de thèse portent sur la reconnaissance des gestes sportifs à partir de vidéos et sont appliqués au cas du tennis de table. Le but est de concevoir un environnement informatique intelligent sur lequel étudiants et enseignants peuvent analyser la façon de jouer des sportifs. Le méthode développée permet de segmenter et de classifier automatiquement les coups de tennis de table effectués par les joueurs à partir de vidéos. Ainsi le profil des joueurs peut être renseigné et l'enseignant peut adapter son cours pour améliorer au mieux leurs performances.

Pour ce faire, nous avons enregistré des jeux de tennis de table avec des étudiants en STAPS. Ces enregistrements ont ensuite été annotés temporellement par des professionnels sur une plate-forme participative d'annotation. Cette nouvelle base de données, nommée "`TTStroke-21`", nous a permis d'entraîner et de tester notre méthode de classification.

Nous avons introduit un réseau de neurones jumeau à convolutions spatio-temporelles prenant en entrée le flux vidéo et le flot optique estimé sur la séquence. Traitées parallèlement, ces données permettent une classification efficace des segments vidéo. À partir de ces classifications, les frontières temporelles des coups effectués et leur classe peuvent être estimées.

1 Introduction

L'objectif de ces travaux est la reconnaissance d'actions sportives à partir de vidéos dans le but d'améliorer les performances des athlètes. Notre cadre applicatif est le tennis de table. Nous présentons une nouvelle base de données, "`TTStroke-21`", qui comporte vingt classes de coups de tennis de table et une classe de rejet supplémentaire. Cette taxonomie a été conçue avec des professionnels du domaine afin de recouvrir toutes les variations des actions de ce sport. Nous travaillons sur des vidéos enregistrées à la Faculté des Sports de l'Université de Bordeaux - STAPS. Les étudiants sont les athlètes filmés et les professeurs supervisent les exercices effectués lors des séances d'enregistrement. Les enregistrements sont sans capteurs, ce qui permet aux joueurs de jouer dans des conditions naturelles. L'objectif est de développer un outil d'analyse automatique que les enseignants et les étudiants pourraient utiliser pour analyser les matchs des joueurs de tennis de table. Ainsi, avec un retour automatisé de leur performance, les séances d'entraînement peuvent être adaptées plus facilement en fonction de leurs besoins. Cet ensemble de données constitue la première contribution de cette thèse.

La deuxième contribution est la méthode de classification. Un réseau Jumeau de neurones à Convolutions Spatio-Temporelles (T-STCNN), est introduit à cette fin. Nous comparons les performances en utilisant notre ensemble de données avec la méthode Two-Stream I3D proposée par Carreira and Zisserman (2017). Nous identifions deux types de tâches : la classification des actions avec les frontières

temporelles connues, et la classification des vidéos sans frontières temporelles.

Nous présentons en Section 2 la base de données sur laquelle les travaux sont menés. Section 3 expose notre méthode de classification et ses résultats sont reportés en Section 4. Enfin la conclusion et les ouvertures pour de futurs travaux sont exposées en Section 5.

2 TTStroke-21

TTStroke-21 est constituée de vidéos centrées sur le joueur de tennis de table. Ces vidéos sont enregistrées avec des caméras GoPro utilisant plusieurs angles de vue. Cependant, afin d'avoir des résultats cohérents et comparables, nous considérons seulement les vidéos enregistrées à 120 images par seconde avec vue aérienne. Les séquences sont enregistrées en intérieur avec lumière artificielle. Des experts en tennis de table annotent les vidéos par le biais de la plateforme d'annotation en utilisant vingt classes de coups conformément aux règles du tennis de table :

- 8 services : *Service coup droit coupé, Service Coup droit lifté, Service Coup droit latéral, Service Coup droit rapide, Service revers coupé, Service revers lifté, Service revers latéral, Service revers rapide.*

- 6 coups d'attaque : *Att. coup droit frappe, Att. coup droit lifté, Att. coup droit flip, Att. revers frappe, Att. revers lifté, Att. revers flip.*

- 6 coups de défense : *Def. coup droit poussette, Def. coup droit bloc, Def. coup droit coupé, Def. revers poussette, Def. revers bloc, Def. revers coupé.*

Ces coups peuvent également être catégorisés en fonction de s'ils sont des coups "droits" ou des "revers".

Pour obtenir un ensemble de données exploitables, les annotations sont filtrées et fusionnées lorsque celles-ci se superposent. En effet, une labellisation des données par deux annotateurs est préférable afin d'éviter les erreurs d'inattention et de valider une première annotation. Cependant, ce critère n'est pas toujours vérifié car ce travail est basé sur le volontariat et qu'un nombre important de vidéos restent à annoter. Le processus d'acquisition et la plateforme d'annotation sont représentés sur la Figure 1.

Sur notre sélection de vidéos, un total de 1387 annotations sont considérées, 1074 sont conservées après filtrage, menant à la segmentation de 1048 coups de tennis de table. Une classe de rejet est construite à partir de cette segmentation en considérant les portions de vidéos entre chaque coup. Un extrait de la base de données est représentée en Figure 2.

a. Acquisition vidéo avec vue aérienne.

b. Plateforme d'annotation.

Figure 1 – Préparation de la base de données `TTStroke-21`.

Figure 2 – Image de présentation de la base de données `TTStroke-21`.

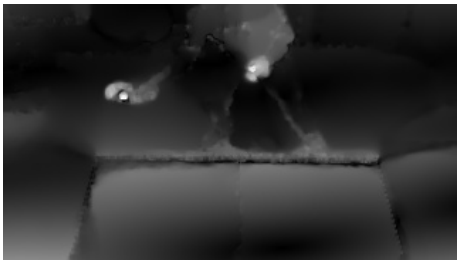
3 Méthode développée

Les frames de la vidéo sont redimensionnées à 320×180 pixels et leur flot optique (FO) $V = (v_x; v_y)$ est calculé. Le FO encode le mouvement horizontal v_x et vertical v_y entre deux images. L'image couleur avec trois composantes : Rouge, Vert, Bleu (RVB) et le FO sont considérés pour la classification des segments vidéo.

3.1 Flot optique et extraction de la région d'intérêt

Différentes méthodes de calcul du flot optique ont été étudiées. Après comparaison des différentes méthodes, il a été décidé d'utiliser la méthode "Beyond Pixel" (BP) (Liu, 2009) pour calculer le FO. Cette méthode permet de capturer des éléments en mouvement qui ne sont pas forcément bien détectés avec les autres méthodes considérées. La Figure 3 représente le filtrage du flot optique qui utilise l'estimation de l'avant plan (Zivkovic and van der Heijden, 2006).

a. Image RVB



b. Magnitude du FO

c. Avant-plan

d. Magnitude du FO après filtrage

Figure 3 – Filtrage du flot optique.

Une région d'intérêt (RI) de taille (W, H) est ensuite déduite. Le centre de la RI est noté : $\mathbf{X_{ri}} = (x_{ri}, y_{ri})$ et est calculé à partir du centre de masse de la carte d'amplitude de mouvement du premier plan et des coordonnées du point maximum de l'amplitude. Cette opération est formalisée par équation 1 :

$$
\begin{aligned}
&\mathbf{X_{max}} = (x_{max}, y_{max}) = \underset{x,y}{argmax}(||\mathbf{V}||_1) \\
&\mathbf{X_g} = (x_g, y_g) = \frac{1}{\sum_{\mathbf{X}\in\Omega}\delta(\mathbf{X})}\sum_{\mathbf{X}\in\Omega}\mathbf{X}\delta(\mathbf{X}) \\
&\text{with} \quad \delta(\mathbf{X}) = \begin{cases} 1 & \text{si } ||\mathbf{V}(\mathbf{X})||_1 \neq 0 \\ 0 & \text{sinon.} \end{cases} \\
&x_{ri} = \alpha\ f_{\omega_x}(x_{max},\ W) + (1-\alpha)\ f_{\omega_x}(x_g,\ W) \\
&y_{ri} = \alpha\ f_{\omega_y}(y_{max},\ H) + (1-\alpha)\ f_{\omega_y}(x_g,\ H)
\end{aligned}
\tag{1}
$$

avec $\alpha = 0.6$, $\mathbf{\Omega} = (\omega_x, \omega_y) = (320, 180)$ la taille des images. La fonction $f_\omega(u, V) = max(min(u, V - \frac{\omega}{2}),\ \frac{\omega}{2})$ permet d'avoir des régions qui sont dans la limite des dimensions de l'image. De façon à éviter une instabilité des RI au cours du temps, un filtre temporel Gaussien est appliqué sur le centre de la RI pour stabiliser son évolution dans le temps.

Les données RVB sont normalisées en les divisant par leur maximum théorique 255 tandis que les données du FO sont normalisées en utilisant une normalisation statistique, nommée "Normale", basée sur la distribution des valeurs maximales du FO telle que :

$$
\begin{aligned}
&v' = \frac{v}{\mu + 3 \times \sigma} \\
&v^N(i,j) = \begin{cases} v'(i,j) & \text{si } |v'(i,j)| < 1 \\ SIGN(v'(i,j)) & \text{sinon.} \end{cases}
\end{aligned}
\tag{2}
$$

avec v et v^N représentant respectivement une composante du FO $\mathbf{V}$ et sa normalisation. μ et σ sont la moyenne et l'écart-type de la distribution de la valeur absolue d'une composante du FO estimée sur toute la base de données. A noter que différentes techniques de normalisation ont été testées et cette dernière a obtenu les meilleures performances.

3.2 Classification des données avec un réseau de neurones Jumeau - T-STCNN

Notre modèle T-STCNN, qui signifie en anglais Twin Spatio-Temporal Convolutional Neural Network, est constitué de deux branches individuelles avec pour chacune trois couches convolutionnelles 3D utilisant chacune 30, 60 et 80 filtres de taille $3 \times 3 \times 3$, avec comme fonction d'activation ReLU. Chaque couche convolutionnelle est suivie d'une couche maxpooling permettant de diviser par deux la dimension de nos données. Chaque branche se termine par une couche entièrement connectée de taille 500. Les deux branches sont ensuite fusionnées grâce à une fonction bilinéaire du type $y = x_1^T A x_2 + b$ avec x_1 et x_2 les données de chaque branche, A et b respectivement les poids et biais entraînables, et avec en sortie un vecteur y de la taille du nombre de classes considérées, c'est à dire 21. La sortie est ensuite suivie d'une

fonction Softmax pour obtenir une probabilité de classification. L'architecture du T-STCNN est représentée en Figure 4. Cette architecture est dite *Jumelle* par son organisation identique des deux branches.

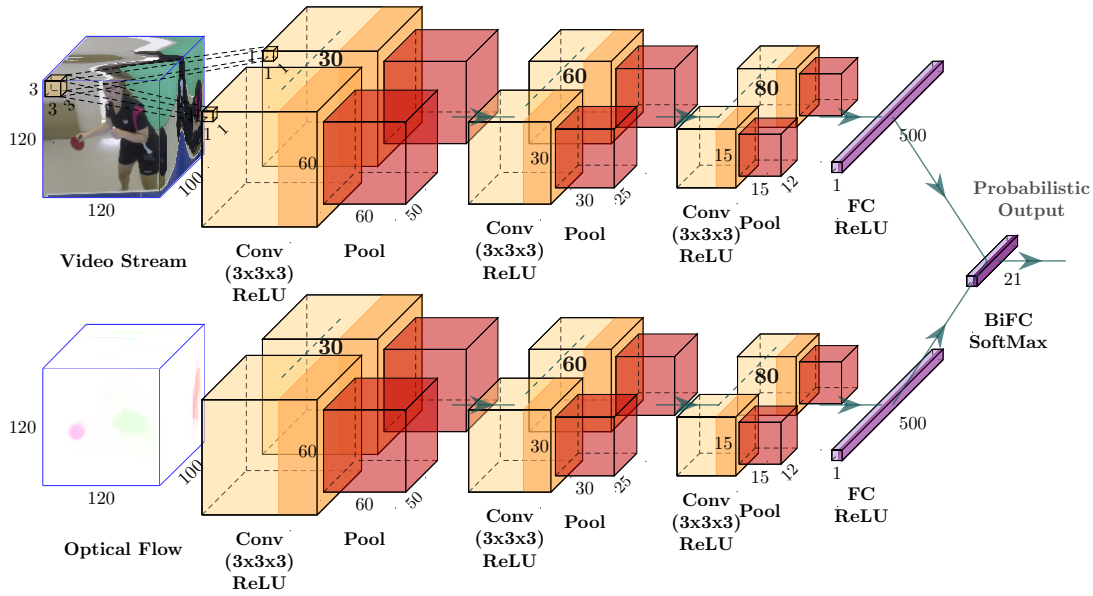


Figure 4 – Réseau Jumeau de neurones à Convolutions Spatio-Temporelles.

Ce réseau prend en entrée les données RVB et leur FO correspondant sous forme de cuboïdes de taille $(W \times H \times T) = (120 \times 120 \times 100)$ avec W la largeur, H la hauteur et T la durée. Ce modèle est comparé avec les modèles I3D (Carreira and Zisserman, 2017) : RVB-I3D, FO-I3D et Two-Stream I3D.

Les résultats sont également comparés avec nos modèles une branche qui prennent en entrée les données RVB et FO séparément : RVB-STCNN et FO-STCNN. Le modèle une branche est inspiré de la même architecture, mais à la place d'une couche bilinéaire, une couche entièrement connectée est utilisée. Deux autres méthodes de fusion sont également considérées :

- une méthode de fusion précoce : RVB et FO sont concaténés en données d'entrées à cinq composantes (R, G, B, v_x, v_y) que l'on notera FP-STCNN pour Fusion Précoce.

- une méthode de fusion tardive : la moyenne des sorties des modèles une branche FO et RVB est considérée pour la classification que l'on notera FT-STCNN pour Fusion Tardive. Cette méthode est similaire à celle utilisée par le modèle Two-Stream-I3D, notre modèle de référence.

De plus, les mécanismes d'attention sont aussi étudiés. Un bloc d'attention 3D est construit en s'inspirant des travaux menés en 2D par Wang et al. (2017a). Ce

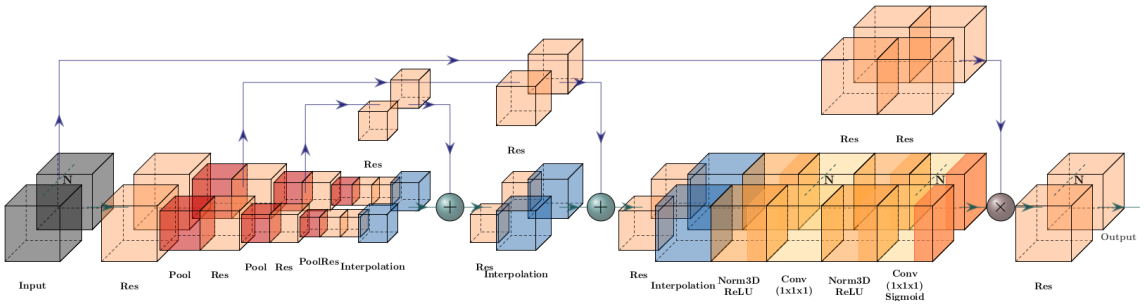


Figure 5 – Bloc d'attention 3D.

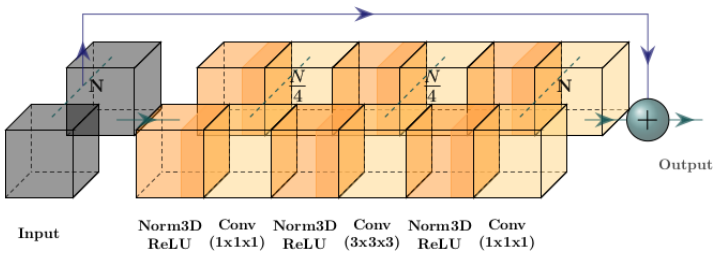


Figure 6 – Bloc résiduel 3D.

bloc est présenté en Figure 5. Ce dernier utilise des blocs résiduels 3D qui sont représentés en Figure 6.

Les blocs d'attention sont insérés à la sortie des couches convolutionnelles et les performances sont comparées avec les méthodes sans mécanisme d'attention. Les blocs d'attention utilisent une normalisation par lot. Le lot étant le nombre de données fournies en entrée du modèle en même temps. Cette normalisation peut s'effectuer statistiquement avec des variables entraînées ou par lot durant l'inférence (pendant l'étape test) en considérant seulement les données en entrée. Les résultats des deux manières sont reportés dans la Section 4.

3.3 Entraînement des réseaux

L'estimation des paramètres de nos réseaux se fait par descente de gradient stochastique (SGD) avec un momentum de Nesterov (Sutskever et al., 2013) de 0.5. Notre fonction objective est la fonction de perte entropie-croisée. Le nombre de segments négatifs (sans coup de tennis de table) extraits de `TTStroke-21` est choisi deux fois plus grand que la moyenne du nombre de coups par classe. `TTStroke-21` est divisée en plusieurs ensembles afin d'entraîner nos réseaux : Entraînement (Ent), Validation (Val) et Test, avec comme proportions respectives : 70%, 20% et 10% comme représenté sur le Tableau 1. Le nombre total d'extraits est reporté dans la colonne "Tot". La durée des coups en nombre d'images est aussi précisée afin de mieux appréhender leur particularité.

Comme observable dans le Tableau 1, les services ont en moyenne une durée

Table 1 – Distribution des données sur chaque set et durée des coups.

	# **échantillons**				# **frames**		
Coups de tennis de table	**Ent**	**Val**	**Test**	**Tot**	**Min**	**Max**	**Moy***
Service coup droit coupé	58	17	8	83	125	269	182 ± 35
Service Coup droit lifté	56	16	8	80	100	273	171 ± 51
Service Coup droit latéral	57	16	9	82	101	273	192 ± 39
Service Coup droit rapide	67	19	9	95	100	273	184 ± 52
Service revers coupé	56	16	8	80	133	261	188 ± 31
Service revers lifté	43	12	6	61	100	265	186 ± 42
Service revers latéral	60	17	9	86	129	269	193 ± 33
Service revers rapide	57	16	8	81	100	273	175 ± 48
Att. coup droit frappe	28	8	4	40	100	173	134 ± 21
Att. coup droit lifté	21	6	3	30	100	229	155 ± 32
Att. coup droit flip	25	7	3	35	100	265	195 ± 49
Att. revers frappe	45	13	6	64	100	233	158 ± 34
Att. revers lifté	23	7	3	33	101	277	177 ± 43
Att. revers flip	31	9	5	45	113	269	186 ± 44
Def. coup droit poussette	6	2	1	9	121	229	155 ± 31
Def. coup droit bloc	19	5	3	27	100	261	131 ± 37
Def. coup droit coupé	22	6	3	31	121	233	189 ± 25
Def. revers bloc	8	2	2	12	100	137	115 ± 14
Def. revers poussette	23	7	3	33	105	177	143 ± 19
Def. revers coupé	29	8	4	41	129	229	177 ± 25
Extrait négatif (non coup)	74	21	11	106	100	1255	246 ± 154
Total	808	230	116	1154	100	1255	182 ± 65

* sous la forme : valeur moyenne ± déviation standard.

plus grande que les coups en milieu de jeu. Cette particularité nous a encouragé à développer des méthodes de classification prenant en compte l'intégralité des coups effectués et non pas seulement les $T = 100$ images considérées dans nos cuboïdes vidéos en entrée des réseaux. Ces méthodes d'évaluation sont décrites en Section 3.5.

3.4 Augmentation des données

L'augmentation des données est effectuée pendant l'entraînement pour économiser de l'espace de stockage et pour obtenir des données continuellement nouvelles durant la phase d'apprentissage. Nous effectuons d'une part une augmentation spatiale : une rotation aléatoire, une translation aléatoire et une homothétie aléatoire sont appliquées à la fois sur les images RVB et le flot optique. Les transformations sont centrées sur nos régions d'intérêt. Enfin, nous effectuons un retournement horizontal de la frame avec une probabilité de 0.5. D'autre part, nous effectuons une augmentation temporelle : T images successives sont extraites du segment vidéo considéré. L'image centrale est déterminée en suivant une distribution normale de probabilité autour du centre temporel de l'extrait considéré. Cette augmentation temporelle est schématisée sur la Figure 7.

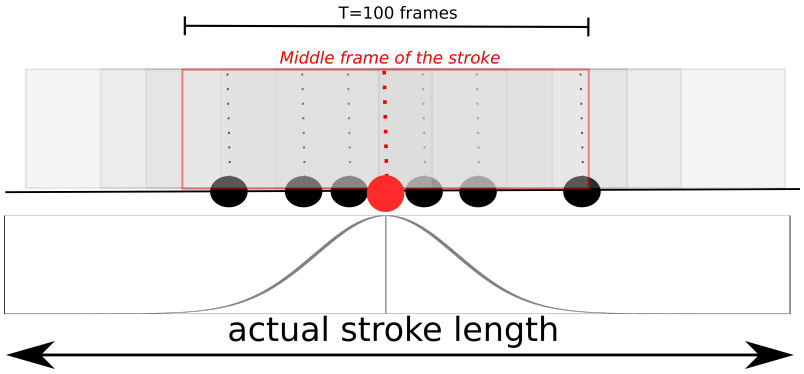


Figure 7 – Représentation de 7 extractions de données à partir d'un même coup en utilisant une augmentation temporelle.

L'augmentation des données se fait seulement sur l'ensemble d'entraînement. Les données non augmentées en entrée du réseau sont centrées temporellement sur le coup : $t_{milieu} = t_{debut} + \frac{t_{fin} - t_{debut}}{2}$.

3.5 Évaluation des performances

Les performances des modèles sont évaluées sur deux tâches distinctes : i) classification des données segmentées et ii) détection et classification des données à partir des vidéos non segmentées. Pour cette deuxième tâche, l'ensemble des vidéos sélectionnées pour nos expériences est utilisé. L'évaluation pour les deux tâches se fait en terme de précision : c'est à dire nombre de segments bien classifiés divisé par le

nombre total de segments. Concernant la décision pour la tâche de classification, en plus d'utiliser la décision centrée temporellement, trois décisions temporelles sont effectuées sur le test set :

- "TVote" prend en compte la décision de chaque fenêtre et attribue la classe à celle obtenant le plus de votes.
- "TMoy" prend la moyenne des probabilités des fenêtres et attribue la classe à celle qui obtient la probabilité maximale.
- "TGauss" pondère la probabilité obtenue par chaque fenêtre en utilisant une pondération Gaussienne. Les fenêtres centrées sur les données auront donc plus d'influence sur la décision que celles aux extrémités.

Ces décisions temporelles utilisent une fenêtre glissante temporelle de taille dix.

Pareillement, pour la tâche conjointe de détection et classification, une fenêtre temporelle glissante avec un pas de un est utilisée pour la classification. Les sorties sont ensuite lissées en utilisant les trois mêmes principes avec une fenêtre de taille 150 pour les méthodes "Vote" et "Moyenne", et un filtre de taille 201 pour la méthode "Gaussienne".

4 Résultats

Les résultats sont présentés en deux temps. Dans un premier temps pour la tâche de classification pure et dans un deuxième temps pour la tâche conjointe de détection et classification.

4.1 Performances pour la tâche de classification pure

Le Tableau 2 décrit les performances obtenues pour chaque modèle entraîné. Ce dernier précise aussi le nombre d'époch[1] nécessaire pour obtenir ces résultats.

Comme nous pouvons le remarquer sur ce tableau, les modèles aux meilleures performances sont ceux utilisant les mécanismes d'attention. De plus, en comparant le nombre d'époch, les mécanismes d'attention permettent une convergence plus rapide des modèles. RVB-STCNN avec attention obtient le meilleur taux de classification. Le flot optique quant à lui a certes une efficacité correcte pour la classification sur les données centrées temporellement, mais a plus de difficultés en utilisant les méthodes d'évaluation temporelles `TVote`, `TMoy` et `TGauss`. C'est certainement pourquoi le réseau Jumeau n'obtient pas de meilleures performances malgré un plus grand nombre de modalités en entrées : la modalité FO l'induit probablement en erreur. Notre deuxième hypothèse provient du fait que le modèle Jumeau, par sa plus grande taille, est plus gourmand en ressources GPUs que les autres. La taille

[1] itérations sur le set d'entraînement en entier

Table 2 – Comparaison des performances de classification des modèles en terme de précision.

		Précision en %					
Modèles	**Epochs**	**Ent**	**Val**	**Test**	**TVote**	**TMoy**	**TGauss**
RVB-I3D	778	98.3	72.6	69.8	84.5	84.5	84.5
RVB-STCNN	1665	96.7	88.7	89.8	67.6	74.6	70.3
RVB-STCNN avec attention	524	96.5	88.3	92.4 93.2*	93.2 94.9*	94.1 95.8*	92.4 **96.6***
FO-I3D	1112	98.8	74.8	73.3	82.8	82.8	82.8
FO-STCNN	1449	97.5	79.6	75.9	80.2	80.2	78.5
FO-STCNN avec attention	732	96.4	83.5	85.6 90.7*	66.1 71.2*	71.2 69.5*	66.1 70.3*
Two-Stream I3D	-	99.2	76.2	75.9	84.5	87.1	86.2
FP-STCNN	1450	90.8	84.8	82.2	81.4	83.9	83.9
FT-STCNN	-	97	88.7	89.8	87.3	87.3	87.3
FT-STCNN avec attention	-	97	88.7	90.7 94.9*	90.7 93.2*	92.4 94.1*	92.4 94.1*
T-STCNN	1784	95.8	87.8	**93.2**	91.5	90.7	91.5
T-STCNN avec attention	591	97.3	87.8	92.4 95.8*	71.2 77.1*	72 78*	72 77.1*

* normalisation sur le lot

du lot a donc du être diminuée, par rapport aux autres modèles, pour l'entraîner. De ce fait, l'entraînement du modèle est moins efficace, ce qui pourrait expliquer ses performances limitées. Cependant, sans mécanisme d'attention, les meilleurs résultats sont obtenus avec le T-STCNN. Les autres méthodes de fusion obtiennent des performances moindres. Une fusion précoce des modalités n'a en effet pas grande signification, car chaque modalité représente des entités différentes (apparence ou mouvement). Une fusion tardive est *correcte*, mais il n'y a pas de pondération des caractéristiques extraites de chaque modalité comme on peut l'avoir avec une fusion intermédiaire. C'est donc sans surprise que le T-STCNN obtient les meilleurs scores.

Nos méthodes de référence I3D (Carreira and Zisserman, 2017) obtiennent des performances correctes mais en deçà de nos modèles (sauf pour le FO-STCNN sans mécanisme d'attention). Ceci peut s'expliquer par la profondeur de leur modèle. Ce dernier est nettement plus profond que notre architecture proposée. Ainsi, celui-ci aura plus tendance à se focaliser sur des caractéristiques qui ne sont propres qu'à l'ensemble d'entraînement. Ce sur-apprentissage des données se caractérise par un fossé entre la précision de l'ensemble de validation et celui d'entraînement, ce qui est en effet observable dans le tableau.

4.2 Performances pour la tâche conjointe de détection et classification

Le Tableau 3 reporte les résultats de nos modèles pour la tâche conjointe de détection et classification. Cette dernière s'effectue sur l'ensemble des vidéos considérées comportant un nombre minimum de dix actions pour éviter les vidéos non complètement annotées. Ce tableau est divisé en deux : la première partie considère toutes les classes de notre tâche, la deuxième ne considère pas les classes négatives (absence de coup). Ceci est motivé par la présence accrue de segments vidéos de la classe négative. En effet, un modèle classifiant toute la vidéo en "négatif", pourrait obtenir de meilleures performances qu'un modèle classifiant correctement certains coups, ceci étant dû au déséquilibre des classes. Cette présence accrue de données négatives est due aux temps morts entre deux jeux ou aux balles perdues qui nécessitent d'être récupérées. Aussi, chaque début et fin de vidéo sont souvent accompagnés d'un temps sans aucune activité.

Comme on peut le voir à partir du Tableau 3, les meilleures performances pour cette tâche sont obtenues avec le modèle RVB-STCNN utilisant le mécanisme d'attention. Néanmoins, on peut remarquer l'instabilité des résultats obtenus en fonction de la partie du tableau et du mode de normalisation utilisé dans les blocs d'attention. Des performances un peu moindre mais nettement plus stables sont obtenues avec le modèle T-STCNN sans bloc d'attention. Ces résultats soulignent l'importance de la normalisation des données, mais aussi de la fusion des modalités. Le modèle T-STCNN avec le mécanisme d'attention est trop lourd à entraîner. Un compromis entre blocs d'attention et modalités sur lesquelles l'appliquer reste donc à être déterminé.

Table 3 – Performances des modèles implémentés pour la tâche conjointe de détection et classification.

	Précision en %			
Modèles	**Brut**	**Vote**	**Moyenne**	**Gaussienne**
RVB-STCNN	57	80.1	80.8	80.2
RVB-STCNN avec attention	43.8 70.1*	63.3 85.9*	64.7 **86.4***	63.6 86.1*
FO-STCNN	70.3	80.5	80.9	81
FO-STCNN avec attention	10.7 69.3*	20.1 78.4*	21.5 79.2*	21.1 79.8*
T-STCNN	60.8	79.8	**80.2**	79.7
T-STCNN avec attention	31 72.9*	46.8 82.1*	47.7 82.3*	47.3 83*
sans prendre en compte les segments négatifs				
RVB-STCNN	41.5	44.8	46.2	49.1
RVB-STCNN avec attention	65.4 66.9*	80.4 74.3*	81.9 74.8*	**84.6** 77.6*
FO-STCNN	50.4	55.4	59.2	62.4
FO-STCNN avec attention	40 33.8*	52.9 20.9*	55.8 22.9*	58.6 26.5*
T-STCNN	60.5	76.8	76.9	**78.4**
T-STCNN avec attention	45.2 45.6*	63.8 35.1*	65.6 35*	67.9 39.4*

* normalisation sur le lot

5 Conclusion et perspectives

Ces travaux de thèse visent à améliorer les performances des athlètes en développant de nouvelles méthodes et de nouveaux outils pour les entraîneurs et les étudiants. Il a été montré que nos modèles peuvent obtenir des résultats convaincants en comparaison avec les modèles I3D de référence. Les résultats ont été comparés suivant une méthode d'ablation, c'est à dire en isolant les contributions de chaque étapes de la méthode dans la performance globale. Au final, les meilleurs résultats sont obtenus avec le modèle RVB-STCNN utilisant les blocs d'attention. La limitation de nos équipements ne nous permet pas d'entraîner de la même manière le modèle Jumeau et le modèle RVB utilisant les mécanismes d'attention, ce qui malheureusement ajoute un biais dans les performances réduites du modèle Jumeau. Cependant une meilleure stabilité des performances est observée avec le modèle Jumeau T-STCNN. Il a aussi été montré l'importance de la normalisation des données et de l'apport des mécanismes d'attention en terme de convergence des modèles.

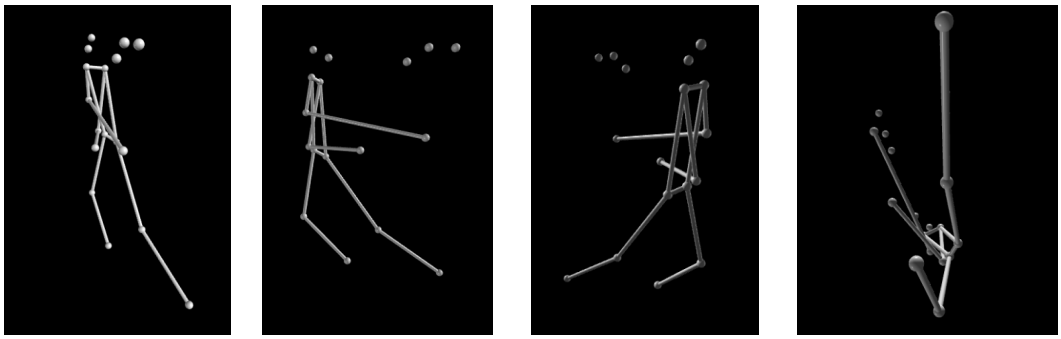
Les méthodes présentées pour la classification fine des coups de tennis de table peuvent être améliorées. Même si nous avons fait de nombreux tests sur les types d'architecture, la position des blocs d'attention ou le nombre de filtres à utiliser, ces variables peuvent toujours être améliorées. Le grand nombre de combinaisons possibles fait que nous nous sommes focalisés sur des aspects qui nous semblaient les plus importants. De plus, les méthodes de l'état de l'art sont aussi en constante évolution. De nouvelles méthodes ont vu le jour depuis le commencement de cette thèse et pourraient être appliquées à la base de données `TTStroke-21`. Les méthodes d'estimation de flot optique ont elles aussi évolué. Prendre en compte d'autres méthodes d'estimation de mouvement et les comparer en terme de classification fine est aussi une des voies de développement possible. Enfin, la base de données `TTStroke-21` est continuellement enrichie. Des tests sur les nouvelles vidéos annotées avec des cadences d'enregistrement plus grandes et des points de vue différents sont aussi une piste d'investigation.

La caractérisation de la qualité d'un coup effectué est l'une des pistes principales pour compléter notre objectif d'amélioration des performances des sportifs. Des méthodes reposant sur l'estimation de la pose ont déjà vu le jour afin de donner un aspect qualitatif aux mouvements effectués par le sportif (Morel et al., 2017; Einfalt et al., 2018). La Figure 8 propose un début de piste combinant : i) un calcul de profondeur à partir d'une image (Ramamonjisoa and Lepetit, 2019) et ii) l'estimation de la pose du joueur (Newell et al., 2016).

Cette modélisation 3D pourrait être comparée avec un modèle de référence, en utilisant une métrique adaptée. Il serait, de cette manière, possible d'obtenir une évaluation qualitative des coups de tennis de table.

Enfin, même si notre cadre applicatif est le tennis de table, le protocole de notre méthode peut être étendu à d'autres sports. Il conviendrait de construire une base de données pour le sport d'intérêt et de l'annoter en utilisant la même plateforme d'annotation développée. Le modèle de reconnaissance lui peut être

a. Pose du joueur sur image RVB **b.** Profondeur estimée



c. Vue 3D face **d.** Vue 3D coté **e.** Vue 3D arrière **f.** Vue 3D dessous

Figure 8 – Estimation de la pose et de la profondeur à partir d'une image combinées pour donner un modèle 3D.

adapté en changeant la taille de la dernière couche (donnant la probabilité de sortie) afin qu'elle soit de la taille du nombre de classes du sport considéré.

Les travaux présentés dans cette thèse ont été validés par des publications dans une revue internationale, cinq papiers de conférences internationales, deux papiers d'un workshop international et une tâche reconductible dans le workshop MediaEval où les participants peuvent appliquer leurs méthodes de reconnaissance d'actions à notre base de données `TTStroke-21`. Le workshop est décrit en anglais en appendice B. Deux autres papiers de workshop internationaux sont en cours de préparation, ainsi qu'un chapitre de livre. La liste de ces publications est disponible en appendice A. Le code permettant de construire les différents réseaux développés, ainsi que la façon de les entraîner, sont disponibles publiquement comme décrit en appendice C. Différentes vulgarisations scientifiques furent aussi menées en lien avec cette thèse; ces dernières sont listées en appendice D.

Contents

General Introduction

1 Introduction

The importance of computer vision in our society has grown over the last decades, and is now present in many aspects of everyone's life. The access to a large part of the population to new technologies has widen the challenges, topics of interest and applications of computer vision. High performances reached by recent Artificial Intelligence (AI) technologies are bound to have a considerable impact on the society in the near future, for the worst or hopefully for the best. A brief overview of computer vision contributions in the society is presented, as research conducted in this thesis fits into this context.

The recent technological breakthrough in Machine Learning, in particular in Deep Learning, has raised a strong interest among the population, leading to the development of many user-oriented applications. One can mention the work of Feng et al. (2019); Jiang et al. (2018) who apply deep learning tools to fashion in order to advise potential customers on how to dress. Tools to beautify images (Chen et al., 2018a) are widely used on social media. As people are taking a huge amount of pictures with mobile devices, methods for indexing and retrieving them automatically are being developed (Kuzovkin et al., 2018).

Applications for monitoring the natural environment are also possible through computer vision. The analysis of plant growth is for example performed in the work of Wang et al. (2019c); Yasrab et al. (2019, 2020); Smith et al. (2019). Applications available to the public are also investigated, with for instance Krause et al. (2018) who propose "What The Plant" system, based on Convolutional Neural Network (CNN) in order to identify plants from images. In the topic, we can also mention the Pl@ntNet project[2].

The democratization of technological tools has also increased the access to information and news. Methods are investigated to rank the quality of media (Marcelino et al., 2018) or to connect the images of an article to its text content (Oostdijk et al., 2020). However, on the other side, the amount of fake news has exploded, and images from unrelated contents can be used to spread false information. To overcome such nuisance, methods have been developed, for example to retrieve images from database to find their source using image matching techniques (Vo et al., 2019).

Closely related, a very active area of applications is video retrieval from large databases (Schoeffmann, 2019). The best performances are obtained using deep learning approaches (Rossetto et al., 2019). With the huge increase of online videos and expansion of Video On Demand (VOD) platforms, methods are being developed to automatically describe video contents (Alayrac et al., 2018) or provide video summarization (Bost et al., 2019). Conversely, Li et al. (2019a) try to localize specific temporal segments fitting a text description. Also in the field of multimedia, Cohendet et al. (2018) investigate the video memorability with the aim to develop contents that can be better memorized.

[2] https://plantnet.org/

Computer vision can also benefit to culture. One can mention the work carried out by Koch et al. (2018) which assess building conditions in cities from images. Coupled with a cultural heritage database (Obeso et al., 2016), this can help to map city historical buildings and prevent their deterioration. In addition, with the advent of Unmanned Aerial Vehicles (UAV) utilization, it becomes easier to rapidly and automatically map entire cities (PAN et al., 2019).

Recently autonomous vehicles have also raised many challenges, and such system driven by AI should be totally reliable. Even if some studies show that they already outperform human driving abilities, legal responsibility of AI system is still in debate. With respect to Computer Vision tasks, autonomous vehicles implies image semantic segmentation (Durand et al., 2017), which evolved to video segmentation (Galasso et al., 2013; Sundberg et al., 2011). Best methods are based on 3D-CNN with end-to-end encoder-decoder approaches (Hou et al., 2019; Saffar et al., 2018). They are trained using a big amount of data (Yu et al., 2020) and incorporating different modalities such as the optical flow. With the progress of such technologies, Billy et al. (2019) are able to propose 3D scene reconstruction in real time for autonomous driving; which can be coupled with anomaly detection on highways (Singh et al., 2020). With the increasing performance of real-time computer vision, autonomous driving becomes more and more reliable, opening the way for implementation at larger scale in the society.

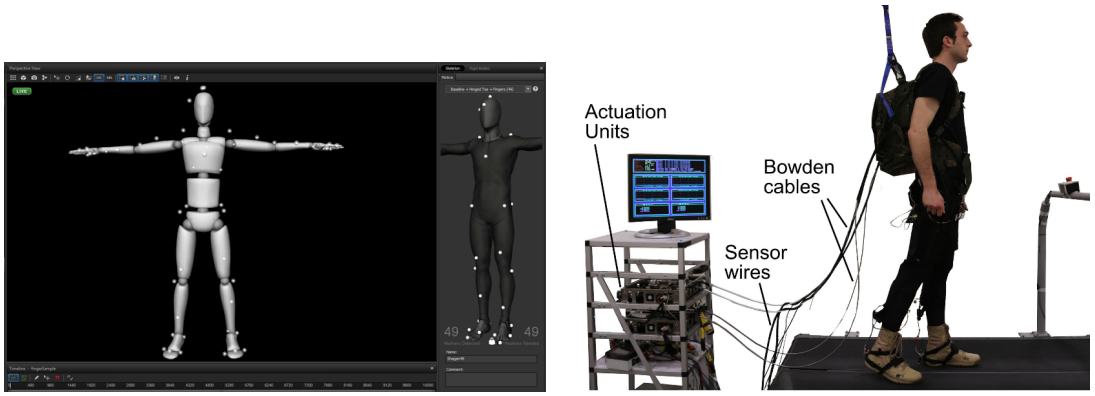
Another aspect of the increasing responsibility of AI methods is their role in medicine. Computer vision methods are being developed with the aim to assist in clinical procedures (Leibetseder and Schoeffmann, 2018; Sokolova et al., 2020) or provide accurate imaging for better diagnosis of the patient (Emilien et al., 2013; Çiçek et al., 2016). Deep learning methods can also be applied for emergency needs in the society such as the COVID-19 pandemic. One can refer to the promising work of Chatterjee et al. (2020) using chest X-Ray images which classifies patients with COVID-19 or pneumonia from healthy ones.

This research thesis was funded by the New Aquitania region. It is part of the regional project CRISP about computer vision for sport performance, which long term ambitions are to promote Sport and Health in the society.

2 The CRISP Project

The ComputeR vIsion for Sport Performance (CRISP) project is a multidisciplinary and regional project. Despite intensive research, recognition of actions with low inter-class variability remains a challenge. The target application of our research is fine grained action recognition in sports with the aim of improving athletes performance. Without loss of generality, we are interested in recognition of strokes in table tennis. The purpose is to make cameras "smart" to analyse sport practices. Collecting individual data on physical activity (connected watches, smart clothes, exoskeleton) might be a potential source of information for innovative research in

the field of sport and well-being. However, the analysis of sport gesture is often confined to laboratory studies as represented in Figure 9, and limit the applications.



a. Interface for recording exoskeleton with sensors.

b. Man on conveyor belt with sensors to analyze his way of walking.

Figure 9 – Representation of experiments in laboratories to model human actions.

Analysis of physical practices in ecological context, meaning without sensors or markers, is primordial to avoid induced stress and/or discomfort which could hinder the athlete in its performance and its practice. CRISP aims at developing digital tools in computer vision allowing acquisition, recognition and analysis of sport gestures in an ecological context. The objective is to help students learning in training centres or sport faculties through those tools. Our case study is table tennis.

CRISP project, in addition to the financial support of the New Aquitania region, has also the support from the Table Tennis League of New Aquitania. The financial support allowed to finance

- a full PhD grant (2017-2020) which led to the fruit of this work.
- the acquisition of high speed cameras used in another strongly related project (Calandre et al., 2021)

The collaboration of two scientific communities, Sport and Computer Science, led to:

- `TTStroke-21`: a new dataset annotated through a crowdsourcing platform designed for this purpose.
- AI methods for fine-grained recognition of sportive gestures using machine learning and video processing methods
- an ongoing work conducted at the University of La Rochelle devoted to fine analysis of sport gestures using computer vision.

2.1 TTStroke-21

In TTStroke-21, twenty stroke classes and an additional rejection class are considered according to the rules of table tennis. This taxonomy was designed with professional table tennis teachers. Videos are recorded at the Faculty of Sports of the University of Bordeaux - STAPS. Students are the athletes filmed and the teachers supervise exercises conducted during the recording sessions. The recordings are markerless and allow players to perform in natural conditions. This dataset, Figure 10, is the first step of the project.

Figure 10 – Teaser image of the TTStroke-21 dataset.

The dataset TTStroke-21 focuses on the different moves of a particular sport, which are the different strokes in table tennis. In our case, the considered video dataset is complex for classification task, as some stroke classes have only weak differences with respect to their visual appearance. This low inter-class variability makes the classification task much more challenging than in the classical sport datasets.

2.2 AI for Sport Performances

The second step of the CRISP project is the classification process. A Twin Spatio-Temporal Convolutional Neural Network (TSTCNN) is introduced for this purpose. The model similarly processes RGB images and Optical Flow through a succession of spatio-temporal convolutions. An intermediate fusion is done before the calculation of the class scores. We use spatial and temporal data augmentation during the training phase. Performances are compared for models using only RGB images or Optical Flow data, and also with early and late fusion approaches. Moreover, normalization of the optical flow and incorporation of attention mechanism are investigated. Additionally, we compare our approach with our baseline being the Two-Stream I3D model proposed by Carreira and Zisserman (2017). For evaluation, two different

tasks can be considered: the classification task which is performed with trimmed videos (strokes segmented in time) and joint action detection and classification from untrimmed videos.

3 Conclusion and Thesis Outline

The role of computer vision in society and its use have expanded exponentially these last years. The aim of this thesis is to use the recent approaches of computer vision in the domain of sport. Our objective is to improve athletes performances by developing new methods and tools for coaches and students. Our case study is table tennis but the same protocol could be extended to different sports.

The layout of this manuscript comprises three parts and annexes.

- In Part I, we examine the state-of-the-art methods in computer vision related to action classification. This part is divided into three chapters. Chapter 1 presents a brief overview of the first action classification methods using hand-crafted features and their evolution over time.
  In Chapter 2, deep learning methods, which have greatly evolved in the recent years, are exposed and compared.
  The various datasets for action recognition along with `TTStroke-21` dataset are presented in Chapter 3.

- In the second part of the manuscript, we present the different methods investigated to perform stroke classification on `TTStroke-21`.
  Chapters 4, 5 and 6 focus on stroke classification respectively using RGB data, Optical Flow data and both modalities.
  Features of our best model are analyzed in Chapter 7.

- In a last and third part (Chapter 8), we incorporate an attention mechanism in our method in order to increase the classification performances. Our results are highlighted at the end of each chapter for a better understanding of the course of this thesis.
  Finally, the conclusion and the prospects for future works are drawn in the General Conclusion.

Research presented in this manuscript was validated with publications in one international journal, five international conference papers, two international workshop papers and a task in MediaEval evaluation campaign where participants can apply their action recognition methods to `TTStroke-21`. The workshop description is available in appendix B. Two additional international workshop papers are in process along with one book chapter. The list of the publications is reported in appendix A. The code allowing to build and train the presented models are publicly available online as described in appendix C. This thesis has also given birth to many scientific popularisation events which are listed in appendix D.

Part I

Related Work on Action Recognition from Videos

Abstract

This first part focuses on the existing work conducted on action classification. The performances and the techniques have evolved years after years, at the same time that the computation capacity were increasing. The engineered features have incorporated more and more deep learning tools in order improve the classification scores. Then classical classification methods using handcrafted features have given way to the fully deep learning methods. The deep learning methods have thus expanded in number and complexity until becoming the state-of-the-art for action classification. The datasets dedicated to action recognition, in order to always offer new challenges to the methods reaching maximum scores on the previous datasets, have grown in term of classes, complexity, tasks and number of videos. `TTStroke-21` dataset, dedicated to fine-grained action recognition in table tennis, is introduced in this context.

Keywords

Action classification, Handcrafted features, Deep neural networks, Action recognition datasets, Table Tennis

Summary

Chapter 1

Action Recognition Using Handcrafted Features

1 Introduction

Recognition of actions in videos is one of the key problems in computer vision. Despite intensive research, recognition and discrimination of visually very similar actions remain a challenge. The current trend nowadays is to use deep learning methods for classification tasks. Before the advent of Deep Neural Networks (DNNs), methods were focusing on building handcrafted features that are discriminant enough to classify actions. Many of the early methods for video recognition were more or less a direct extend of techniques developed for images.

The method of Ji et al. (2013), was one of the first to use Deep Learning via a 3D CNN for action recognition, and they did not obtain better results than the state-of-the-art methods on the `KTH` dataset. It is only from 2014 and the innovative *two-stream network* approach of Simonyan and Zisserman (2014) that temporal coherence will be exploited in CNN and that Deep Learning approaches will begin to supplant other methods.

Better performances using Deep Neural Network (DNN) does not mean that engineered features have to disappear. On the contrary, human understanding of visual scenes influences the choice of the designed features such as e.g. Optical Flow (OF). Still recent work may use handcrafted features as a baseline or fuse them with the deep features extracted by a DNN. Budnik et al. (2017) confirm that Support Vector Machine (SVM) does not perform better than a partially retrained Deep Convolutional Neural Network (DCNN) and that the learned features lead to better results than engineered ones; however the fusion outperforms the single modalities. The same conclusion is drawn by Ceroni et al. (2019) who use images to measure their degree of exoticism. These works confirm the findings of the community which are that the fusion of multiple features allows to improve recognition scores for complex visual or multi-modal content understanding (Ionescu et al., 2014).

Both engineered and deep features may also be fused at different steps of an end-to-end method. For example, Truong et al. (2018) focus on retrieving events from lifelogging data. Their method has several steps using different types of features. Grouping of images is done using motion vector from FlowNet (Dosovitskiy et al.,

2015) and linking of the different groups with Bag-of-Visual Words (BoVW) Nguyen et al. (2015) and Scale-Invariant Feature Transform (SIFT) descriptor (Lowe, 2004). Ultimately they use deep features (Zhou et al., 2014) for scene classification. Conversely, combination of handcrafted features might feed a DNN in order to perform classification (Sivaprasad et al., 2018; Rahmani et al., 2018).

As the number of potential features were increasing over time, Tang et al. (2013) presented AND/OR graphs in order to combine best features. The context was event detection from videos, another challenge in video understanding and indexing (Over et al., 2011b), which also led to the well known Bag Of Fragments video encoding method (Mettes et al., 2015).

In this short chapter, we present in Section 1 a quick chronological overview of the handcrafted feature methods developed for action recognition. The overview comprises three subsections which focus on action recognition in general (2.1), scene classification (2.2) and video understanding in racket sports (2.3). We discuss their impact on the classification of actions challenges in Section 3. The datasets on which such methods have been applied will be described in Chapter 3.

2 Handcrafted Features in Videos

Handcrafted features extracted from videos started to our best knowledge from feature extraction from images. Efforts were afterwards made for extracting information from the temporal domain. Such features were mainly used in the action recognition task, but also for other tasks such as scene and event recognition from videos. Most of the approaches using handcrafted features seek for their compact representation. The model of Bag of Words (BoW) or BoVW. BoW convert vector representations to "words" and, using a clustering method, such as k-means, lead to a "dictionary" which will represent a class.

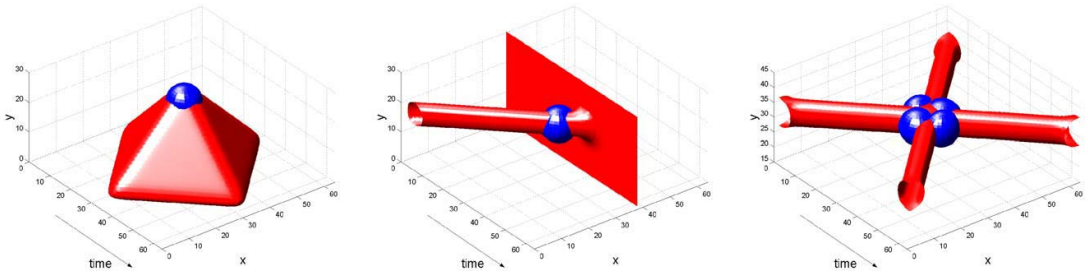
In order to have a well rounded overview, we treat action recognition and scene classification in two different subsections. We also treat in a third subsection, video recognition in the domain of racket sports, which is our target domain of interest.

2.1 Action Classification From Videos

Laptev (2005) introduced in 2003 Spatio-Temporal Interest Points (STIP) features, which is an extension to the temporal dimension for videos of the 2D corner detector (Harris and Stephens, 1988; Förstner and Gülch, 1987). The equivalent to image corner in video are points which change direction over space and time, as depicted with synthetic examples in Figure 1.1.

They show that their descriptor matches with the action performed, meaning that their points will be located where and when the action happens. As presented in Figure 1.2, the knee is well detected and the leg pattern is extracted.

In a following work, Schüldt et al. (2004) apply their method on their newly



a. Moving corner **b.** Ball hitting a wall **c.** Two balls colliding

Figure 1.1 – Synthetic examples of STIP (Laptev, 2005).

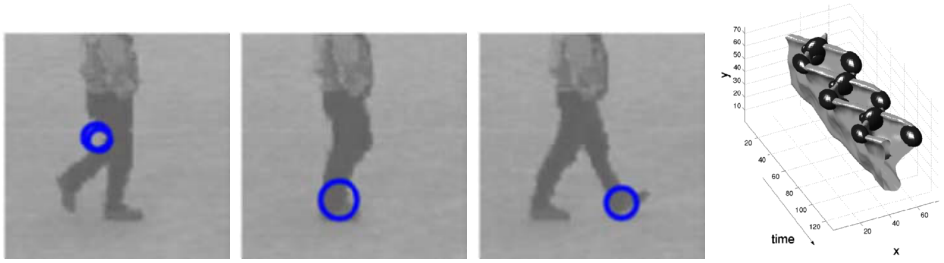


Figure 1.2 – Examples of STIP for walking action with resulting leg pattern (Laptev, 2005).

created KTH dataset, which became one of the first widely used action dataset. Comparison is done using SVM classifier (Boser et al., 1992) and Nearest Neighbor Classification (NNC) on histogram (Bag-of-Words) of their spatio-temporal local features. It results in a good classification score for actions which are not similar; but scores on similar actions such as "Walking", "Jogging" and "Running" remained low. The motion of those actions are very close and the STIP are concentrated on the same body parts. Dollar et al. (2005) use the similar principle for extracting spatio-temporal interest points with the aim to increase the number of extracted points to help in the classification process.

Laptev and Pérez (2007) present the Coffee and Cigarettes dataset dedicated to action detection and classification in movies. This dataset comprises only two actions: drinking and smoking. In their paper, they use jointly the Histogram of Oriented Gradients (HOG) descriptor (Dalal and Triggs, 2005) and Motion Boundary Histogram (MBH) descriptors (Dalal et al., 2006), using AdaBoost algorithm (Collins et al., 2002). Histogram of Oriented Optical Flow (HOF) features, similar to HOG, are presented in Figure 1.3.

At the same time, Gorelick et al. (2007) introduce Space-Time Shapes (STS) features for action classification, represented in Figure 1.4, along with the new Weizmann action dataset.

Classification of STS is based on the same ideas as image shapes classification introduced a year earlier by Gorelick et al. (2006) using Poisson equation. The authors classify the computed shape with NNC procedure using euclidean distance.

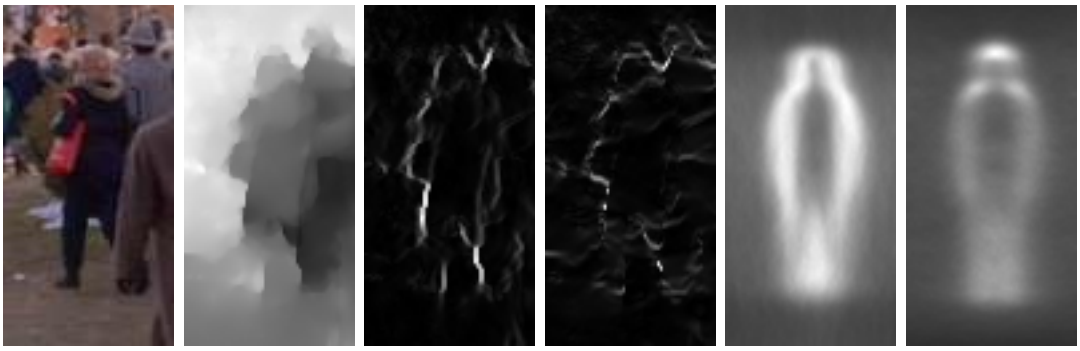


Figure 1.3 – Motion Boundary Histogram (MBH) computation process with from left to right the image, its OF amplitude, its horizontal and verticals gradients and their average over the training set (Dalal et al., 2006).

Figure 1.4 – Space-Time Shapes (STS) of "jumping-jack", "walking" and "running" actions (Gorelick et al., 2007).

They reach an accuracy of 97.8% on their dataset with however a low confidence on similar actions. It is important to stress that their dataset is acquired in a controlled environment and has the same complexity as the KTH dataset.

Schindler and Gool (2008) concatenate Gabor filters features (Daugman, 1988) and OF features in order to perform action classification with a SVM approach on KTH and Weizmann datasets. They show the superiority of their model on both datasets, compared to other methods, and come to the conclusion that one frame is enough to get a satisfying classification score. Indeed, using only one frame and the posture of the person, actions in both datasets are easily distinguishable. The same year, Liu and Shah (2008) introduce the "video words" features. The histogram of the video words for each type of action is different enough to allow an efficient classification. With SVM classifier, the authors obtain an accuracy of 94.15% on KTH dataset.

In 2009, action recognition methods were already reaching very high accuracies on both KTH and Weizmann datasets and the introduction of UCF11 dataset by Liu et al. (2009) gave more space for improvements. Indeed, this dataset is more challenging since it is recorded "in the wild", meaning in natural conditions, under the constraints of camera motion and flickering for example. The UCF101 samples are extracted from the YouTube platform. Along with their dataset, they propose a method for classification based on motion features and static features. They use AdaBoost learning method on the histogram-based representation and compare it with k-means clustering method. AdaBoost leads to better results and the hybrid combination resulted in 93.8% of accuracy on KTH dataset against 71.2% on UCF11 showing the higher complexity of the task for such dataset with the same number of classes.

Chaudhry et al. (2009) introduce at the same moment Histogram of Oriented optical Flow (HOF) features and reach 94.4% of accuracy on Weizmann actions dataset. HOF computation is represented in Figure 1.5. The method is simple and easy to reproduce and will be used later on in by Wang et al. (2011), and then extended in (Wang et al., 2013), along with MBH, HOG features to compute dense trajectory features. Dense trajectory features are computed using dense optical flow field (Farnebäck, 2003). The overview of the method is presented in Figure 1.6.

It is similar to the work carried by Sun et al. (2010) but differs from the trajectory which are computed using KLT tracker (Lucas and Kanade, 1981) and SIFT points trajectories.

Next breakthrough in action classification was obtained with Fisher Vectors introduced by Csurka and Perronnin (2010). First used for image classification (Krapac et al., 2011), the method is based on Fisher Kernel principle (Jaakkola and Haussler, 1998). By combining the MBH and SIFT features, they reach 90% of accuracy on UCF50 dataset (Reddy and Shah, 2013) which is an extension of UCF11 (Liu et al., 2009). They also add Mel-Frequency Cepstral Coefficient (MFCC) audio features (Rabiner and Schafer, 2007), for the event recognition challenge proposed by TRECVID (Over et al., 2011b).

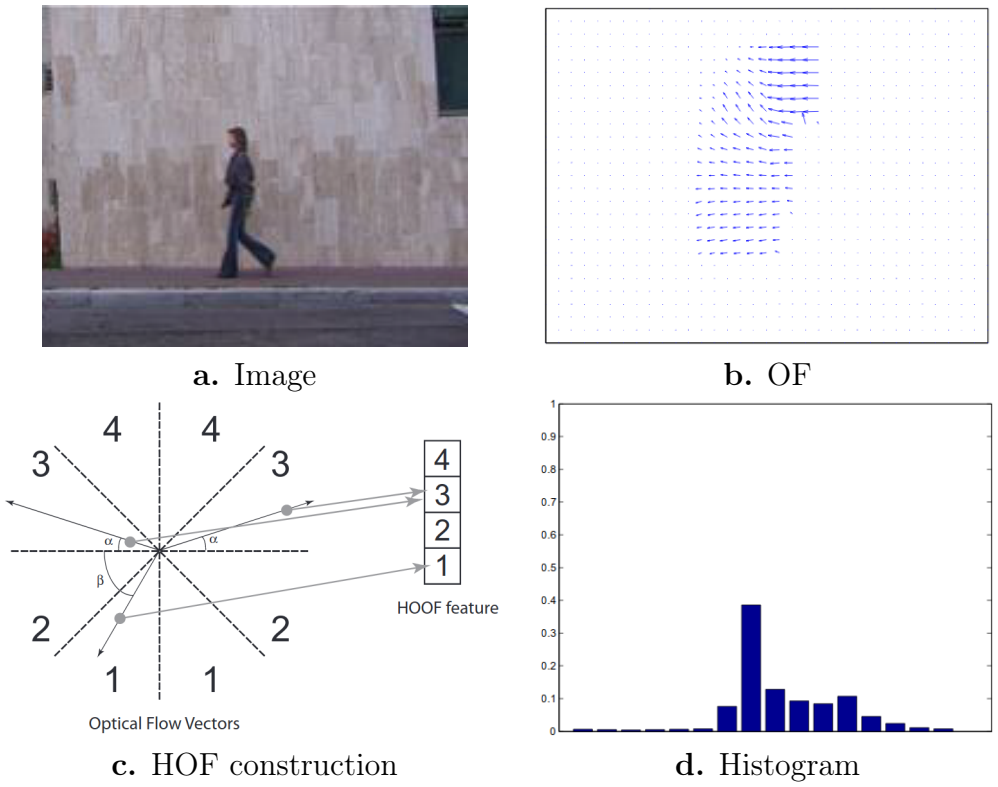


Figure 1.5 – Histogram of Oriented Optical Flow (HOF) computation process (Chaudhry et al., 2009).

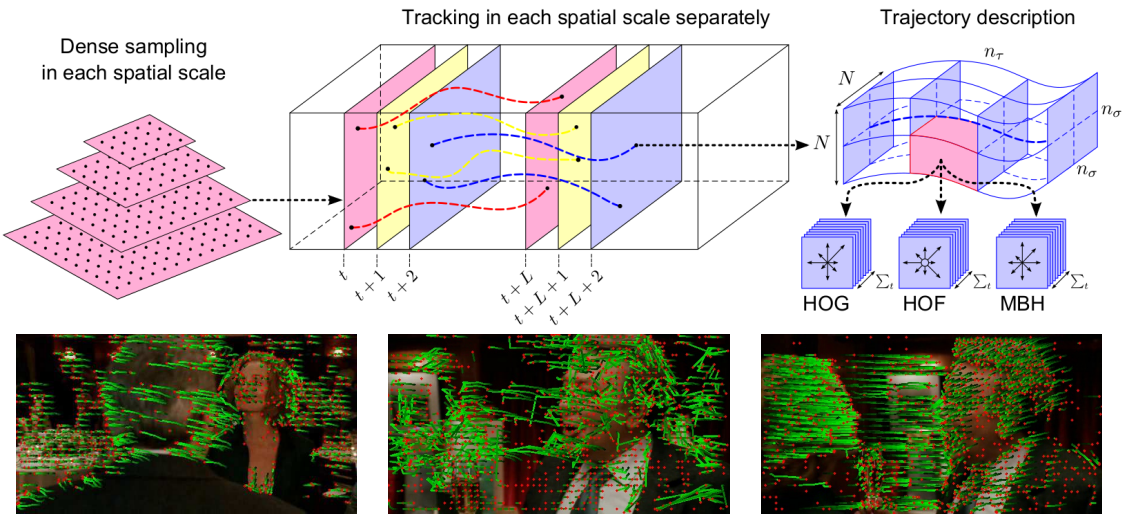


Figure 1.6 – Dense trajectory features computation process (Wang et al., 2011) and results.

Wang and Schmid (2013) improve dense trajectory features by considering camera motion. Camera motion is estimated using dense optical flow and Speeded Up Robust Features (SURF) descriptors Bay et al. (2008). A homography is estimated using Random Sample Consensus (RANSAC) Fischler and Bolles (1981). The Improved Dense Trajectories (IDT) perform 91.2% of accuracy against 88.6% with regard to the original dense trajectory features. This work is used later on many applications such as action localization Yuan et al. (2016).

Gaidon et al. (2013) redefine actions as "actoms". Actom is a short atomic action with discriminative visual information, such as opening a door. It is therefore useful for action localization but can also be applied to classification-by-localization. The definition of actoms is important in the field of action recognition to decompose an action in individual parts. Actoms thus can be present across different actions such as entering or leaving a room with "opening door" as an actom. Their understanding can lead to better video representation and accordingly to better classification. However, a too great number might lead to representation of features from the training set, but not present in the test set and unrelated to the action performed. The number of actoms to consider becomes then a variable to control according to the complexity of the actions to classify.

The optical flow data can also be analysed to detect critical points as done by Beaudry et al. (2014). Sequences are then characterized by the trajectory of the critical points in the frequency domain using Fourier coefficients. Trajectory of such critical points are illustrated in Figure 1.7.

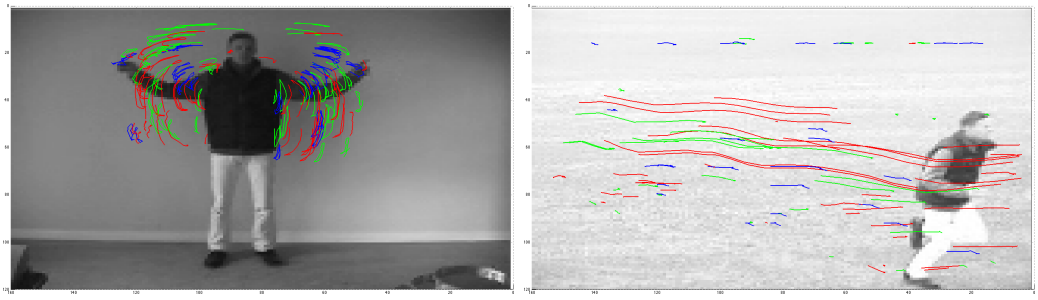


Figure 1.7 – Trajectory of critical points for action classification (Beaudry et al., 2014).

Then, associated which classic HOG and HOF descriptors through concatenation of Fisher vectors on each modality, their method leads to accurate classification on classical benchmark datasets such as KTH, UCF11 and UCF50.

Another way to perform action recognition is developed by Jain et al. (2014) who introduce the concept of Tubelets. It is a sampling method to produce 2D+T sequences of bounding boxes where the action is localized. This method, which tackles the localization and classification problem of actions at the same time, is based on super voxel generation through an iterative process using color, texture and motion to finally create tubelets as represented in Figure 1.8. Those tubelets are then described by MBH features, and one BoW per class method is used for

classification. The classifier with the maximum score assigns the class to the tubelet. This method was motivated by the challenging MSR dataset (Cao et al., 2010) which contains videos where different actions can happen at the same moment but at different localization.

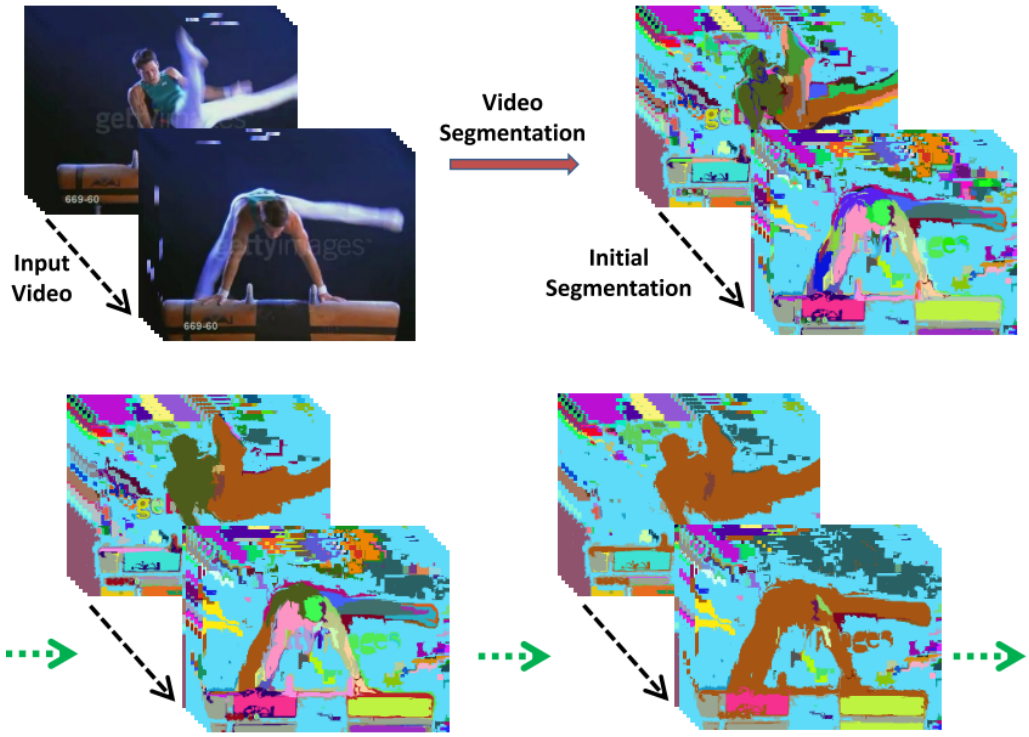


Figure 1.8 – Tubelets generation representation (Jain et al., 2014).

More recently, Mi et al. (2018) base their method on improved dense trajectory (IDT) features. Their field of application is action recognition from wearable-camera videos characterized by strong camera motion. IDT are therefore extremely appreciated in such context since the calculated homograhy should filter out the non desirable motion. To assess the efficiency of their method, they train their model on fixed cameras and apply it to wearable cameras. The used dataset is similar to EDUB[1] (Bolaños et al., 2015) which is also egocentric but with images more or less temporally correlated. On this particular dataset, the scope of the research is more lifelog related and aims at summarizing and tracking the actions performed during the day. In the same application field, Asnaoui and Radeva (2020) cluster the images using histobins in Hue, Saturation, Lightness (HSL) color space and dynamic time wrapping features (Bellman and Kalaba, 1959).

[1] http://www.ub.edu/cvub/dataset/

2.2 Scene Classification

Scene classification might not be directly link to action recognition, but temporal features to extract might be useful for both scenarios.

Shroff et al. (2010) use the degree of busyness, degree of flow granularity and degree of regularity as motion attributes to categorize the different scenes. They model the dynamic data using the theory of chaotic systems based on the evolution of GIST descriptors (Oliva and Torralba, 2001) expressing scales and orientations, over time. By doing so, they can compute the chaotic invariants which will serve as discriminant dynamic features. Their dataset called Maryland "in the wild" consists of 13 different scenes with only ten videos per class: `Avalanche, Iceberg Collapse, Landslide, Volcano eruption, Chaotic traffic, Smooth traffic, Forest fire, Waterfall, Boiling water, Fountain, Waves and Whirlpool`. Best performances are obtained by merging computed dynamic features with spatial GIST features using their mean values for the whole video. The same conclusions are drawn later by Derpanis et al. (2012). They report a gain in classification of scenes using the temporal information of the video. Hence, they also introduce the Yupenn dataset for scene classification. Both datasets have now been widely used later on by the scientific community for the scene classification task.

Theriault et al. (2013a) (extended in Theriault et al. (2014)) apply Slow Feature Analysis (SFA) (Wiskott and Sejnowski, 2002) on V1 features (Theriault et al., 2013b) on both datasets a year later, as presented in Figure 1.9.

V1 features are based on convolution and maxpooling using Gabor filter and trained using HMAX framework (Serre et al., 2007) (those methods are borderline between DNN and handcrafted features). The SFA is based on computation of temporal derivatives and assumes a smooth motion pattern which allows stable dynamic feature extraction. They claim that such a method could also be beneficial in the action recognition domain. SFA was actually used for action classification by Zhang and Tao (2012) with the Principal Component Analysis (PCA) (Jolliffe, 1986) features of the foreground mask, without achieving state of the art results.

2.3 Video Understanding for Racket Sports

Our interest is fine-grained action recognition in table tennis. We therefore present methods focusing on video classification and/or segmentation in the domain of racket sports with also some interest for end-users.

Efros et al. (2003) propose a motion descriptor based on optical flow in order to classify actions in sports. For this purpose they consider three different datasets: Ballet, Football and Tennis datasets which are presented in Chapter 3. They track the player (or the person performing the action) and build a 3D volume based on their motion. Their motion descriptor has four channels: the positive and negative values for horizontal and vertical motions.

Then classification is performed following a nearest neighbourhood approach

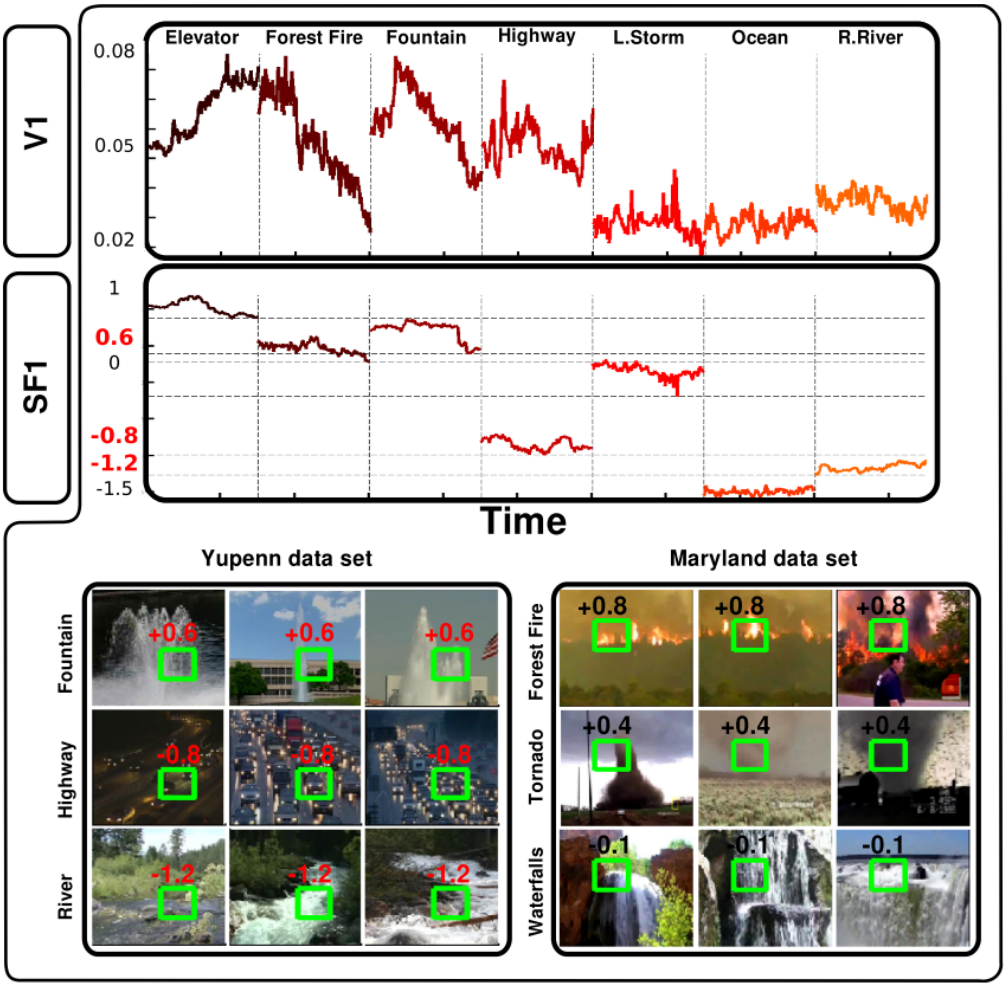
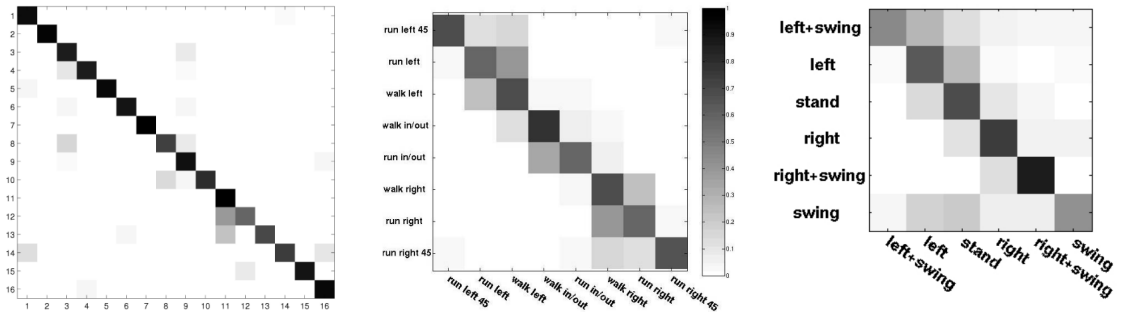


Figure 1.9 – Comparison of V1 and Slowest Features of V1 (SF1) on Yupenn and Maryland datasets with SF1 localisation and value represented on input (Wiskott and Sejnowski, 2002).

using similarity metric:

$$S = \sum_{k} C_1(k) C_2(k) \tag{1.1}$$

with C_1 and C_2 being two cuboids samples based on the motion descriptors at coordinate k. Their performances for each dataset is discussed through the confusion matrices obtained, which are reported in Figure 1.10.



a. Ballet dataset **b.** Football dataset **c.** Tennis dataset

Figure 1.10 – The different confusion matrices for each dataset using 3D volume of OF (Efros et al., 2003).

Even if Ballet dataset and Tennis dataset are acquired in a controlled environment, performances for the Tennis dataset are more limited. Football dataset comes from broadcast source which explains the limited performances. Moreover, the number of classes for the Tennis dataset is lesser than the two others, however, it is where their method is the less efficient. This underlines the greater complexity of racket sport and their fine-grained aspect.

Another research field in video classification aims at identifying the different parts of tennis broadcasting. To do so Hidden Markov Models (HMMs) are applied to tennis action recognition by Kijak et al. (2006). Their model is statistic and integrate the structure of tennis match. They combine audio and key frame features to be able to segment, with a good accuracy, the different parts of the tennis broadcasting such as the first serves, rallies, replays and breaks. The sound of the crowd such as applause, the sound of the ball or the commentator speech combined with key frames which capture visual information lead to 86% of segmentation accuracy compared to 65% and 77% with only respectively visual features and audio features. HMMs are also adapted and modified later on to be applied to other classification problems (Pentland et al., 2005; Yu, 2010; Morency et al., 2010). Such applications are interesting for sport coaches who wish to comment and examine only sequences of sports.

de Campos et al. (2011) present a new dataset for tennis actions. This one contains only three types of classes: “hit”, “serve” and “non-hit-class”. The dataset is

build from TV broadcasts of tennis games (matches of females in the Australian Open championships). They are interested in action localization and their classification. To do so, they introduce a local BoW method on the Spatio-Temporal gradients HOG3D features (Kläser et al., 2008) which are an extension of the classical 2D HOG features in three dimensions. They also use STS features. Both features are from the located actor and classification is performed using Fisher discriminant analysis. They obtain an accuracy of 77.6% using STS model based. Their confusion matrix is represented in Figure 1.11.

Global accuracy of 77.6%

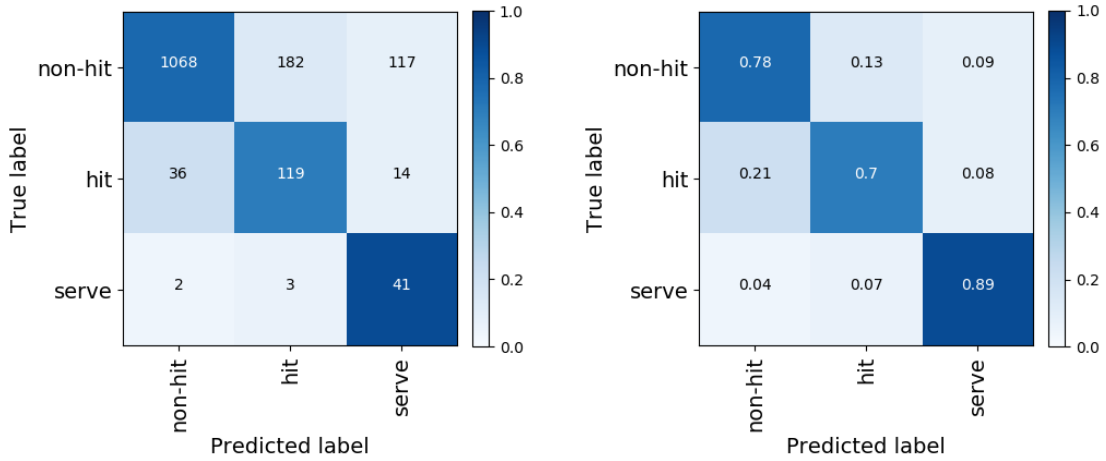


Figure 1.11 – Confusion matrix of tennis game from TV brodcast (de Campos et al., 2011). Left with number of samples and right normalized.

One can see that "serve" samples are easier to classify than "hit" or "non-hit" samples. This is certainly due to the time that a service takes and its decomposition in time, which starts by large movement of the player when launching the ball. Hit and non-hit classes are then harder to distinguish because the hit class is very limited in time. Looking only at the player shape, the ball might not be visible, and feature might look the same as when the player is simply moving in the field.

Recently, deeply related to our domain, Calandre et al. (2019) use the OF Singularities with BoW and SVM in order to classify very similar actions. This task is also called fine-grained action classification. They apply their method on the `TTStroke-21` dataset which contains 20 different strokes and a negative class. Their method is inspired from Beaudry et al. (2014) which uses the trajectory of critical points for classification. In this case, the actions to recognize are the different types of strokes preformed during table tennis training session. However, the scores remain low due to the high similarity of the different strokes and the limited amount of video samples. It makes generalization of extracted features harder.

Table Tennis stroke recognition is also performed by Liu et al. (2019b). It is

based on their Body Sensor Network (BSN). Their sensors collect acceleration and angular velocity information from the upper arm, lower arm and the back of the player. From the recorded signals, they extract PCA features (Jolliffe, 1986) which are then fed to a SVM. They reach an accuracy of 97.4%, however they use only five classes: "forehand drive, "block shot", "forehand chop", "backhand chop" and "smash". Similarly, Xia et al. (2020) recently propose classification from integrated wearable sensors using K-means and DBScan clustering (Ester et al., 1996; Schubert et al., 2017). Their taxonomy is more limited than in `TTStroke-21` since they use only nine classes across badminton and table tennis sports. They reach an accuracy of 86.3% when considering all the classes. This score reaches 92.5% when considering only table tennis but this classification is limited to four classes: "Service", "Stroke", "Spin" and "Picking up". The extent of their taxonomy is thus limited and does not contribute much to the player experience. Furthermore, using such sensors, strongly limits the application possibilities and has a greater cost regarding training equipment adaptation. Also their system does not offer visualisation of the stroke performed since it is based on sensors, and it limits the feedback for the player.

Recently, a method is introduced by Wang et al. (2020) to get the tactics of the players based on their performance in past matches. Their model is based on Hidden Markov Model (HMM) and aims at characterizing and simulating the competition process in table tennis. They use richer taxonomy and terms of stroke techniques than the previously presented methods : 13 different classes and four player positions which can be combined. Compared to `TTStroke-21`, we consider ten classes with two player positions: "Forehand" and "Backhand". Their goal is therefore, not to classify an input, but to simulate matches between two different players. It is not directly linked to action recognition methods, but it does give a tool for players to simulate sport encounters and give credits to the `TTStroke-21` dataset which propose much richer taxonomy than previous datasets.

3 Conclusion and Discussion

In this chapter we went through the main achievements in terms of action classification using engineered/handcrafted features. The scientific teams working on this task tried to extend our knowledge in image classification by using the temporal domain in different manners, either considering a 2D object as 3D, or extracting 2D features along the temporal axis and fuse the image features with different techniques.

Such handcrafted features allow a good classification on datasets that remain simple: either with a low number of classes or with classes that are easily separable. It does become more complicated when the task focuses on one particular sport with different actions within. In our field, non experts would have difficulties to classify different actions performed in `TTStroke-21`. Even trained players do not agree on which task a stroke fall by simply looking at the recordings. Thereby,

models that can reach better performances than humans are described in the next chapter dedicated to the DNN features.

Chapter 2

Deep Neural Networks for Action Recognition

1 Introduction

Most recent action recognition methods developed in the literature have been using deep learning approaches and high-dimensional spaces. In the domain of image classification, a specific kind of Neural Networks first introduced by LeCun et al. (1989) has become very popular: the Convolutional Neural Networks (CNN). CNN, since the breakthrough at the 2012 ImageNet Challenge, have demonstrated a great improvement for image classification (Krizhevsky et al., 2012). They are also known as shift invariant or space invariant neural networks, based on their shared-weights architecture and translation invariance characteristics. Convolutional Neural Networks (CNNs) are regularized versions of multilayer perceptrons. Classic multilayer perceptrons, also called Fully Connected (FC) networks, means that each neuron in one layer is connected to all neurons in the next layer (refer to equation 2.1). CNN principle is to perform convolution operations using trainable filters, usually of size 3, at different level from input to output. The convolutions result in feature maps which are, in most cases, reduced in size using max-pooling layers. Those layers keep the maximum value of the feature map using, most of the time, 2×2 filters with stride two in both directions. After a certain deepness, the resulting feature map is flatten and feed to a FC layer, equation 2.1:

$$y = xA^T + b \tag{2.1}$$

with y being the output of the dense layer of length N, N being in our case the number of class considered, x the features flattened, A the matrix of size $length(x) \times N$ with trainable weights and b the trainable bias of length N. In order to obtain a probabilistic output (summing to one), in most of the cases y is processed by Softmax function as described in equation 2.2:

$$y'_i = \frac{exp(y_i)}{\sum_j^N exp(y_j)} \tag{2.2}$$

Thenceforth, the output is fed to a loss function. The loss function measures

how much training penalizes the deviation between the prediction y and the desired output (true labels or ground truth). Various loss functions may be used, depending on the task of interest. Euclidean loss is often used for real-value regression, e.g image reconstruction (Nogas et al., 2020). In the case of classification task it is often the Cross Entropy Loss ($\mathcal{L}$), which is being used as described in equation 2.3:

$$\mathcal{L}(y, class) = -log(\frac{exp(y'_{class})}{\sum_i^N exp(y_i)}) \tag{2.3}$$

Finally all the weights are being updated by back propagating the loss into the network using an optimizer and a learning rate. A classical iterative optimizer is the Stochastic Gradient Descent (SGD) as presented in equation 2.4:

$$\theta = \theta - lr \nabla \mathcal{L}(\theta) \tag{2.4}$$

With θ being the trainable weights, lr the learning rate, $\mathcal{L}$ the objective function (loss) and $\nabla \mathcal{L}(\theta)$ the gradient of the weights according to the loss. Through iterations, the network parameters are tuned in order to decrease the loss function and therefore, lead to features meant to perform the classification task.

The first deep learning breakthrough in natural image classification with a relatively shallow CNN architecture, AlexNet (Krizhevsky et al., 2012, 2017), has inspired new deep learning methods, relatively different in terms of both deepness and architecture, such as GoogLeNet (Szegedy et al., 2015), VGG-Net (Simonyan and Zisserman, 2015) and Residual Network (ResNet) (He et al., 2016).

A CNN can be considered as a two step end-to-end classifier, in which first layers (convolutional layers) serve for feature extraction and the last layers (fully connected layers), as Neural Network (NN) classifier such as Multi-Layer Perceptron (MLP) (Minsky and Papert, 1987) with one or more hidden layers. An example of such a network is the AlexNet Architecture illustrated in Figure 2.1.

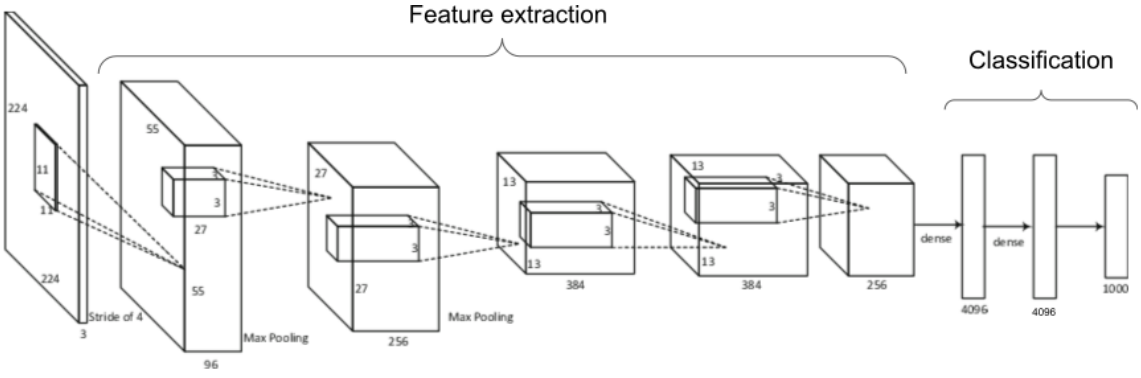


Figure 2.1 – AlexNet Archiecture (Krizhevsky et al., 2012).

Convolutional layers perform convolution operation with multiple trainable filters. Results of convolution are submitted to the neurons with non-linear activation,

such as ReLu function (eq. 2.5), and then pooled in order to reduce the input dimension.

$$ReLu(x) = max(0, x) \tag{2.5}$$

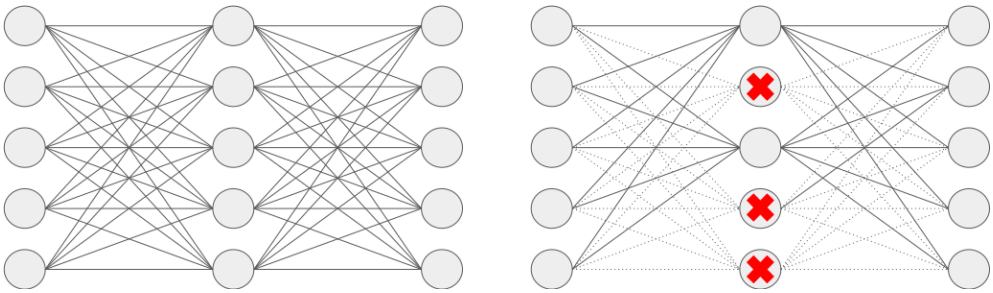
With Zemmari and Benois-Pineau (2020), this process of convolution/pooling is explained from the signal processing point-of-view, as building a pyramidal representation of the input 2D signal. The difference here is the fact that the filters are trainable and non-linearity functional layers are introduced at each level of the pyramid.

A Deep CNN is a supervised classifier, its parameters which are filter coefficients have to be trained in the global optimization process of supervised learning. Training methods starting from plane Gradient Descent method, as used by LeCun et al. (1989), have incorporated modifications such as momentum and Nesterov momentum (Dahl et al., 2013; Gillot et al., 2018), elements of adaptation to the data such as AdaGrad (Duchi et al., 2011). Furthermore, various regularisation techniques have been proposed such as adding regularisation terms in the global loss function to minimize, as described in equation 2.6.

$$\mathcal{L}^{\text{reg}}(\theta) = \mathcal{L} + \frac{\lambda'}{2} \|\theta\|_2^2 \tag{2.6}$$

with λ' the L_2 regularizer which is similar to a weight decay λ when $\lambda' = \frac{\lambda}{lr}$ (Hanson and Pratt, 1988).

Drop out layers were also introduced in the networks (Hinton et al., 2012), and have the effect that some neuronal connections are randomly cut, as depicted in Figure 2.2.



a. MLP without drop out **b.** MLP with drop out on second layer

Figure 2.2 – Drop out representation on Multi-Layer Perceptron (MLP) (Hinton et al., 2012).

Then, an initial MLP is being transformed by stacking into the union of different "local networks", some of them sometimes sharing data.

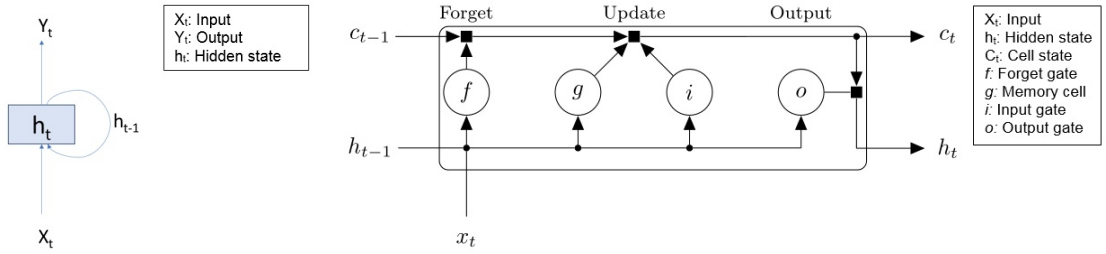
In the field of speech recognition, CNN (Sainath and Parada, 2015) and latter Convolutional Recurrent Neural Network (CRNN) (Arik et al., 2017) also greatly improve the performances. Convolution layers also have been modified to be adapted

to certain tasks. As well, receptive fields (Luo et al., 2016), sparse convolutions either in 2D or 3D (Li and Yu, 2018; Wang et al., 2019a), or time asymmetric convolutions (Wu et al., 2019) have been investigated.

For video applications in general and action recognition in particular, first models proposed were more or less direct extension of image classification methods. Many questions were raised, such as how to consider the temporal dimension, how to train the models, which architecture to use for spatio-temporal data. As presented in previous chapter for handcrafted features, we can consider videos as 3D data and process them similarly to 2D images; or treat the temporal dimension differently; or extract temporal information such as dynamic data that can fed a DNN.

However, most methods need to consider extra information, which obviously leads to larger networks, greater number of parameters and the need of a greater number of Graphics Processing Units (GPUs) with bigger capacities. This might not be possible for every research team, and brought some of them to try attaining accurate results with restrictions on the model size or computation time. This aspect thus brings many shades in the performances, and have brought to light many different methods which shall not be compared only in terms of performances, but also by their means to achieve them. In addition to such limitations, the choice of the architecture for a specific task remained open, which Peng et al. (2019) tried to solve by creating an algorithm to automatically design neural networks for action recognition.

Numerous works use models based on temporal networks such as Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) (Ullah et al., 2018), represented in Figure 2.3.



a. RNN **b.** LSTM

Figure 2.3 – Representation of a simple RNN and its LSTM sub-category which overcomes the issue of the vanishing gradients by using memory gates (Baccouche et al., 2011). Images from MathWorks.

However, RNN may be harder to train depending on the application context. Besides, LSTM are more efficient when they are coupled to the output of a CNN (Ng et al., 2015).

3D convolutional neural networks are a good alternative as well for capturing long-term dependencies (Carreira and Zisserman, 2017), and involve 3D convolutions in space and time. This type of approach translates in a powerful way, through

the prism of deep learning, what we knew beforehand: extract features from time windows and use them for classification (Stoian et al., 2016). Recent methods also improve performances of 3D CNNs by either capturing simultaneously features at different video frame rates Feichtenhofer et al. (2019) or by adding non local operations in the network (Wang et al., 2018a) (inspired by the classical non-local means method (Buades et al., 2005)).

The use of different modalities as inputs to a DNN boosts generally performances. In video, the use of optical flow together with Red, Green, Blue (RGB) images is common practise. Presented briefly in the last chapter, this modality is presented and discussed in details in Chapter 5. To deal with multiple modalities at several spatial and temporal scales, Neverova et al. (2016) present their "ModDrop" method in order to drop certain modalities to perform classification. This method was inspired from classical dropout. It was recently use by Jing et al. (2019) to perform WiFi-based indoor localization.

In this chapter, we will cover chronologically the different DNNs developed to perform action classification. As their number exploded in recent years, we focus on the most important ones and the ones which offer interesting alternatives to take into account the temporal dimension. We compare performances on one of the most used datasets for action classification, namely: UCF101 (Soomro et al., 2012), which contains 101 different classes. Performances, when available, are reported in Table 2.1. The different datasets for action classification are presented in next chapter.

2 2D Convolutional Neural Networks for Action Classification

2D convolution refers to the fact that convolutions are perform on a 2D spatial support of the image. For RGB data, 2D convolution actually uses 3D kernels to weight each color channel differently. Similarly, 2D convolution can be applied on 3D data by considering the third dimension as the channels. It is for example the approach of Debard et al. (2018) for touch gesture recognition, where the finger position is encoded at different time as the channels. 2D and 3D convolutions are really close one to another. And the presence of channels in the input make their differentiation even more tricky. Indeed, what we call 2D convolution on RGB data, actually uses 3D kernels to weight each channel differently. However the way that the kernel will move in the image will only be in 2 dimensions. 2D filters are thus usually of size ($N_{channels} \times 3 \times 3$) (capture of spatial information) while 3D filters are of size ($N_{channels} \times 3 \times 3 \times 3$) (capture of spatial and temporal).

In the scope of action recognition, it is also what Simonyan and Zisserman (2014) perform. They introduce a Two-Stream Convolutional Network which takes one single RGB frame for one stream, and for the other stream, several frames of the computed OF. Each stream is respectively called "Spatial stream ConvNet" and

"Temporal stream ConvNet" and is represented in Figure 2.4.

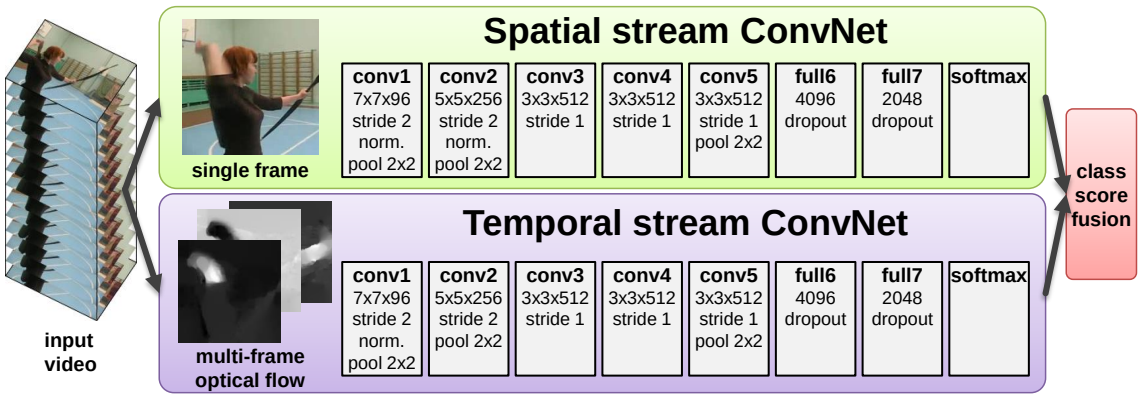


Figure 2.4 – Two-Stream 2D CNN (Simonyan and Zisserman, 2014).

They notice that "Temporal stream ConvNet" reaches much better performances compared to "Spatial stream ConvNet". This could easily be explained, as in addition to the dynamic genre of the input data, the temporal stream uses up to ten frames against only one for the spatial stream branch. Performances are much more alike when the temporal branch uses only one frame. Of course, the fusion of the two streams using a SVM method performs the best. Performances for each modality on UCF101 dataset are depicted in Table 2.1. Their work is later adapted with the aim to perform real-time action classification (Zhang et al., 2016) by replacing optical flow with motion vectors obtained directly from the compressed videos; and also recently by Zhao et al. (2020) for performing joint action localization and recognition.

Singh et al. (2016) proceed in a similar way to take into account the temporal dimension. They train different VGG networks (Simonyan and Zisserman, 2015), with different inputs. The inputs are six consecutive frames from the RGB channels, computed OF, and both of their Region-of-Interest (ROI) (around the person performing the action). This Multi-Stream Network (MSN) feeds then a FC layer which feeds itself a Bi-directional LSTM network (Graves et al., 2013). The LSTM network is fed by the consecutive outputs of the MSN.

Sun et al. (2015) present their Factorized Spatio-Temporal Convolutional Networks (FSTCN) which performs 2D convolutions and temporal convolution on the features transmitted into the network. By using pre-training on ImageNet and fusing the output probabilities obtained for each video segment using Sparse Concentration Index (SCI) method (Wright et al., 2009) that takes into account sparsity and correlation of the data, they achieve 88.1% of accuracy on UCF101 dataset.

Li et al. (2016) present their Tube Convolutional Neural Network (T-CNN) which can be decomposed into two distinct networks. They first create motion segmented tubes using Residual Convolutional Neural Network (R-CNN) (Ren et al., 2017). Then those tubes feed a VGG-like network using 20 motion amplitude frames distributed along the channel dimension. A similar method is also proposed but using

3D convolutions and is presented thereafter.

Long-term Recurrent Convolutional Network (LRCN) models are introduced by Donahue et al. (2017) for action recognition. They extract features using 2D CNN for each image which feed a LSTM from start to end. The decision is based on the average score. This simple model is tested with OF and a single RGB image. The fusion of the two modalities perform obviously the best. Similarly, a Temporal Segment Network (TSN) (Wang et al., 2016, 2019b) extracts features from RGB frames, OF and the RGB difference between two consecutive frames, through the same 2D CNN and their features are aggregated for decision. Their method is applied also for classification of untrimmed video (Wang et al., 2017b).

Xu et al. (2018) introduce densely-connected dilated convolutions layers inspired by Huang et al. (2017) in 2D for action classification. To do so, they use as backbone the TSN network to extract spatio-temporal features of every snippet (Wang et al., 2019b) and feed them to their network. They reach an accuracy of 96.2% which is 1.2% more than the TSN model.

Another method (Bilen et al., 2018) uses dynamic images as input of an aggregated ResNet: ResNeXt (Xie et al., 2017). Examples of dynamic images are presented in Figure 2.5.

Figure 2.5 – Examples of dynamic images used as input of the ResNeXt model (Bilen et al., 2018). From left to right, they represent the actions: “blowing hair dry”, “fencing” and “balancing on beam”.

Dynamic images computed from stacked RGB images are a way to encode the temporal information and therefore to provide video representation in one image. They use also RGB images, OF and dynamic OF (same as dynamic images but with OF), which feed different CNN. By averaging their scores, they reach an accuracy of 95% on the UCF-101 dataset. Models are pre-trained before on the ImageNet dataset Russakovsky et al. (2015). Similarly, Safaei et al. (2018) develop a CNN to predict dynamic images from RGB and the skeleton image (estimated pose (Newell et al., 2016) projected to an image), which feed another CNN to predict the action performed.

Lin et al. (2019) presented recently the Temporal Shift Module (TSM). Those modules use past and future features within the network and add them to the current extracted features to perform classification. They manage to keep 2D convolution complexity while performing like state-of-the-art 3D convolution methods. The

method is used on pre-existing CNN such as ResNet-50 He et al. (2016). Similarly, Luo and Yuille (2019) use spatio-temporal aggregation by decomposing the feature channels into spatial and temporal groups in parallel. Recently, Sudhakaran et al. (2020) developed Gate-Shift Module (GSM) in order to turn a 2D-CNN into a spatio-temporal feature extractor. To that aim, they use a spatio temporal convolution with *tanh* activation and temporal shift on the 2D-extracted features for feeding the GSM, as presented in Figure 2.6.

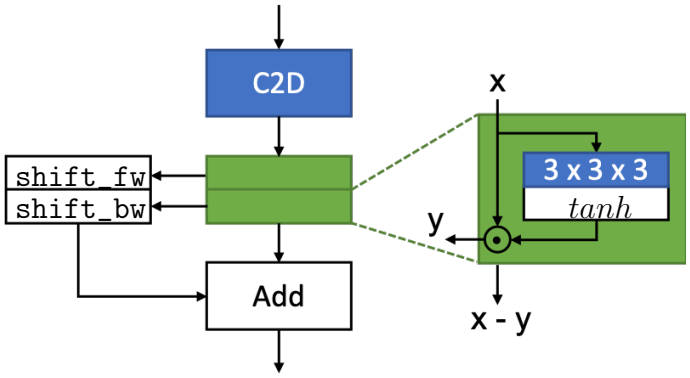


Figure 2.6 – Gate-Shift Module (GSM) architecture with forward and backward temporal shift.

They improve the performances of most of the 2D-backbones used (models for feature extractor) by using 3D convolutions on the extracted features. This proves that 3D convolutions better extract temporal information. This leads to now focus on 3D CNN based method for action recognition.

3 3D Convolutional Neural Networks for Action Classification

The first dedicated 3D CNN applied to action recognition has been proposed, as far as we now, by Kim et al. (2007). They consider actions as 3D volumes on which they apply CNN techniques. Their work then is extended by Ji et al. (2010, 2013). The method is applied to the KTH dataset and TRECVID data (Over et al., 2008). They were using 3D convolution filters but 2D subsampling on the obtained feature maps to keep the same temporal dimension. Their performances on KTH dataset did not, at that time, outperform yet handcrafted features methods.

In Baccouche et al. (2011), a 3D CNN, depicted in Figure 2.7, coupled with an RNN using one LSTM cell is used for classifying actions from the KTH dataset. Reported performances still do not outperform engineered features (Gao et al., 2010).

The Convolutional 3D (C3D) model is presented by Tran et al. (2015), which consists of 8 consecutive convolutional layers using $3 \times 3 \times 3$ kernel and five max-pooling layers. Three models are pretrained on different datasets: Sports-1M dataset

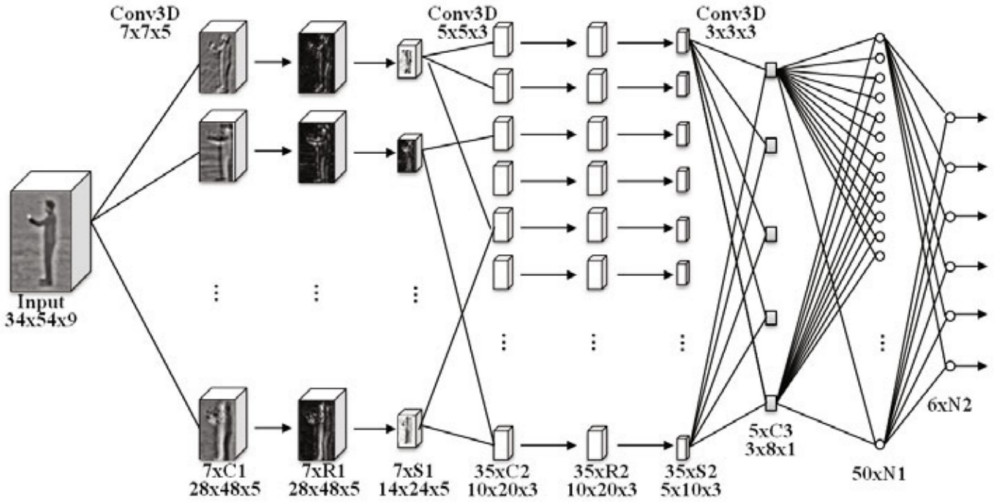


Figure 2.7 – Comparison of V1 and Slowest Features of V1 (SF1) on Yupenn and Maryland datasets with SF1 localisation and value represented on input (Wiskott and Sejnowski, 2002).

(Karpathy et al., 2014), I380K dataset (intern dataset) and I380K fine-tuned on Sports-1M (Karpathy et al., 2014). From those features and the IDT feature from UCF101 dataset classified using SVM, they reach 90.4% of accuracy. Lima et al. (2017) apply similar architecture but more shallow, and obtain an accuracy of 97.6% on UCF50 dataset. A similar architecture was also used for infrared images and OF data for action classification (Jiang et al., 2017).

C3D is also used by Hou et al. (2017) for their T-CNN. As in previous section, it computes action tubes but using 3D convolutions. Videos are first divided into equal length clips and are segmented using 3D R-CNN to create tube proposals. Tubes are then linked together. The pipeline of this method is presented in Figure 2.8.

By using the features extracted from the segmented video tubes with C3D model, they increase the performance compared to a direct application of the C3D model.

Feichtenhofer et al. (2017) update what was done in 2D by Simonyan and Zisserman (2014), in 3D to introduce their Spatio-Temporal Residual Network (ST-ResNet). They replace simple CNN branches by R-CNN with one connection between the two branches. Their results prove that RGB stream processed alone gets better performances than the OF stream. By processing them together, they reach an accuracy of 93.4% on UCF101 dataset.

A major breakthrough was proposed by the method of Carreira and Zisserman (2017), with much higher scores obtained on action classification. They present their Two-Sream I3D model as the combination of RGB-I3D and Flow-I3D models trained separately. The architecture of the model is presented in Figure 2.9.

Each of their models uses inflated inception modules, inspired from the 2D inception modules Szegedy et al. (2015). The major strength of their model is the

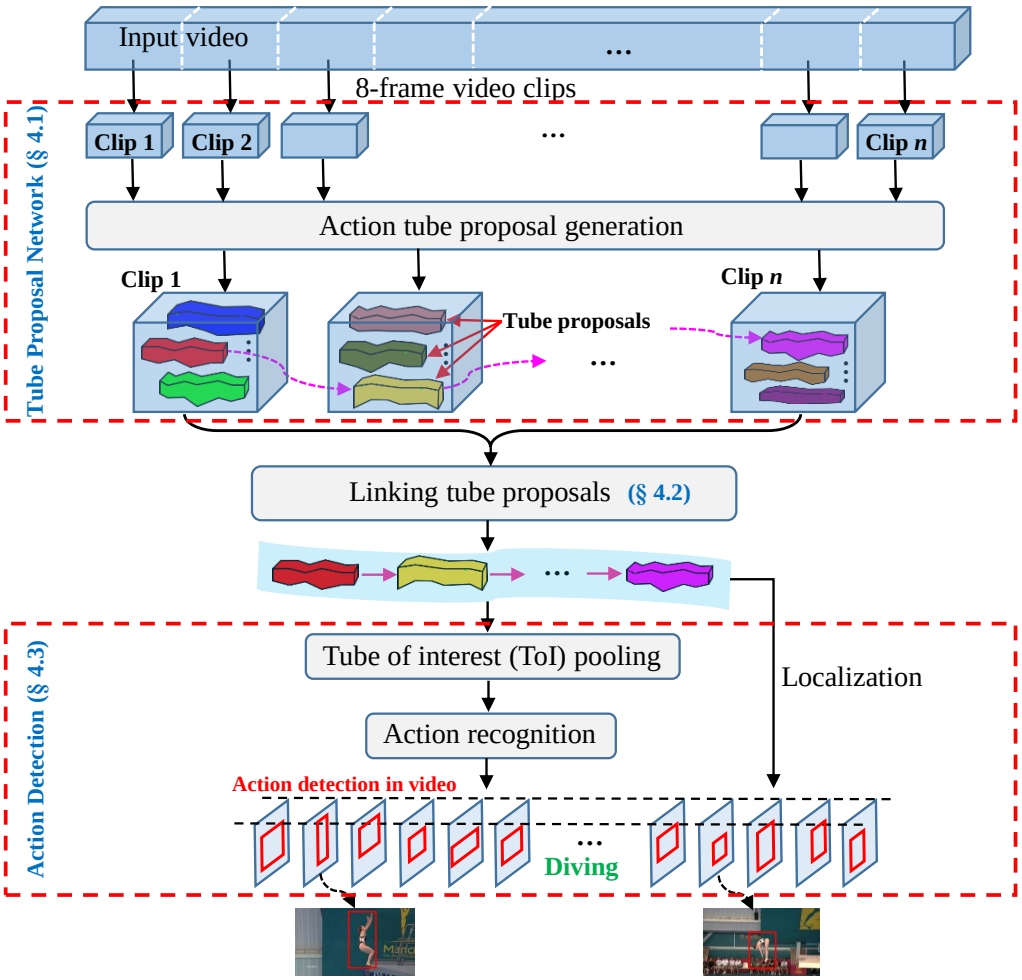
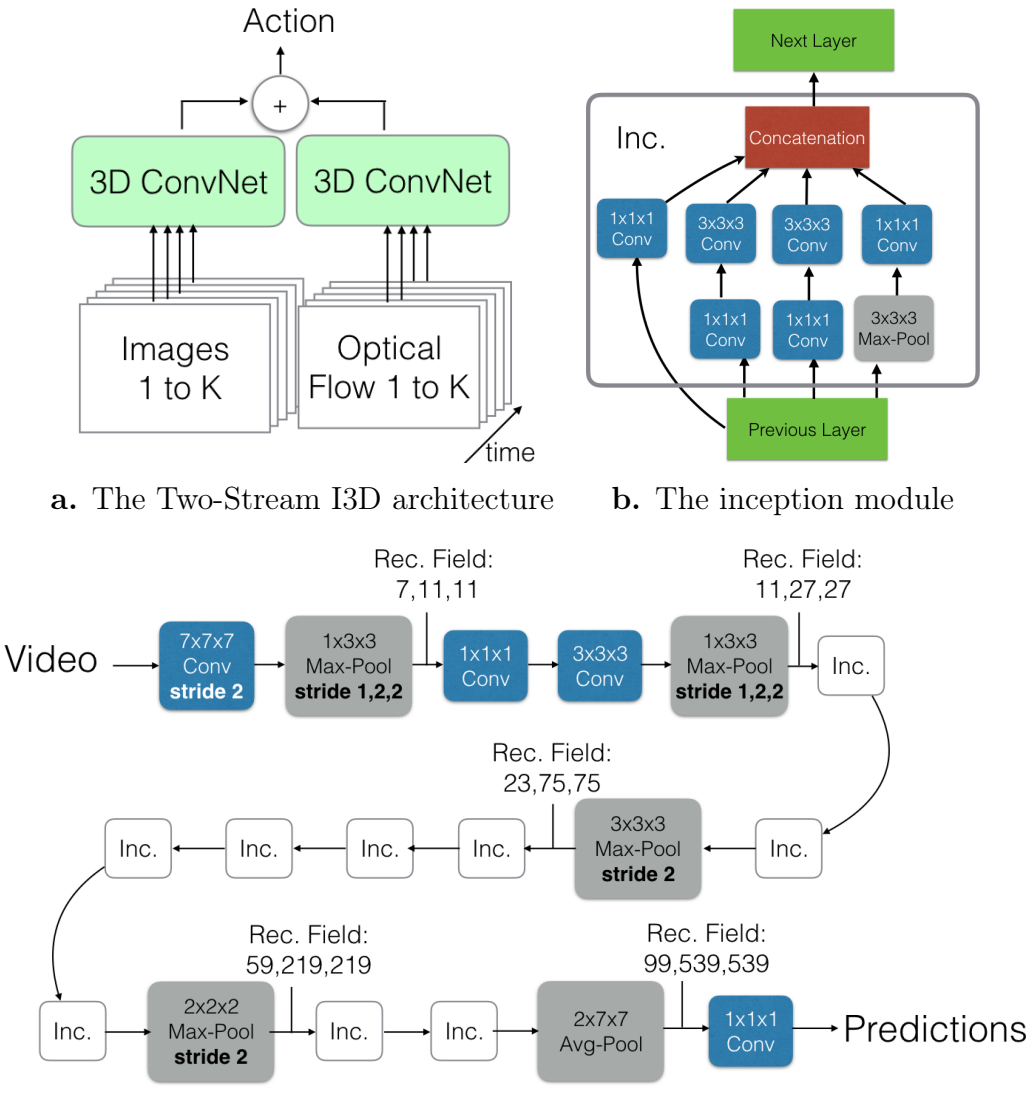


Figure 2.8 – T-CNN pipeline (Hou et al., 2017).



a. The Two-Stream I3D architecture **b.** The inception module

c. The Inflated Inception-V1 architecture

Figure 2.9 – The I3D models (Carreira and Zisserman, 2017).

pretraining on ImageNet first (Deng et al., 2009) and then on Kinetics-400 dataset (Kay et al., 2017), more complex than UCF101 with 400 classes. By using Kinetics, they boost performances from 93.4% to 98% on UCF101. They reach also 74.2% of accuracy on Kinetics-400 dataset. Inception modules have already proven their efficiency for image classification on ImageNet dataset (Russakovsky et al., 2015), and have evolved through time. The first version inception_v1 presented along with GoogLeNet by Szegedy et al. (2015) have evolved to inception_v2 and inception_v3 (Szegedy et al., 2016). Inception_v1 has been used for the I3D models. The inception_v2 module was used by Jiang et al. (2018) for extracting features, along with directional LSTM (Graves et al., 2013). The goal was to create an intelligent fashion consultant system able to generate stylish outfits for given items. After reaching the limit of the UCF101 dataset, the scientific community started using Kinetics-400 to benchmark action recognition methods. Fan et al. (2019) propose an architecture which processes in parallel low and high resolution videos and reaches 73.5% of accuracy on Kinetics-400.

Long-term Temporal Convolutions (LTC)-CNN were introduced by Varol et al. (2018). They experiment different temporal size for input video clips, in order to improve classification (Figure 2.10).

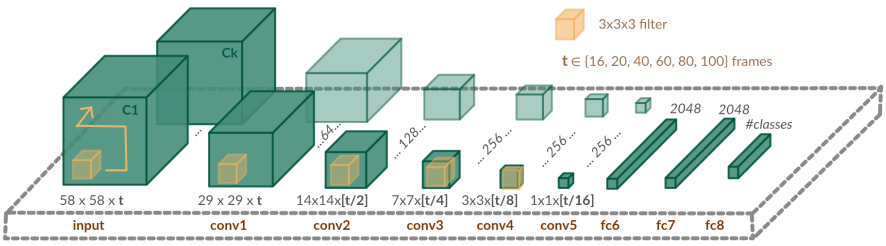


Figure 2.10 – LTC-CNN architecture (Varol et al., 2018).

Better accuracies are obtained when considering a greater number of frames as input, especially on long lasting actions which have a longer temporal support. On the other hand, Tran et al. (2018) present their R(2+1)D model, which decompose 3D convolution by tensorial product of 2D spatial convolution and 1D temporal convolution. They reach 97.3% on UCF101 using a Two-Stream configuration fed by RGB and OF streams.

Ng et al. (2018) present ActionFlownet in order to predict jointly the OF and action class from RGB images. They use transfer learning to predict OF and fine-tune the models according to the considered action datasets. Even if the method does not reach state-of-the-art results, it achieves a reasonable score of 83.9% of accuracy compared to other methods which do not use pre-training. Same motivation has also driven Crasto et al. (2019) who train a 3D ResNet (Xie et al., 2017) from the RGB stream in order to mimic OF features. This model, so called Motion-Augmented RGB Stream (MARS), outperforms classical independent approaches and reaches 97.1% of accuracy using a temporal window of 64 frames. Liu and Hu (2019) try

also to mimic the motion information using Spatio-Temporal Relation Networks (STRN) from only RGB data. RGB features are extracted using 2D CNNs representing the appearance stream and a motion stream is built using a 3D CNN from the RGB stream. Connections between feature are made through "relation block" going from appearance stream to motion stream. Features are fused before decision and the streams are trained together. This method reaches an accuracy of 91.8% on UCF101.

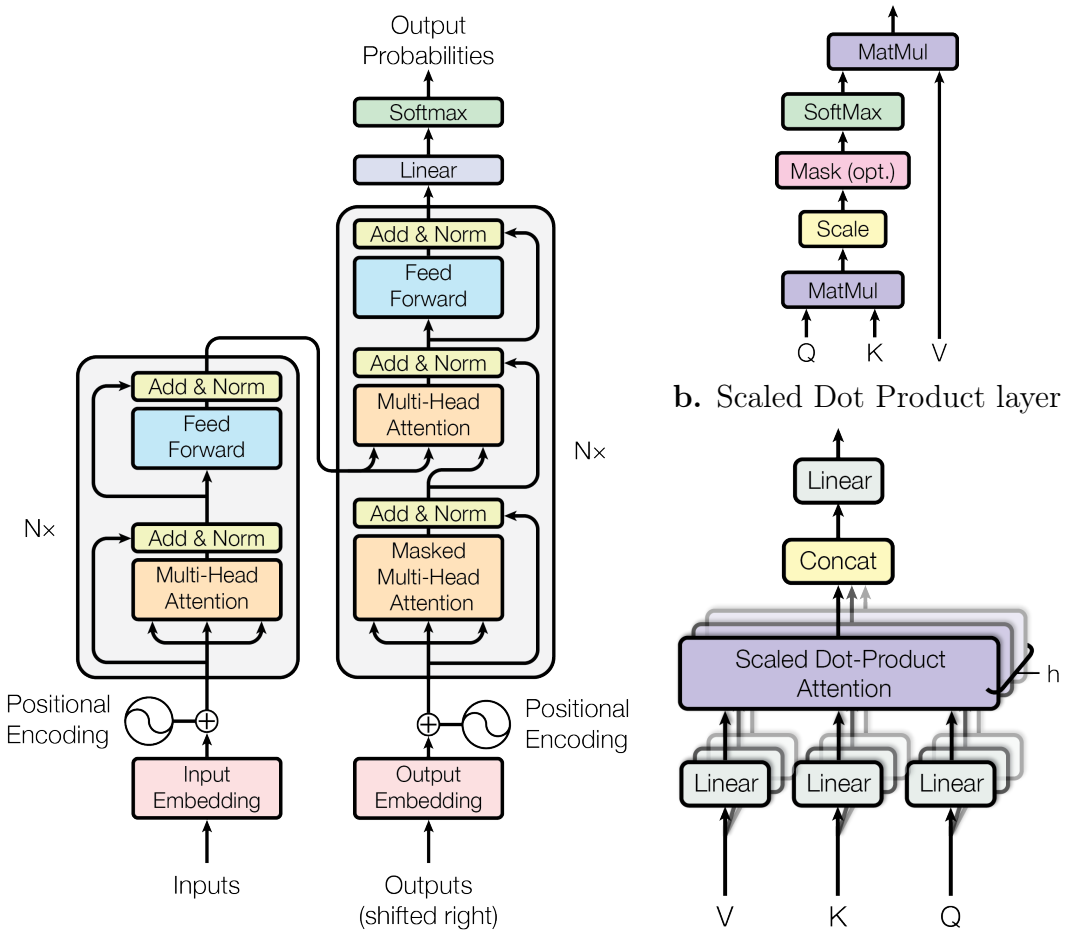
Wang et al. (2018b) introduce Spatial Temporal Pyramid Pooling Layer (STPP) layer using 3D convolutions in a two-stream like network fed by RGB and OF streams. The output becomes the input of a LSTM network. The use of LSTM allows classification of videos of arbitrary size and length. Each modality performs similarly: 85% and 83.8% of accuracy for RGB and OF stream respectively. When fused together, the method reaches 92.6% of accuracy.

Recently, Nogas et al. (2020) propose a fall detection method using spatio-temporal convolutional auto-encoders. The principle of auto-encoder is to reconstruct the given input after reducing its dimension through convolution and pooling layers. The reconstruction from inner features are done using interpolation methods such as 3D convolutions coupled with 3D up-samplings. In order to perform fall detection, the authors train their model on Activities of Daily Living (ADL), which does not contain any falls. At test time, the falls are detected when the reconstruction error is high. The method is tested on several dedicated datasets such as SDU dataset (Ma et al., 2014), UR dataset (Kwolek and Kepski, 2014) and Thermal Fall Dataset (Vadivelu et al., 2016).

Kalfaoglu et al. (2020) introduce the Bidirectional Encoder Representations from Transformers (BERT) layer to better make use of the temporal information of BERT's attention mechanism firstly used for language understanding (Devlin et al., 2019; Vaswani et al., 2017). The BERT layer is based on the use of the Multi-Head Attention layer (Figure 2.11.c), which uses Scaled Dot Product layer (Figure 2.11.b). The Multi-Head attention layer is part of a bigger network, the Transformer model (Figure 2.11.a) which is dedicated to translation tasks.

The incorporation of the BERT layer in the REsNeXT, R(2+1)D and I3D models, previously described, improve their performances. They reach the state-of-the-art results on both HMDB51 and UCF101 datasets with respectively 85.1% and 98.7% of accuracy using the R(2+1)D architecture (Tran et al., 2018). It is a ResNet-type architecture with separable temporal and spatial convolutions and a final BERT layer in order to better use the obtained features. One important point to stress is also the use of IG65M dataset (Ghadiyaram et al., 2019a) for pre-training their model. IG65M dataset is build from the Kinetics-400 (Kay et al., 2017) class names. Those class names are then used as hashtags on Instagram and lead to 65M clips from 400 classes. Their dataset is however not publicly available.

Very recently an article was submitted to ICLR 2021 OpenReview.net (2021): "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale". The paper is currently under double-blind review process and available on OpenRe-



a. Transformer model

b. Scaled Dot Product layer

c. Multi Head attention layer

Figure 2.11 – Transformer architecture and its layers (Vaswani et al., 2017).

view[1]. Their work has been noticed because they use image attention for classification task without using any convolutional layers, by using "only" standard transformer encoder. Their Transformer encoder is inspired by Vaswani et al. (2017) whose model is presented in Figure 2.11. Developed by Google, they apply their method on the WMT 2014 English-to-German and English-to-French translation tasks (Association for Computer Linguistics, 2014) [2]. The method is based on blocks of parallel linear layers using dot product between their output, concatenation of the parallel branches followed by a final FC layer. Between these blocks, residual connection are used. Comparable approaches are used for similar tasks (He et al., 2020b). In "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale", the presented transformer is adapted for image classification task. Images are divided into patches and fed to a similar network, which also embeds the patch position. No convolution layers are implied. The authors notice that the use of the feature maps from pretrained model (He et al., 2016) instead of the image is also possible and increases performances. Classification scores match or exceed the state of the art methods on image classification task for several datasets. Moreover, the authors stress the fact that such model is cheap to pre-train and that their scalable design does not introduce image-specific inductive biases for decision. Their results let us question on the future use of CNNs for image or video classification.

4 Conclusion and Discussion

This chapter was focused on the overview of NN classifiers for the problem of action recognition in videos. Having reviewed a large variety of the numerous published work so far, we come to the conclusion that 3D CNNs outperform 2D frame-based CNNs yielding better classification scores.

The second statement we can do is that the two stream model based on color and motion information is the way to go for classification of actions. Indeed, in Table 2.1, we can notice the best performances are obtained when both modalities are considered. This is why in the present work we choose to focus on 3D spatio-temporal CNN. Chapter 4 presents our work using 3D based CNN with an RGB stream while Chapter 5 presents the one using OF stream. Also, from this overview, we noticed how important it was to consider both modalities at the same time during the training process. It motivates the design of a Two-Stream based architecture that we call Twin due to the same architecture of each branch. This is the subject of Chapter 6. We analyse the obtained features in Chapter 7 with a method we are proposing. State of the art methods also motivate our work on attention mechanism which is the subject of Chapter 8.

We have also seen that pre-training could lead to different performances according to the dataset used for this process. The same drawback is noticed by

[1] https://openreview.net
[2] http://www.statmt.org/wmt14/translation-task.html

Table 2.1 – Performances of the different reviewed model on UCF101 (Soomro et al., 2012).

Models	Input	PreTrain	Acc in %
Spatial stream ConvNet 2014	RGB	No	52.3
Spatial stream ConvNet 2014	RGB	Yes	72.8
Temp. stream ConvNet 2014	OF	No	73.9
Temp. stream ConvNet 2014	10 stacked OF	No	83.7
Two-Stream 2D 2014	RGB + 10 stacked OF	Yes	88
FSTCN 2015	RGB stream	Yes	88.1
C3D + SVM 2015	RGB stream	Yes	85.2
C3D + IDT + SVM 2015	RGB stream	Yes	90.4
T-CNN 2D 2016	20 stacked OF	Yes	92.3
T-CNN 3D 2017	RGB stream	Yes	87.5
LRCN 2017	RGB	Yes	68.2
LRCN 2017	OF	Yes	77.3
LRCN 2017	RGB + OF	Yes	82.3
ST-ResNet 2017	RGB stream	Yes	82.3
ST-ResNet 2017	OF stream	Yes	79.1
ST-ResNet 2017	RGB stream + OF stream	Yes	93.4
RGB-I3D 2017	RGB stream + OF stream	Yes	95.6
Flow-I3D 2017	RGB stream + OF stream	Yes	96.7
Two-Stream-I3D 2017	RGB stream + OF stream	Yes	98
RGB-Stream-Hakan 2017	RGB	Yes	87.6
Flow-Stream-Hakan 2017	OF	Yes	84.9
Four-Stram-Hakan 2017	RGB+OF+D-RGB+ D-OF	Yes	95.4
ActionFlowNet 2018	RGB stream	No	83.9
TSN 2019b	RGB + OF	Yes	95
TSM 2019	RGB	Yes	95.9
MARS 2019	RGB stream	Yes	97.1
R(2+1)D 2018	RGB stream + OF stream	Yes	97.3
STRN 2019	RGB stream	Yes	91.8
R(2+1)D + BERT 2020	RGB stream	Yes	**98.7**

Ghadiyaram et al. (2019b). The authors stress that pre-training method leads to very strong feature representations for action recognition. The models can thereupon be fine-tuned for the dedicated task (Donahue et al., 2014). Since we designed our model ourselves and that pre-training would require high capacity in terms of GPUs and disk usage, all presented models are trained from scratch. It also makes the comparison between methods easier, since no dataset biases are added (Khosla et al., 2012). We will be using as our baseline the Two-Stream I3D method (Carreira and Zisserman, 2017), which was the state-of-art method when this thesis started and is still a reference method, and whose codes is publicly available.

Next chapter describes the most used video datasets for action recognition and presents the `TTStroke-21` dataset developed during the project for classification of table tennis strokes.

Chapter 3

Datasets for Action Classification

1 Introduction

The need of datasets for action recognition has grown those last years, especially because methods are performing better year after year. These datasets can change in terms of number of videos starting from a few videos up to millions of videos. In addition to their size, the number of categories and their complexity also vary from few classes up to few hundreds, or even thousands in some cases. Each dataset can be labelled with annotations either by enriching the terminology, localising the action in space and time or by adding modalities information such as joint skeleton.

In addition, the action classification task can be split in several categories according to the targeted application. In videos, one might want to focus on action localization (Weinzaepfel et al., 2015; Qiu et al., 2018; Jain et al., 2017), in space and/or in time, with the aim to spatially track different individuals and also their actions over time. Another aspect of action classification is prediction of future events (Guen and Thome, 2020; Lu et al., 2019; Akbarian et al., 2017). Finally, fine-grained action classification focuses on recognition of different actions that are very similar, which is not the case, for example, of UCF101 dataset. In our case, we focus on sport and the objective is to be able to segment in time and classify the different strokes performed in table tennis videos recorded during matches or training sessions. The final goal is to offer a platform open to players and coaches to analyse player performances, with indexed profile and statistical performances. It could be improved later by trajectory modeling of the ball (Calandre et al., 2020; Wu and Koike, 2020), skeleton modeling (Morel et al., 2017) or segmentation (Voeikov et al., 2020), in order to adapt training sessions and give an automatic feedback to the players.

Section 2 presents the different annotation approaches to build a dataset. Some are based on voluntary work (most of the time carried out by the students or the research team), or by hired people, which implies funds from an organization. Therefore one will notice disparity between the presented datasets and the precision in their annotation. In Section 3, different datasets are presented. An overall table presents their specificity such as the number of classes, acquisition process and number of videos (Table 3.1). Then, Section 4 introduces the dataset developed for our field of interest, i.e. stroke classification in table tennis from videos: `TTStroke-21`. Conclusions are drawn in Section 5.

2 Annotation Processes

One can distinguish two ways to annotate a dataset: automatic or by hand. Each modality can be split or merged. It is usual to have first an automatic tool to have candidates and then hand-label them. Also, it is common to have a dataset split in "auto" and "clean" sets. The "auto" being the one annotated using automatic methods and "clean" the one automatically labeled, verified and adjusted by hand. The two annotation processes are first described before presentation of the datasets.

2.1 Automatic Annotation

Tags from social network platforms can be used to build a rich dataset. It is the case, for example, with the IG65M dataset (Ghadiyaram et al., 2019a), used for pre-training the R(2+1)D (Tran et al., 2018) combined with BERT layer (Kalfaoglu et al., 2020). As presented in last chapter, IG65M dataset is built from the Kinetics-400 class names. The class names are then used as hashtags on Instagram and led to 65M clips from 400 classes. Such methods are powerful since they reach the state-of-the-art performances for action classification using this dataset to pre-train their model. However their dataset is not publicly available. Such annotation process requires filtering in order to refine the annotations. Seymour and Zhang (2018) focus on such filtering process in order to build a dataset of images.

In movies, the script can also help to automatically label the sequence. It is for example what Marszalek et al. (2009) have done for the Hollywood2 dataset. They generate the samples this way and clean them manually for the test set.

Similarly, datasets can be constructed from the description of the videos from online platforms hosting them (Chesneau et al., 2018). Such a method is presented in Figure 3.1.

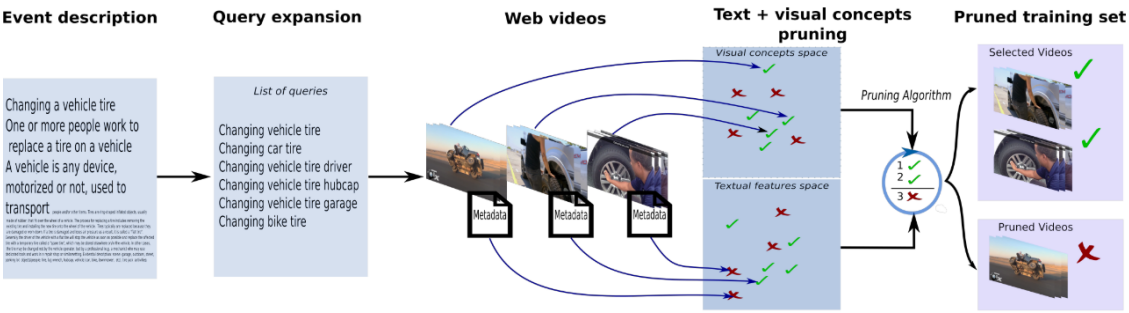


Figure 3.1 – Overview of an automatic annotation method for online videos (Chesneau et al., 2018).

Then, according to a description, datasets can be generated in an automatic way. This method was initially designed in order to expand training data for event detection task in TRECVID (Over et al., 2011a).

2.2 Manual Annotation

The most common way to annotate a dataset, especially when this one is not large, is to label all the samples by hand . This is done often through an annotation platform with the help of students or people working on the project. Some tools might be used to help in the process such as a pre-classification if a model already exists, or localization segment candidates of the actions when the video is untrimmed. One can distinguish two ways in the hand-labelling process: if the annotation is done by one person or several. By one person, the risk is that an inattention might lead to errors in the dataset or make it biased according to the point of view of the annotator. To overcome this issue, a crowdsourcing method can be used.

Crowdsourcing is based on the annotation of the same segments by different persons. It relies on the collective intelligence and should give better results than with only one person annotating. Outliers annotations are not considered in the final decision. Different rules might apply, e.g take the mean of the annotators when possible or consider only the annotator that performed the best today; in order to take an annotation decision. There are also datasets which provide gross crouwdsourced annotations and it is the team working on the dataset that decides which decision to make.

A new trend appeared recently: the use of Amazon Mechanical Turk (AMT)[1]. AMT, also called "MTurk", is a crowdsourcing marketplace for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these *micro-tasks* virtually. Here it is applied to annotation and AMT are paid according to the number of annotations performed. It started to be used with ImageNet dataset (Deng et al., 2009) dedicated to image classification with 3.2 millions of images over 5 247 classes. The annotation process is the same method than in ImageNet. It then spread for image and video annotations (Heilbron and Niebles, 2014; Vondrick et al., 2010). The strategy is in two folds:

- search the web for video candidates related to the dataset taxonomy
- AMT workers refine the candidates

AMT workers verify the presence of the action in the video candidates and they can also temporally annotate them. An AMT platform is represented in Figure 3.2.

It is often coupled with a crowdsourced method meaning that each video will be annotated by several AMT workers. This allows the construction of a large dataset in a short amount of time. However this can be done only with the appropriate funding.

[1] https://www.mturk.com/

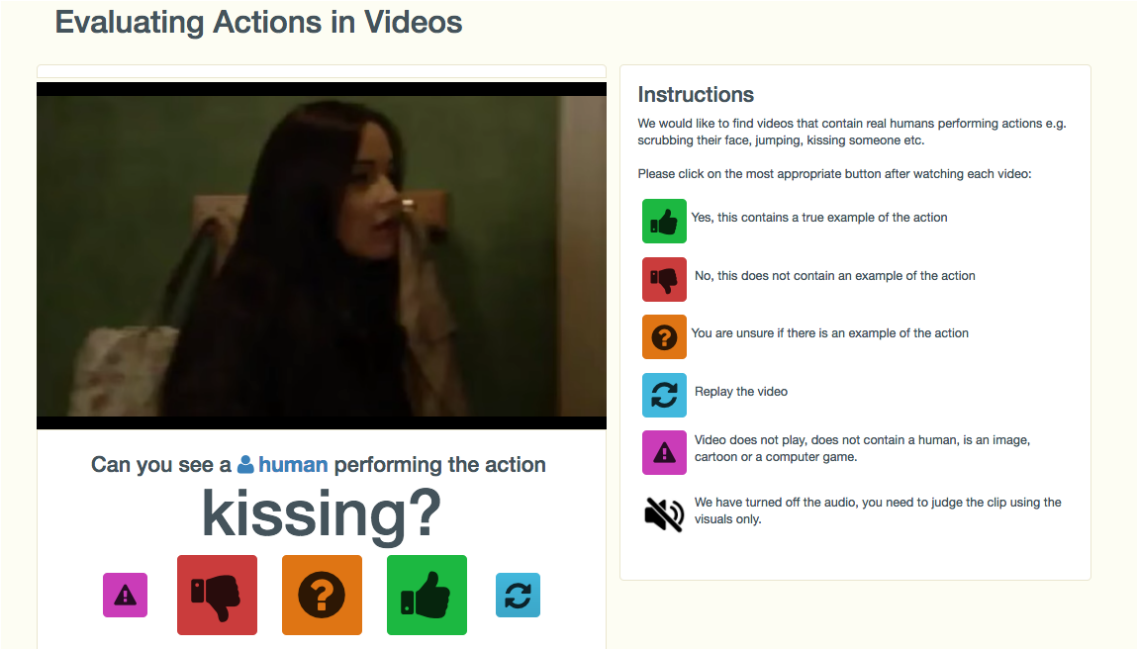


Figure 3.2 – AMT platform used for Kitenics datasets (Kay et al., 2017).

3 The Datasets for Action Classification

Datasets of actions can be categorized in many ways. In this section, the datasets are clustered according to the acquisition process: acquired in a controlled environment, film based or "in the wild". Of course, this clustering method is not perfect since some datasets mix different types of videos.

3.1 The Acquisition-Controlled Datasets

These datasets are often self made by the authors and they decided in what type of environment the actions will be performed. It does not always mean that the dataset is easier than "in the wild" since difficulties can be added on purpose. The databases from broadcasts or recordings not meant for action recognition task are too considered in this subsection because the acquisition environment can be taken into account in the classification process.

CMU-Pittsburgh AU-Coded Face Expression Image Database

Kanade et al. (2000) introduce the "CMU-Pittsburgh AU-Coded Face Expression Image Database"[2], so called here "Coded-Faces". Coded-Faces is based on the Facial Action Coding System (FACS) (Stöckli et al., 2018) which is a human-observer based system designed to detect subtle changes in facial features. It consists of 44 action

[2] https://www.cs.cmu.edu/~face/database.htm

units which can be described by muscle movements. The database is constituted of 1 917 video clips from 182 men and women recorded in a control environment. They made an extended version with 210 subjects. It may not be used a lot recently, but it gave first clues on how to perform classification. Indeed, as presented in Figure 3.3, the tracking and segmentation of some face parts or muscles might help to solve this classification problem.

Figure 3.3 – Mouth segmentation from a sample of AU-Coded Facial Expression Image Database (Kanade et al., 2000).

Ballet, Football and Tennis Datasets

Efros et al. (2003) introduce several datasets in order to evaluate action classification methods[3]. The different datasets, labeled by hand by the authors, are represented in Figure 3.4

a. Ballet dataset **b.** Football dataset **c.** Tennis dataset

Figure 3.4 – Different datasets introduced by Efros et al. (2003).

Ballet dataset consists of choreographed actions recorded using stationary camera in a controlled environment. Videos are from two men and two women performing mostly standard ballet moves. The database comprises 24 800 frames with 16

[3] http://graphics.cs.cmu.edu/people/efros/research/action/

classes. The authors stress that the controlled environment and the choreographed nature of the actions make this classification task easier.

Football dataset is built from the extracted tracked players of the World Cup football games from an National Television System Committee (NTSC) video tape. Therefore it is not acquired in a fully controlled setting but rules can be defined from the broadcast acquisition conditions and might help in the classification process. As one can see in Figure 3.4, the images are blurred and pixelated. This is due to the recordings made with a moving camera in order to follow the game and the players. The dataset is composed of 4 500 frames from 72 tracked sequences. The taxonomy has eight classes and remains very simple: "run left 45°", "run left", "walk left", "walk in/out", "run in/out", "walk right", "run right" and "run right 45°".

Tennis dataset is composed of 6 415 frames from videos of two tennis players. It has six easily recognisable classes: "swing", "move left", "move right", "move left and swing", "move right and swing" and "stand". It has been recorded with two amateur players playing on a tennis field. The scene is visible in Figure 3.4.

KTH

As presented in the first chapter, KTH dataset was introduced by Schüldt et al. (2004)[4]. KTH stands for "Kungliga Tekniska Högskolan" in Swedish which is the Royal Institute of Technology (Stockholm, Sweden), institution of the authors. The dataset, depicted in Figure 3.5, is composed of six classes: "Walking", "Jogging", "Running", "Boxing", "Handwaving" and "Hand clapping".

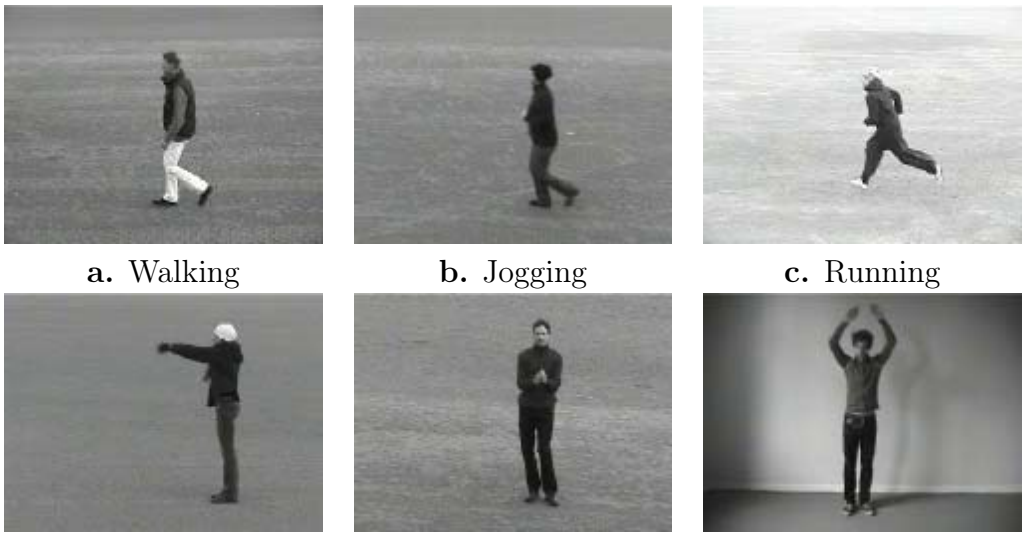
The acquisition was done in a controlled environment, homogeneous background, static camera at 25 frames per second (fps), with 25 actors and has 2 391 video clips across 600 videos. Videos are recorded outdoors and indoors.

Weizmann

Gorelick et al. (2007) present the Weizmann action dataset[5]. It is constituted of 81 video sequences recorded at 25 fps at low resolution (180×144). Each video contains a different person performing one of the nine following actions (used as the class): "running", "walking", "jumping- jack", "jumping-forward-on-two-legs", "jumping-in-place- on-two-legs", "galloping-sideways", "waving-two-hands", "waving-one-hand" and "bending".

The KTH and Weizmann datasets were the most popular at the early ages of action recognition research. Despite their simplicity, some amount of researchers continued using them as a benchmark. However, these datasets are largely outdated with regard to the necessity of action recognition in "in-the-wild".

[4] https://www.csc.kth.se/cvap/actions/
[5] http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html



a. Walking **b.** Jogging **c.** Running

d. Boxing **e.** Hand clapping **f.** Hand Waving

Figure 3.5 – KTH dataset samples (Schüldt et al., 2004).

MSR

Cao et al. (2010) presented the MSR dataset[6] which includes three classes: "hand waving", "clapping" and "boxing". It contains 54 sequences acquired in different noisy scenes such as parties, outdoor traffic and walking people. The challenging aspect of this dataset is the activities in the background which might make the spatial and temporal segmentation and action recognition harder.

ACASVA

de Campos et al. (2011) from "Adaptive Cognition for Automated Sports Video Annotation" introduce a tennis action dataset so called "ACASVA"[7]. Their objectives is to evaluate classical action recognition approaches to player action recognition in tennis games. To serve their purposes, they collected data from TV broadcasts tennis. The videos are then spatially segmented on the players and temporally annotated using three classes: "hit", "serve" and "non-hit". The complexity of the the dataset remains simple. Their results are detailed in Chapter 1.

The Fall Datasets

Several datasets were created in order to perform fall detection task as referenced in Chapter 2. The SDU dataset (Ma et al., 2014) and UR dataset (Kwolek and Kepski, 2014) use Microsoft Kinect depth camera to record the videos. They both made acquisitions with different subjects performing different actions: "falling forward",

[6] www.microsoft.com/en-us/download/details.aspx?id=52315
[7] https://www.cvssp.org/acasva/Downloads.html

"falling backward", "bending", "squatting, "sitting", "lying" and "walking" for SDU; and "walking", "sitting down", "crouching down", "lying down in bed", "fall standing" and "fall sitting" for UR. Thermal Fall dataset (Vadivelu et al., 2016) is dedicated to pure fall action detection and was recorded using thermal camera and contains 44 videos out of which 35 have the falling action.

MERL Shopping

Singh et al. (2016) present the MERL Shopping Dataset which they have used in their experiments and shared to the scientific community. It is only 96 videos two minutes long recorded with a fixed camera in a grocery shop. Their field of interest is fine-grained action recognition and for this purpose they consider five classes: "Reach to Shelf", "Retract from Shelf", "Hand in Shelf", "Inspect Product" and "Inspect Shelf". All videos are annotated in time. Even if the authors stress the fine-grained aspect of their dataset, the actions are fairly easy to differentiate.

Datasets for Surgery

In term of action recognition in the medical field, surgery actions are also considered. Those have many applications: e.g assist surgeon, education in medicine university or follow the patient for better diagnostic after surgery. Petscharnig and Schöffmann (2017) built a dataset of nine hours containing 111 medical interventions such as: "Suction & Irrigation", "Suture", "Dissection", "Cutting", "Cutting (cold)", "Sling", "Coagulation" and "Injection". They also provide the "scene" which is, in this case, the body parts where the surgery takes place. Later, the Robust Medical Instrument Segmentation (ROBUST-MIS) challenge Roß et al. (2020) propose a new dataset for instrument tracking and segmentation for operation based on images extracted from 30 different surgical procedure videos.

Diving48

Diving48 dataset[8] presented by Li et al. (2018) is a fine-grained video dataset of competitive diving. It includes 18 000 video clips of major diving competitions retrieved online. The actions consist in dive sequences which can be decomposed in three parts: i) takeoff type, ii) movements performed in flight and iii) the entry in water. The combination of those three components lead to a total of 48 dive classes. Such organisation of a dive can be exploited to perform the classification.

Toyota Smarthome

Das et al. (2019) focus on the recognition of activities at home. The goal of the research is to assist, as fast as, possible if an accident happens. Toyota Smarthhome dataset is built from recordings of 18 senior subjects, filmed eight hours in one day

[8] http://www.svcl.ucsd.edu/projects/resound/dataset.html

with seven Kinect cameras located in different parts of the house. The face of the subjects has been blurred to preserved anonymity. The videos are then clipped into 16 115 activity segments across 31 classes. Their camera can also provide the depth along with the videos.

FineGym

FineGym dataset[9], introduced recently by Shao et al. (2020), is a fine-grained action dataset with a special focus on gym sport. The authors use a rich taxonomy to decompose each actom of a structured figure. They use three levels semantic and analyse four different gymnastic routines: balance-beam, uneven-bars, vault and floor exercise. They have a total of 530 element categories but only 354 have at least one instance. This rich amount of categories is due to all the combination of possible actoms. Such combination is represented through an overview of FineGym, Figure 3.6.

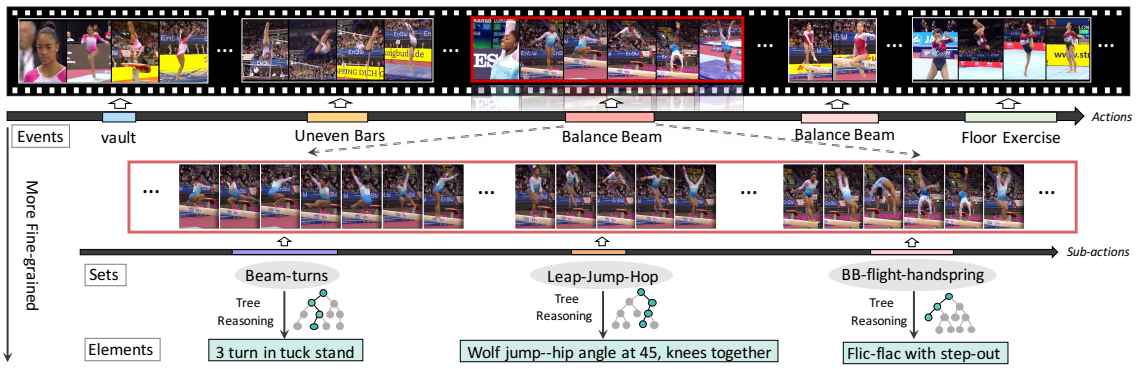


Figure 3.6 – FineGym dataset (Shao et al., 2020).

To alleviate this issue, the authors offer two settings: Gym288 with 288 classes but very unbalanced distribution and Gym99, more balanced but with "only" 99 classes. The total number of samples considering all classes reaches 32 697. The 708 hours of videos are hosted on YouTube with most of them in high resolution.

TUHAD

Lee and Jung (2020) too introduce a dataset dedicated to fine-grained action recognition but on Taekwondo sport: TUHAD. They recorded their own dataset with the help of ten Taekwondo experts and the use of two Kinect cameras with front and side view. They use a low number of classes with only eight Taekwondo moves. Those unit techniques are represented in Figure 3.7.

1 936 actions samples are obtained with depth and IR images along with the RGB data. The difficulty of the dataset seems questionable since the actions seem

[9] https://sdolivia.github.io/FineGym/

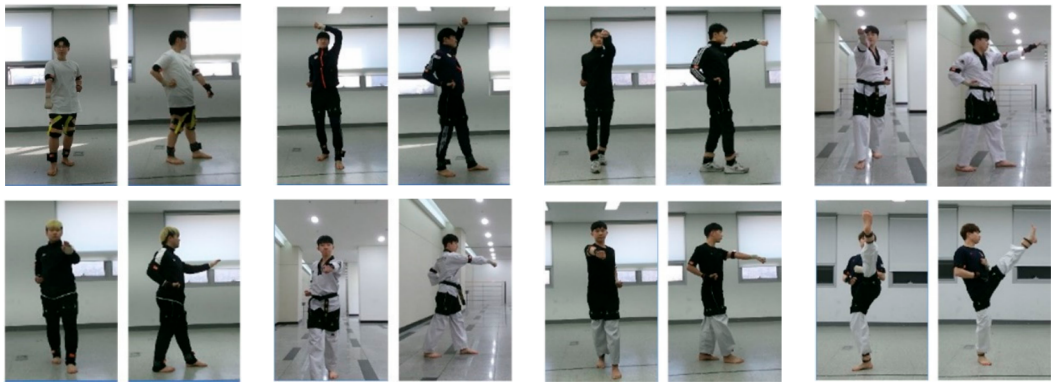


Figure 3.7 – The eight Taekwondo actions in Human Motion Taekwondo Unit Technique Human Action Dataset (TUHAD).

easy to be classified using only one image. Two classes are very similar in many ways but a foot position, which might be overcome with proper features.

3.2 Movie Based Datasets

Movie based datasets are from movies, trailers, often open-source so it can be redistributed. It can rely on the script to help in the annotation process.

Drinking and Smoking

Introduced by Laptev and Pérez (2007), the sequences are from the movie "Coffee and Cigarettes" from Jim Jarmusch, 2003. The movie itself is composed of 11 short stories across three short films where the actors share coffee and cigarettes. This dataset was designed for the joint action detection and classification for the two classes: "smooking" and "drinking" with respectively 141 and 105 samples.

DLSBP

DLSBP name comes from the first letter of authors last name who introduced the dataset: Duchenne et al. (2009). It contains three actions: "Stand up", "Sit down" and "Open door" extracted from 15 different movies, automatically annotated using script based method. They manually annotated 93 open door and 86 sit down actions in three movies: *Living in oblivion*, *The crying game* and *The graduate*.

Hollywood2

Marszalek et al. (2009) introduce the Hollywood2 dataset[10]. This dataset was designed for action and scene classification. It contains ten scene classes and 12 actions:

[10] https://www.di.ens.fr/~laptev/actions/hollywood2/

"Answer phone", "Drive car", "Eat", "Fight person", "Get out car", "Hand shake", "Hug person", "Kiss", "Run", "Sit down", "Sit up" and "Stand up"; over seven hours of video from 69 movies. Action samples of Hollywood2 datasets are presented in Figure 3.8.

a. Drive car **b.** Eat **c.** Run **d.** Hand shake

Figure 3.8 – Hollywood2 dataset samples (Marszalek et al., 2009).

They have a total of 1 694 actions samples. The difficulty lies in the fact that different actions can happen in the same sequence.

Charade

Sigurdsson et al. (2016) present a "Hollywood in Homes" to build their Charade dataset[11] and deserve to be in this section. The dataset is not really from movies but from 267 participants who agreed to make their own movie following a given script. It might imply bias according to how the participants understood the script. The dataset itself focuses on household activities with 157 action classes, 15 different scenes and also 46 object classes. It leads to a total of 9 848 annotated videos with an average length of 30 seconds (55GB). More recently, a similar approach has also given birth to the Charades-Ego dataset (Sigurdsson et al., 2018).

3.3 Egocentric Datasets

Egocentric datasets are recorded from the *first person* point of view. Such recordings often imply strong camera motion and a different perspective of action than a classic *third person* recorded dataset. Their applications are many but it tends to be healthcare applied (González-Díaz et al., 2018). It could allow to identify dangerous actions or situations for persons with a condition in order to prevent an accident or act faster if there is. These types of data are often used for LifeLog challenges (Dang-Nguyen et al., 2018; Münzer et al., 2018) too.

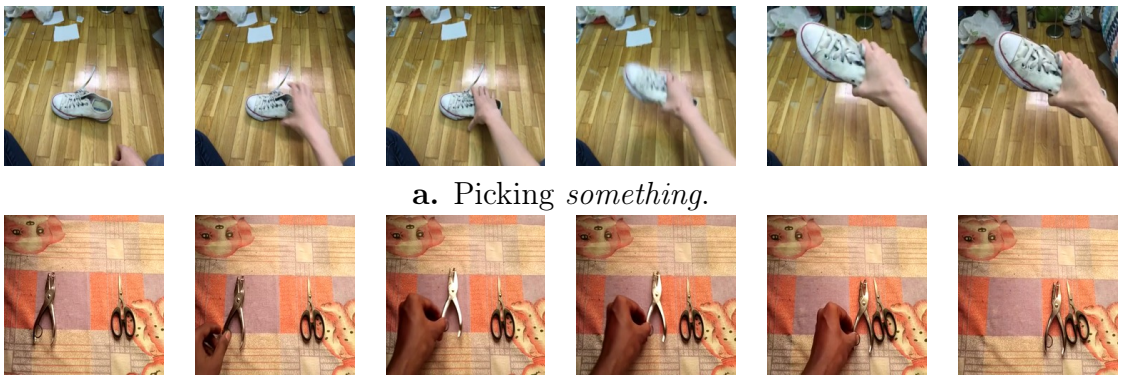
ADL

Cartas et al. (2017) build a new dataset for Dailylife activity recognition based on NTCIR-12 dataset Gurrin et al. (2016). This last one represents 89 593 egocentric pictures belonging to three persons over 79 days. In ADL dataset, they focus on a specific sample of 18 674 frames which they annotate using 21 activity categories.

[11] https://prior.allenai.org/projects/charades

Something-Something

Goyal et al. (2017) present the biggest egocentric dataset: Something-Something[12]. With 108 499 videos across 174 classes with objects interactions. Videos last approximately four seconds (mean value) with 620 videos per class. As presented in Figure 3.9, the goal is to classify the action performed but can also be to identify the objects on which it is performed.



a. Picking *something*.

a. Moving *something* closer to *something*.

Figure 3.9 – Something-Something samples, where *something* is also annotated and can be used to train joint caption and action recognition model (Goyal et al., 2017).

The difficulty of a such dataset lies in the intra-dissimilarity of the actions performed. Indeed, the use of different objects for the same action can have a completely different appearance, bringing much dissimilarity within the class. Hence, classification methods need to take this factor into account. Mahdisoltani et al. (2018) recently extended the dataset to 220 847 videos[13] with the same number of actions.

IXMAS

Mi et al. (2018) present the IXMAS dataset in order to train their model. Their work is also commented in previous chapter. Their dataset is based on 12 common human actions: "check watch", "cross arms", "scratch head", "sit down", "getup", "turn around", "walf", "wave", "punch", "kick", "pickup" and "point". The authors asked 11 actors to perform the actions three times while recording on four different sides plus a top view. The interest of their dataset lies in the different views which can be either treated separately or could be processed in parallel to perform action classification.

[12] https://20bn.com/datasets/something-something
[13] https://20bn.com/datasets/something-something/v2

Epic-Kitchens

Epic-Kitchens[14] was introduced by Damen et al. (2018) for the purpose of object detection, action classification and action anticipation. The specificity of this dataset is its egocentricity and its focus on activities in the kitchen. A timeline with action examples and objects segmentation is presented in Figure 3.10.

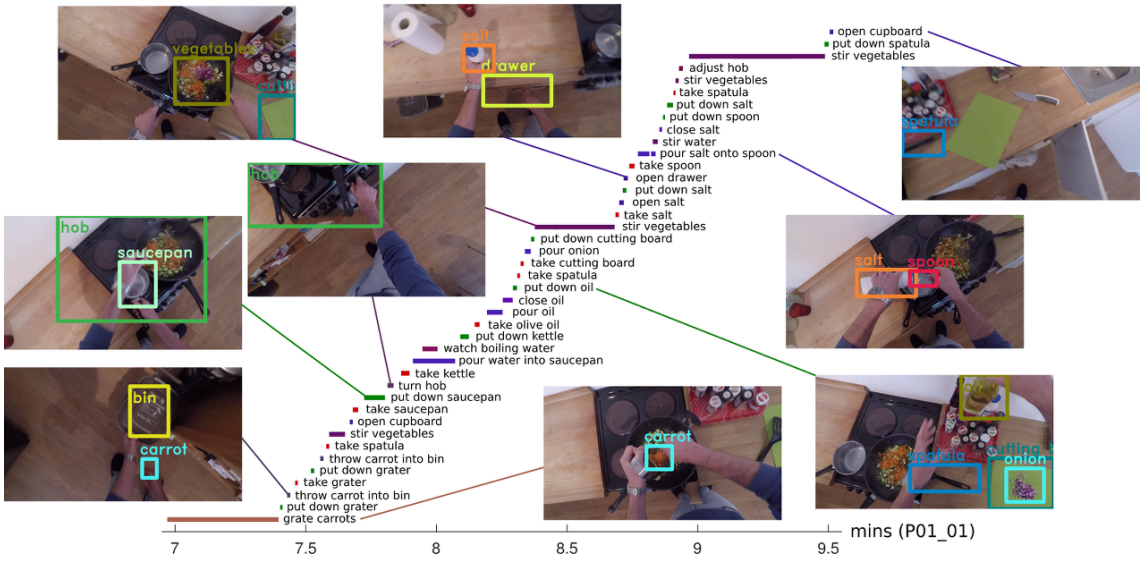


Figure 3.10 – Timeline of a video from Epic-Kitchens dataset (Damen et al., 2018).

The dataset was recorded by 32 persons in four cities of different countries. It counts 55 hours of videos: 39 594 action segments over 125 classes of average length 3.7 seconds. The videos were annotated using post recorded narration from the participants, YouTube automatic tool to match action and script and AMT services. The high number of classes, their similarity and their variation in length make this dataset utterly challenging.

ADLEgo

ADLEgo Dataset, introduced by Cartas et al. (2020), is based on egocentric activity recognition for health monitoring. The acquisition is done with a wearable camera on the chest. However the data acquired are not videos but images. Still, activity recognition is possible and authors annotated the acquired images with 35 different activity classes.

3.4 In the Wild Datasets

"In the wild" means that the videos are from different sources and can be recorded by professional or amateurs. It thus may contain camera motion, strong blur, oc-

[14] https://epic-kitchens.github.io/2018

clusions... everything that can make the action recognition task harder. However, videos can also contain much background information, which might be an exploitable source for the model trained to classify.

The UCF Datasets

The first UCF dataset was UCF-Sports[15] (Rodriguez et al., 2008; Soomro and Zamir, 2014). UCF letters comes from the name of the university in which the datasets have been developed: University of Central Florida.

UCF-Sports dataset contains various sequences from broadcast television channels across nine different sports: "diving", "golf swinging", "kicking", "lifting", "horse-back riding", "running", "skating", "swinging a baseball bat", and "pole vaulting". Pole vaulting is split in two classes: "Swing-Bench" and "Swing-Side" totaling ten classes. It first contained 200 sequences (reduced to 150 later) with an image resolution of 740×480 at 10 fps.

Later, Liu et al. (2009) introduce the UCF YouTube Action also called UCF11 dataset[16]. It consists of 11 classes from 1 160 videos from the online video platform YouTube[17].

Published in 2013 but work carried out in 2011, Reddy and Shah (2013) present the UCF50 dataset[18]. It is an extension of UCF11 with a total of 50 action classes. Soomro et al. (2012) extend this last version to make UCF101 dataset[19]. An overview of the UCF101 dataset is presented in Figure 3.11.

UCF101 includes a total number of 101 action classes which can be divided into five domains: "Human-Object Interaction", "Body-Motion Only", "Human-Human Interaction", "Playing Musical Instruments" and "Sports". Constructed from 2500 videos "in the wild", they extract a total of 13 320 clips in order to have at least 101 clips per class. The dataset is widely used by the scientific community and led to the THUMOS challenge[20] weld in 2013, 2014 and 2015. The dataset is cleaned and enriched with temporal annotations in 2015 in order to provide qualitative benchmark for different methods and be used also for spatio-temporal localization and temporal detection only.

Olympic

Olympic dataset[21] was introduced by Niebles et al. (2010). It is composed of Olympic sports from the online video platform YouTube. It contains 16 different sports with 50 sequences per class.

[15] https://www.crcv.ucf.edu/data/UCF_Sports_Action.php
[16] www.crcv.ucf.edu/data/UCF_YouTube_Action.php
[17] www.youtube.com
[18] www.crcv.ucf.edu/data/UCF50.php
[19] www.crcv.ucf.edu/data/UCF101.php
[20] www.thumos.info
[21] vision.stanford.edu/Datasets/OlympicSports/

Figure 3.11 – Overview of the UCF101 dataset (Soomro et al., 2012).

HMDB51

Kuehne et al. (2011) present the Human Motion DataBase (HMDB)[22]. The dataset comes from different sources of videos: movies, public databases such as the Prelinger archive [23] (collection of films relating to U.S. cultural history, evolution of the American landscape, everyday life and social history) and from other videos available on online platform e.g YouTube or Google videos. Samples are represented in Figure 3.12.

a. Diving **b.** Drinking **c.** Hand waving **d.** Sword fighting

Figure 3.12 – HMDB dataset samples (Kuehne et al., 2011).

There is one action per clip which have been annotated by students using 51 classes. Videos are required to have a minimum height of 60 pixels. All videos are re-scaled to a height of 240 with 30 fps for consistency. With at least 101 clips per action, the dataset contains 6 766 clips from 3 312 different videos. The fact that the clips are from so many different videos increase the intra-class dissimilarity and might make the action recognition task harder. The dataset can be split in five main categories: "General facial actions", "Facial actions with object manipulation", "General body movements", "Body movements with object interaction" and "Body movements for human interaction".

This dataset is extended later by Jhuang et al. (2013) to create the JHMDB dataset[24] which provides, in addition, human joint information.

Sports-1M

Sports-1M (Karpathy et al., 2014) is the first widely used dataset to reach the Million of videos (1 133 158). The teaser frame is visible in Figure 3.13 and shows the richness of their dataset.

It contains 487 classes of sports with top categories such as "Aquatic Sports", "Team Sports", "Winter Sports", "Ball Sports", "Combat Sports" and "Sports with Animals". The granularity of the classes is organized as a tree and becomes fine-grained at the leaves. There are for example 23 types of billiards. The annotations were created in a automatic way using video description and may contain mistakes. They also stress the fact that the action might not happen during the whole video

[22] serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/
[23] https://archive.org/details/prelinger
[24] http://jhmdb.is.tue.mpg.de/

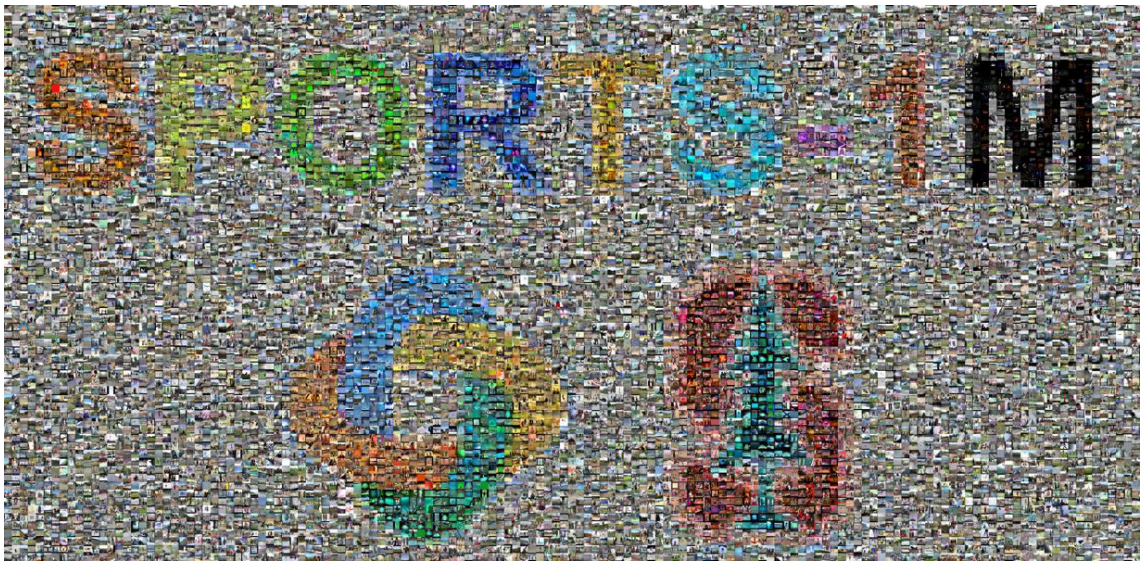


Figure 3.13 – Teaser frame of the Sports-1M dataset Karpathy et al. (2014).

(which can last several minutes), and might be temporally wrongly located, which makes the classification task even harder.

ActivityNet

Heilbron et al. (2015) introduce a large dataset based on activities: ActivityNet[25]. Their taxonomy, partially represented in Figure 3.14, has a stronger granularity than the previous datasets at their time.

Indeed, they offer different categories of action at different levels - 4 levels of granularity. This organization can be used to train cascade models for classification. The seven top categories are: "Household", "Caring and helping", "Personal care", "Work-related", "Eating and drinking", "Socializing and leisure" and "Sports and exercises".

They consider a total of 203 activity classes with an average of 193 sample videos per class (or 137 untrimmed videos with some videos containing more than one activity). It represents 19 994 clips totaling 849 hours of videos.

YouTube-8M

2 years after Sports-1M, YouTube-8M[26] is introduced by Abu-El-Haija et al. (2016). It holds 4 800 classes, a total of 8 264 650 videos meaning 1.9 billion frames and 1.53 Terabytes frame-level features. Videos can also have more than one annotation. In this dataset the classes are not actions but entities spread in 28 top categories such as "Health", "Science", "Games" and "Arts & Entertainment". The entities are

[25] http://activity-net.org/
[26] https://research.google.com/youtube8m/

Figure 3.14 – Sub-tree of the top level category “Household activities” Heilbron et al. (2015).

thus more context based, as for example: "Truck", "University", "News broadcasting" or "Trailer". The annotation system is based on Freebase Open Knowledge Graph (Bollacker et al., 2008).

The Kinetics Datasets

The kinetics datasets[27] started with Kinetics-400 introduced by Kay et al. (2017). Kinetics-400, Kinetics-600 (Carreira et al., 2018) and Kinetics-700 (Carreira et al., 2019) consider respectively 400, 600 and 700 action classes. They are all financed by DeepMind company, specialized in AI which, from 2014, belongs to Google.

The videos are collected from YouTube video platform, automatically annotated and candidates are refined using AMT. An overview of Kinetics-400 is presented in Figure 3.15.

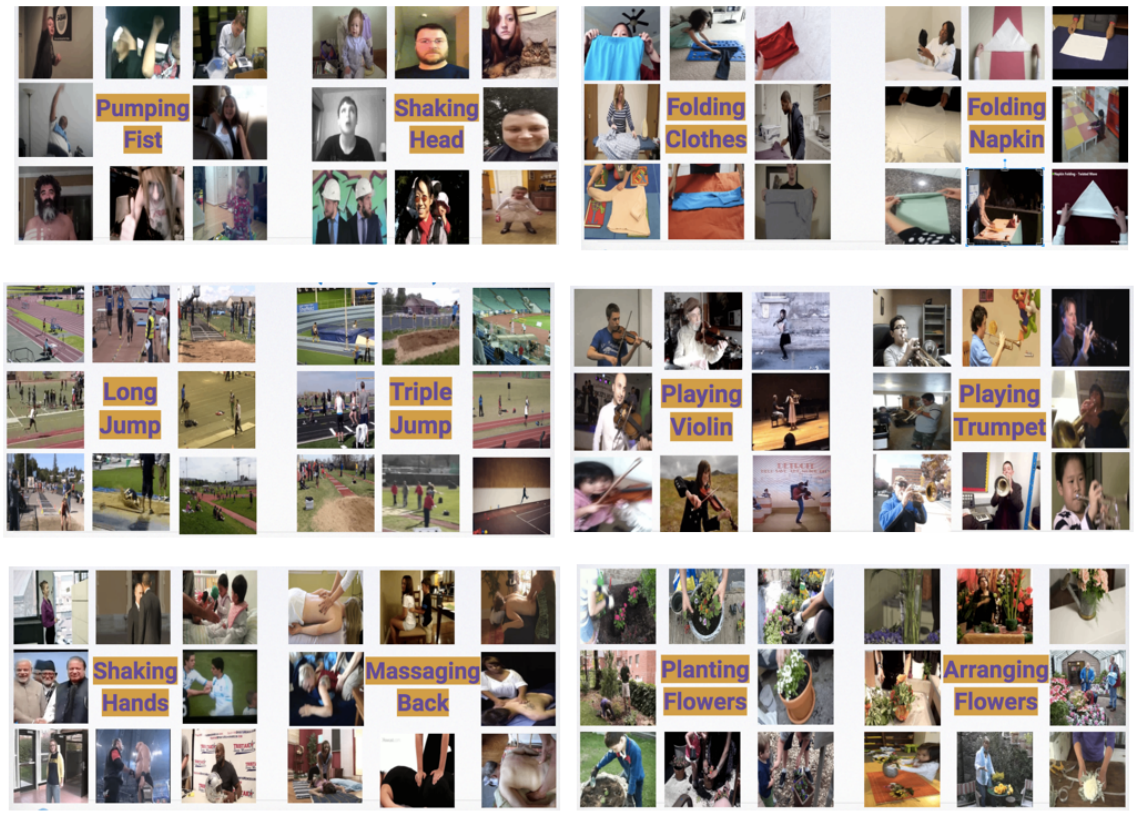


Figure 3.15 – Overview of the Kinetics dataset (Kay et al., 2017).

The difference between the versions of the datasets lies in:

- the number of classes: the number of classes has increased over time. New classes were added and pre-existing classes were refined. Some were merged.

[27] https://deepmind.com/research/open-source/kinetics

- the amount of videos: the number of videos started with 306 245 clips and more than doubled in the last version

- the splits between the different sets: training, validation and test sets have been modified over time. For example, samples belonging to the training set in the first version might belong to the test set in the last version. A new split for Kinetics-700 has recently been proposed by the same team, which increases their classification performances (Smaira et al., 2020).

AVA

Gu et al. (2018) introduced the Atomic Visual Actions (AVA) dataset[28] in order to perform joint localization and classification of actions. It contains 437 videos recovered from YouTube, 15 minutes are extracted from them and annotated every second. They use a vocabulary of 80 atomic actions. The difficulty in this dataset is the overlapping actions in time and their localization, as described in Figure 3.16.

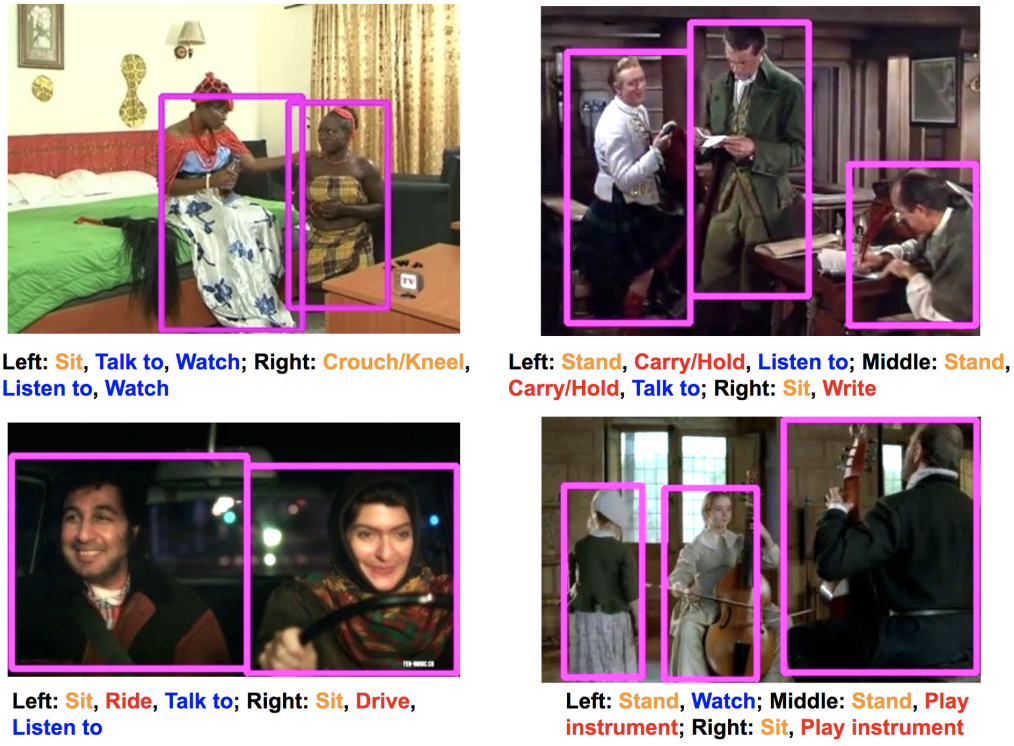


Figure 3.16 – Samples of the AVA dataset (Gu et al., 2018).

They offer a split of the dataset by extracting 900 video segments of three seconds from all the 15 minutes videos. By doing so, the 55 hours of video are split in 392 000 overlapping segments.

[28] https://research.google.com/ava/

Moments In Time

Moments In Time Dataset was first introduced in 2018 by (Monfort et al., 2020). It contains 1 Million videos classified using a taxonomy of 339 different classes. Those classes come from the 4 500 most commonly used verbs from the lexical resource VerbNet Schuler (2006) according to their frequency in the Corpus of Contemporary American English (COCA) Davies (2010). Those verbs are then clustered to end up with a selection of 339 verbs. The main difference with a verb compared to an human based action, is that it can be applied to anything. As it can be seen in Figure 3.17, the videos can have a very strong intra-class variation, i.e falling verb class can be applied to water, person, animal or object.

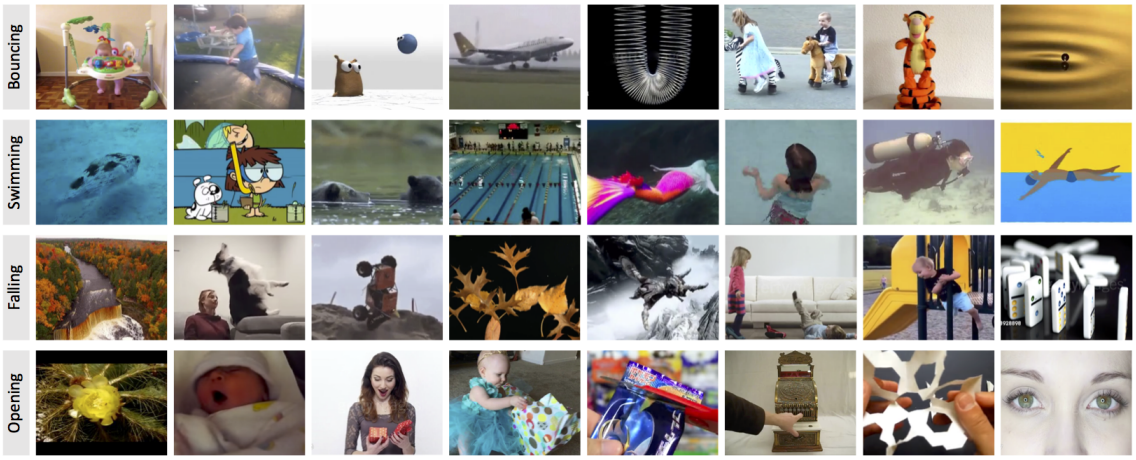


Figure 3.17 – Samples of Moments in Time dataset (Monfort et al., 2020).

The samples are three seconds long and extracted from videos available online. AMT were commissioned for the annotation process. They also extended the dataset to a Multi-Moments version where a video can contain several classes (Monfort et al., 2019).

SAR4

Fani et al. (2019) present the SAR4 dataset which focuses on action in football sport (or soccer). They track and label the players from available videos on YouTube. The actions performed by the tracked players are then annotated using a taxonomy of four classes: “dive”, “shoot”, “pass received” and “pass given”. The total number of sequences is 1 292 with actions lasting from five up to 59 frames.

HACS

Zhao et al. (2019) present the Human Action Clips and Segments (HACS) dataset[29]. The dataset is designed to refine the temporal localization of actions in video. The

[29] http://hacs.csail.mit.edu/

dataset is annotated accordingly to 200 action classes. From the 504 000 YouTube videos, $1.5M$ clips of two seconds are extracted. The authors provide different types of annotations, sparse and dense, according to the classification task covered. This dataset has been recently used in CVPR'20 International Challenge on Activity Recognition Workshop: HACS Temporal Action Localization Challenge.

AVA-Kintetics

Li et al. (2020) presented recently the AVA-Kinetics datasets[30]. The dataset is the merger of the two Kinetics-700 and AVA datasets. Kinetics videos were annotated using AVA protocol. The dataset thus contains over 230 000 video clips spatially and temporally annotated using the 80 AVA action classes.

4 The TTStroke-21 Dataset

The creation of the TT-Stroke21 dataset was initiated at the beginning of the CRISP project. The target application of this research project is fine-grained recognition of sport actions, in the context of the improvement of sport performance for amateurs or professional athletes. Our case study is table tennis, and our goal is the temporal segmentation and classification of strokes performed. The low inter-class variability makes the task more difficult for this content than for more general action databases such as UCF or Kinetics.

Twenty stroke classes and an additional rejection class have been established based on the rules of table tennis. This taxonomy has been discussed and designed with table tennis professionals. Videos recorded at the Faculty of Sports of the University of Bordeaux (STAPS) are considered in our work. The filmed athletes are students, and their teachers supervise the exercises performed during the recorded sessions. These recordings are done without markers, which allows the players to play in natural conditions. The objective of table tennis stroke recognition is to help the teachers to focus on some of these strokes to help the students in their practice.

One can mention that action recognition in table tennis videos is recently getting interest in the research community. Wang et al. (2020) try to visualize and characterize tactics in table tennis competitions using a Markov chain model for comparing the profile of different players. Similarly, Tsai (2018) presents a basketball tactic training framework based on motion capture devices for coaching basketball players. Other works only focus on the ball tracking and trajectory estimation (Lin et al., 2020; Calandre et al., 2021). Voeikov et al. (2020) propose an advanced real-time solution for scene segmentation, ball trajectory estimation and event detection but are not considering stroke classification. Ebner and Findling (2019) also focus on performances stroke recognition but in tennis. Their target application is also improvement of players performance. However they use sensors and only eight strokes

[30] https://deepmind.com/research/open-source/kinetics

Table 3.1 – Presentation of the different datasets in terms of number of classes, acquisition process, the amount of videos and the number of extracted clips.

Datasets	# classes	Acquisition	# videos	# clips
Coded-Faces 2000	44	Controlled	-	1 917
KTH 2004	6	Controlled	600	2 391
Weizmann 2007	9	Controlled	-	81
Coffee and Cigarettes 2007	2	Film	1	246
UCFSports 2008	10	Broadcast	-	150
UCF11 2009	11	In the wild	-	1 160
Hollywood 2009	12	Films	69	1 694
MSR 2010	3	Controlled	-	54
Olympic Dataset 2010	16	In the wild	-	800
UCF50 2011	50	In the wild	-	6 676
HMDB51 2011	51	In the wild	3 312	6 766
UCF101 2012	101	In the wild	2 500	13 320
Sports-1M 2014	487	In the wild	$1.13M$	-
ActivityNet 2015	203	In the wild	-	19 994
MERL Shopping 2016	5	Controlled	96	-
Charade 2016	157	Film at home	9 848	-
YouTube-8M 2016	4 800	In the wild	$8.27M$	-
Something-Something 2017	174	Egocentric	-	108 499
Kinetics-400 2017	400	In the wild		306 245
Diving48 2018	48	Broadcast	-	18 000
AVA 2018	80	In the wild	437	392 000
Epic-Kitchens 2018	125	Egocentric	-	39 594
Something-Something v2 2018	125	Egocentric	-	220 847
Moments In Time 2018	339	In the wild	-	$1M$
Kinetics-600 2018	600	In the wild	-	495 547
SAR4 2019	4	Broadcast	-	1 292
Toyota Smarthome 2019	31	Controlled	126	16 115
HACS 2019	200	In the wild	504 000	$1.5M$
Kinetics-700 2019	700	In the wild	-	650 317
FineGym 2020	354	Broadcast	-	32 697
AVA-Kinetics 2020	80	In the wild	-	230 000

types which limit the applications.

Table tennis strokes are most of the time visually similar. Action recognition in this case requires not only a tailored solution, but also a specific expertise to build the ground truth. This is the reason why annotations were carried out by professional athletes. They use a rather rich terminology that allows the fine-grained stroke definition. Moreover, the analysis of the annotations shows that, for the same video and the same stroke, professionals do not always agree. The same holds for defining temporal boundaries of a stroke, which may differ for each annotator. This variability cannot be considered as noise, but shows the ambiguity and complexity of the data and has to be taken into account. This new database is called TTStroke-21, TT standing for Table Tennis and 21 for the number of classes.

4.1 TTStroke-21 Acquisition

TTStroke-21 is composed of videos of table tennis games with 17 different players. This data set is continuously enriched with videos of different players at different frame rates, spatial resolutions and camera viewpoints. These sequences are recorded indoors without markers using artificial light and GoPro cameras. The cameras can be mounted on tripods on the grounds or on tables to have a higher point of view, or directly on the ceiling as presented in Figure 3.18.b.

The player is filmed in three situations:

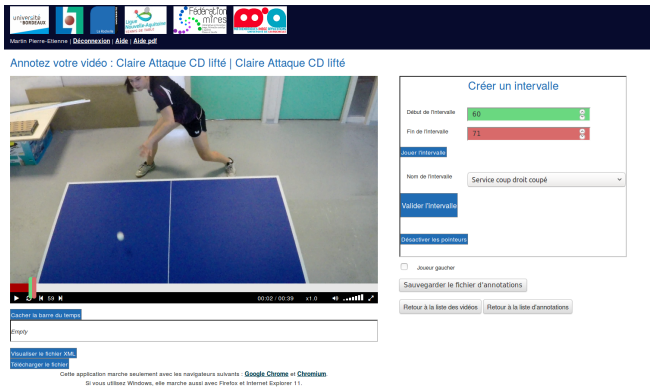
- performing repetition of the same stroke: such repetition are meant for the player to train on specific techniques. However those repetitions might fail once or several times in the video and the player might do another stroke than the one expected.
- simple exchanges between two players: those exchanges are meant to practise the different techniques. The players are not meant to mark points. The flow of the strokes is at normal pace.
- in match conditions: the players are meant to mark points. The game speed is much faster and strokes are shorter in time. The strokes are also harder to annotate in such context because of the speed and improvised strokes to answer to a difficult pass of the ball.

4.2 TTStroke-21 Annotation

The annotation process was designed as a crowdsourcing method. The annotation sessions are supervised by professional table tennis players and teachers. A user-friendly web platform has been developed by our team for this purpose (Figure 3.18.b), where the annotator spots and labels strokes in videos: starting frame, end frame and the stroke class. The annotator also indicates if the player is right-handed or left-handed. The taxonomy is built upon a shake-hand grip of the racket as pictured in Figure 3.19.

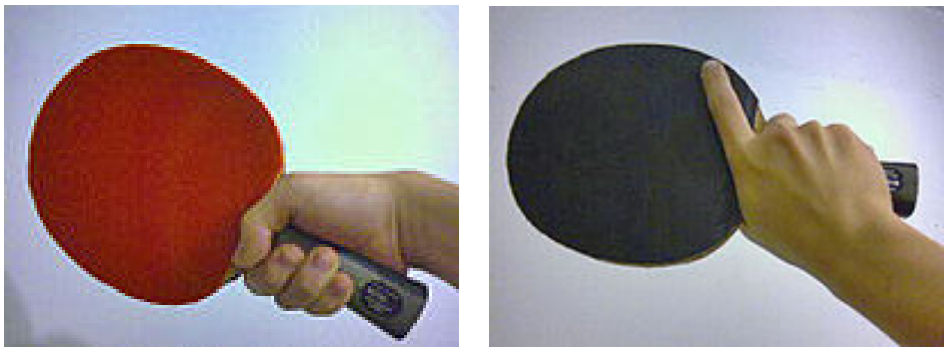
a. Dataset teaser image

b. Video acquisition with aerial view from the ceiling



c. Annotation platform

Figure 3.18 – Overview of the `TTStroke-21` dataset.



a. Forehand **b.** Bachhand

Figure 3.19 – Shake-hand grip of table tennis racket.

The taxonomy comprises 20 table tennis stroke classes, i.e.

- **8 services:** *Serve Forehand Backspin, Serve Forehand Loop, Serve Forehand Sidespin, Serve Forehand Topspin, Serve Backhand Backspin, Serve Backhand Loop, Serve Backhand Sidespin, Serve Backhand Topspin*;
- **6 offensive strokes:** *Offensive Forehand Hit, Offensive Forehand Loop, Offensive Forehand Flip, Offensive Backhand Hit, Offensive Backhand Loop, Offensive Backhand Flip;*
- **6 defensive strokes:** *Defensive Forehand Push, Defensive Forehand Block, Defensive Forehand Backspin, Defensive Backhand Push, Defensive Backhand Block, Defensive Backhand Backspin.*

All the strokes can, as well, be divided in two super-classes: **Forehand** and **Backhand**. This taxonomy was designed with professional table tennis teachers and should cover all possible strokes. This taxonomy could be refined to take into account more ball effects but would have been harder for professionals to annotate the video. Moreover, annotations are fulfilled by professional athletes, who are using quite a rich terminology. The linguistic analysis of annotations shows that for the same video and the same stroke, professionals do not employ the same degree of details in their annotations: an Offensive Forehand Hit can be similar to an Offensive Forehand Loop according to the effect given to the ball. The same problem occurs with temporal analysis: for instance, a service (first stroke when the player releases the ball) might be considered to start i) when the player is in position, ii) when the ball is released or iii) when the racket is moving. This cannot be considered as a noise, but it shows the ambiguity and complexity of real-life data.

In order to avoid annotation errors as much as possible, one recorded video was supposed to be annotated by at least 2 annotators. Unfortunately, this condition was hard to meet for all videos, and despite efforts for cleaning the data sets build from crowdsourced annotations, errors might still remain.

4.3 Crowdsourcing Filtering

We had a team of 15 annotators, professionals in the field of table tennis. In all crowdsourced applications, possible errors of the annotators should be taken into account. As the annotators were not familiar with the annotation platform at the beginning of the annotation sessions, there were some mislabelled portions of the videos. These mislabels have been filtered out automatically by not considering: annotations starting at first frame (default parameter), annotations ending after the end of the video and annotations out of the time range which was set between 0.6 and 2.3 seconds. The length of the time range was set up accordingly to the domain knowledge of professional table tennis players of the Faculty of Sports. This allowed the extraction of strokes ranging from a fast hit (sometimes less than one second) to a long serve (which can take more than two seconds).

Since a video can be annotated by several annotators, stroke detection according to the annotations was necessary. Our data set is player-centered, with only one player in each video. An overlap between each annotation of 25% of the annotated stroke duration is allowed. Indeed, during matches with fast exchanges, the boundaries between strokes are hard to determine and annotators would sometimes overlap the annotations between two successive strokes. Above this percentage, the annotations are considered to be part of the same stroke and are temporally fused.

Another filter is applied by checking if labels of the same stroke are consistent. If not, this portion of video is not considered in our classification task. This filtering, based on multiple annotations for the same recorded video, can still leave some labeling errors since multiple labeling of the same clip by different annotators was not always easy to meet in practice.

4.4 Negative Samples Extraction

Negative samples are created from videos with more than ten detected strokes. This was decided after noticing how some videos were poorly annotated. Indeed, videos are not fully annotated most of the time for different reasons. We suspect that it comes from the annotators' fatigue: they missed some strokes; or stopped the annotation process and did not finish the video annotation later. This would lead to include actual strokes as negative samples.

The negative samples are video sub-sequences between each detected stroke. We allow the overlap with the previous and the subsequent stroke of 10% of our target time window length: 0.83 seconds, which allows to capture short strokes without considering another one. This represents 100 frames at 120 fps. However, this approach was still selecting wrong negative samples because of videos that were only partially annotated. This has been manually cleared to avoid the incorporation of strokes in negative samples. Samples from `TTStroke-21` are represented in Figure 3.20.

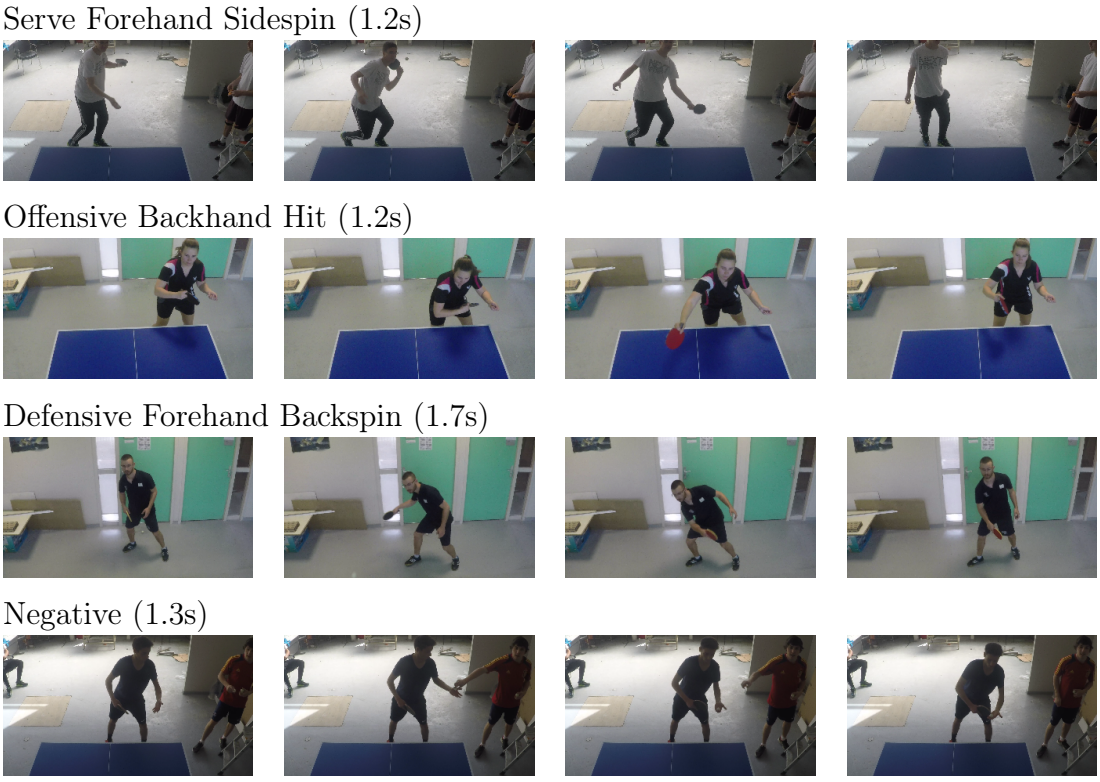


Figure 3.20 – Samples of TTStroke-21 after annotation filtering. In respective order the first frame, frames at 1/3 and 2/3 of the sample duration, and the last frame of the sample.

4.5 Data Distribution

`TTStroke-21` contains in total 241 videos recorded at different frame rates. The number of the videos, annotations and strokes extracted over the frame rates are reported in Table 3.2.

Table 3.2 – `TTStroke-21` database description.

fps	**# videos**	**# frames**	**# minutes**	**# annotations**	**# strokes**
25	46	60 087	40	437	145
30	12	65 523	38	302	280
120	183	2 091 615	291	3470	3003

We report also the distribution of the strokes across the dedicated taxonomy according to the frame rates of the videos in Table 3.3.

Table 3.3 – Stroke distribution Taxonomy

Table tennis strokes	**25 fps**	**30 fps**	**120 fps**	**All**
Serve Forehand Backspin	22	3	145	170
Serve Forehand Loop	3	4	100	107
Serve Forehand Sidespin	0	44	123	167
Serve Forehand Topspin	6	25	257	288
Serve Backhand Backspin	36	3	144	183
Serve Backhand Loop	1	2	63	66
Serve Backhand Sidespin	3	9	91	103
Serve Backhand Topspin	19	2	174	195
Off. Forehand Hit	22	12	231	265
Off. Forehand Flip	0	9	62	71
Off. Forehand Loop	3	18	255	276
Off. Backhand Hit	17	23	326	366
Off. Backhand Flip	12	18	40	70
Off. Backhand Loop	0	10	85	95
Def. Forehand Backspin	1	15	117	133
Def. Forehand Block	0	6	123	129
Def. Forehand Push	0	18	135	153
Def. Backhand Backspin	0	29	179	208
Def. Backhand Block	0	14	255	269
Def. Backhand Push	0	16	98	114
All strokes	145	280	3003	3428

Focus was on recordings using a frame rate of 120 fps in order to better visualize the strokes performed.

4.6 Data for Evaluation

Since the dataset was in constant evolution during the CRISP project, 129 videos at 120 fps have been considered using the aerial view from the ceiling. It represents 94 minutes of table tennis games, totalling 675 000 video frames and 1 387 annotations. After filtering, 1 074 annotations were retained. The peak statistics of stroke duration are $min = 0.64$ second, $max = 2.27$ seconds and the average duration is 1.46 seconds with standard deviation of 0.36. Accordingly, a total of 1 048 strokes were extracted with a *min* duration of 0.83 second, a *max* duration of 2.31 seconds and an average duration of 1.47 seconds with standard deviation of 0.36. Some annotations were merged making the statistical duration a bit longer. After these steps, 681 negative (non-stroke) samples were extracted. They have a mean duration of 2.34 seconds and standard deviation of 2.66 seconds. This high standard deviation comes from the non game activity of long period between strokes, which can be due to a ball lost or talks of players between games. However, as described in next chapter, not all negative samples are considered to avoid biases in the training and evaluation processes.

5 Conclusion

This chapter presented the different datasets for action recognition in a large sense used by the scientific community. From a chronological point of view, one can notice the evolution of the datasets from year 2000 up to now. The number of classes have increased, the complexity and acquisition methods have varied and the number of data has exploded. It can become complicated to test methods on a dataset containing millions of videos. Furthermore, since most of the actual datasets are from online platforms such as YouTube, most of them provide only links to the videos and their annotations. It is for example the case with the FineGym dataset presented earlier. However some videos might no longer be available because of their owner who deleted them. The disk usage is problematic as well . The frame extraction and flow computation from a large number of videos require servers with high storage capacity and an adequate number of Central Processing Units (CPUs). Finally, training a DNN model on such datasets with a correct batch size necessitates GPUs of large size.

Also, with the improvement of the classification methods, the tasks to perform on action classification datasets have evolved. Most of the datasets provide now joint segmentation (spatial and/or temporal) and classification task from videos. Something-something also offers to classify the action and the object(s) of interaction. It is interesting to see how some datasets focus on classifying actions with very low intra-class similarity, such as Something-Something or Moments In Time, while others focus on classifying actions with high inter-class similarity. It is often the case for datasets focusing on fine-grained classification, which offer many user oriented applications, as `TTStroke-21`.

`TTStroke-21` is used in the next chapters for the fine-grained action recognition task. In order to validate our results, we prefer to use I3D methods on our dataset rather than applying our method on other datasets. An attempt was actually made to run tests on the FineGym dataset but after starting downloading the videos, some were missing and the large size required in order to have the full dataset exceeded our capacities. We thus invited the scientific community to try their method on `TTStroke-21`. However, for privacy reasons concerning the players, this dataset cannot be publicly available. We are still working on it by blurring the players face, but the process is not perfect. A portion of `TTStroke-21` dataset is nevertheless available through the specific task Sports Video of MediaEval[31] after agreeing to particular conditions to respect the General Data Protection Regulation (GDPR). The MediaEval task is presented in appendix B.

[31] http://www.multimediaeval.org/mediaeval2020/

Part II

3D CNNs Architectures with Spatio-Temporal Convolutions for Actions Recognition in Videos

Abstract

Part II focuses on the methods we have developed in order to perform fine-grained action classification. The methods were motivated by the state-of-the-art methods at the beginning of the thesis which were the I3D models. The I3D models are applied to `TTStroke-21` and compared with the implemented methods. The models use the same modalities for consistency. The different modalities RGB and optical flow are investigated through two different chapters. Different optical flow normalization are tested and the obtained results stress its importance. The fusion of the two modalities through a Twin architecture is the object of the third chapter. Last chapter of Part II is dedicated to the features analysis of the best classification model.

Keywords

Action classification, Deep Learning, Optical Flow, Spatio-temporal convolution, Data Normalization, Feature understanding

Summary

Chapter 4

RGB Spatio-Temporal Convolutional Neural Network for Action Recognition

1 Introduction

The first deep learning breakthrough in image classification with AlexNet Krizhevsky et al. (2012) has led to many improvements such as GoogLeNet Szegedy et al. (2015), VGG-Net Simonyan and Zisserman (2015) and ResNet He et al. (2016) using RGB images. The next step was to extend these methods to the spatio-temporal domain for video classification. The main challenge in this task is to adapt existing works by taking into account temporal features. However, a direct extension of these methods to 2D+T presents some difficulties. The required memory space for training these models is indeed far greater, necessitating a reduction of the batch size for training neural networks. This leads to a greater computational time, especially if models are trained from scratch. Therefore, the temporal dimension must be taken into account carefully.

The state of the art method in action recognition from videos at the time when this research started was the Two-Stream I3D method (Carreira and Zisserman, 2017), which reaches 98% and 93.5% of accuracy on UCF-101 dataset, respectively with and without pretraining on the miniKinetics dataset (Kay et al., 2017). They follow the architecture of the two stream networks (Simonyan and Zisserman, 2014) but modify some of the convolutional layers with Inception modules along with transfer learning. They proceed by classifying temporal sliding windows, which is a common approach for action classification (Stoian et al., 2016). In their work, the temporal window size is 64 frames which may not be long enough to classify long-term actions. To overcome this limitation, Varol et al. (2018) use LTC considering as input video clips of 100 frames which improves the recognition of long-lasting actions. Temporal windows of 100 frames are used, at the expense of a less effective recognition of short term actions. As pointed out in Varol et al. (2018), this might be due to the repetition of the last frame to fill the required time window length.

Our proposed model was inspired by the method Varol et al. (2018), as we also use a temporal window of $T = 100$ frames, but with a frame rate of 120 fps against

25 fps in UCF-101 dataset Soomro et al. (2012). The choice of this window length is suitable, since actions in table tennis are fast executed and, by doing so, temporal aliasing should be avoided.

Note that video-based monitoring of athletes' performance is quite different from measuring fine movement of sportsmen or sportswomen. For example, Ahmadi et al. (2015) and Noiumkar and Tirakoat (2013) use body-worn inertial sensors. However, the use of invasive tools for monitoring might influence the performances of athletes. We recall that our goal is to develop a monitoring system based on vision only. This is why in this chapter we present our work for stroke recognition on `TTStroke-21` dataset using RGB data only. These results were partially published in our following publications: Martin et al. (2018, 2019c, 2021b, 2020c). Our RGB Spatio Temporal Convolutional Neural Network (RGB-STCNN) model reaches an accuracy of 89.8% against 84.5% for the I3D-RGB model.

2 Proposed Method

To be able to classify highly similar actions, table tennis strokes in our case, a 3D convolutional neural network model has been used to incorporate temporal features along with spatial ones. In this chapter, the stroke is predicted from RGB video frames only. The model is depicted in Figure 4.1 and is called RGB-STCNN.

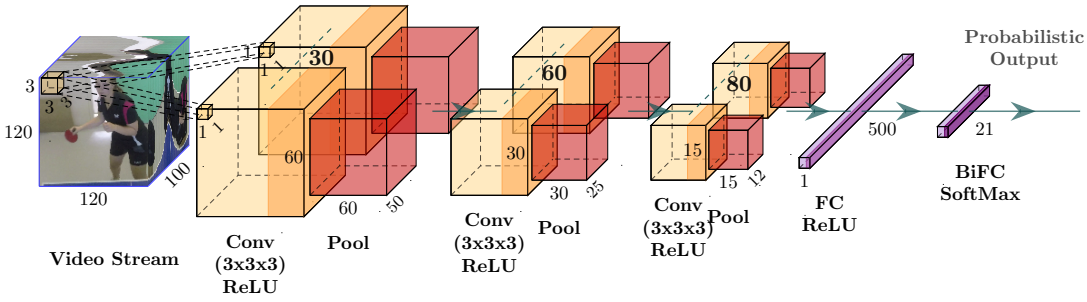


Figure 4.1 – RGB Spatio-Temporal Convolutional Neural Network - RGB-STCNN - architecture.

We address two problems: i) classification of actions, ii) detection by classification. The *classification problem* (i) consists in assigning a label to a temporal segment corresponding to a stroke *with known temporal borders* in a given video recording. The *detection by classification problem* (ii) consists in labelling strokes in the given video recording *without knowing their temporal borders*. In this case, simultaneous partitioning of the recorded video into strokes is fulfilled. In both tasks, temporal windows of several frames have to be classified. In one case, the classification is done inside temporal borders. In the other case, a sliding temporal window is classified at each given time. In both cases, a deep convolutional neural

network classifier has been designed and its architecture is described in the following sections.

2.1 Architecture of the RGB Spatio-Temporal Convolutional Neural Network

Our proposed network architecture RGB-STCNN is constituted of three 3D convolutional layers with 30, 60, 80 filter response maps, followed by two fully connected layers of size 500 and 21 respectively. All 3D convolutional layers use $(3 \times 3 \times 3)$ space-time filters with stride and padding of one in all directions, "ReLU" activation function and are followed by 3D Max Pooling layers using kernels of size $(2 \times 2 \times 2)$ and floor function. A Softmax layer is finally added at the end of our network to obtain a classification score vector of size 21 which corresponds to the number of considered classes. All layers are depicted in Figure 4.1.

2.2 Input Data

The RGB-STCNN takes as input RGB images $(W \times H \times T)$. The extracted frames from the video of size (1920×1080), are resized to (320×180) before feeding them to the network. Optical flow is then computed from the resized frames using "BeyondPixel" method (Liu, 2009), based on iterative re-weighted least square solver. The flow is then filtered and used here only for computing the Region-of-Interest. The detailed process for the flow computation is described in Chapter 5 where different motion estimators are tested. Before feeding the network and after ROI extraction and possible data augmentation, RGB channels are normalized by their theoretical maximum value (i.e. 255 for 8-bit channel coding) to map them into [0,1] interval.

Region-of-Interest Extraction

The ROI center $\mathbf{X_{roi}} = (x_{roi}, y_{roi})$ is estimated from the maximum of the optical flow $\mathbf{V}$ norm and the center of gravity of all pixels with non-null optical flow norm as follows:

$$
\begin{aligned}
&\mathbf{X_{max}} = (x_{max}, y_{max}) = \underset{x,y}{argmax}(||\mathbf{V}||_1) \\
&\mathbf{X_g} = (x_g, y_g) = \frac{1}{\sum\limits_{\mathbf{X}\in\Omega} \delta(\mathbf{X})} \sum\limits_{\mathbf{X}\in\Omega} \mathbf{X}\delta(\mathbf{X}) \\
&\text{with} \quad \delta(\mathbf{X}) = \begin{cases} 1 & \text{if } ||\mathbf{V}(\mathbf{X})||_1 \neq 0 \\ 0 & \text{otherwise} \end{cases} \\
&x_{roi} = \alpha \ f_{\omega_x}(x_{max},\ W) + (1-\alpha) \ f_{\omega_x}(x_g,\ W) \\
&y_{roi} = \alpha \ f_{\omega_y}(y_{max},\ H) + (1-\alpha) \ f_{\omega_y}(x_g,\ H)
\end{aligned}
\tag{4.1}
$$

with parameters $\alpha = 0.6$, set empirically, $\mathbf{\Omega} = (\omega_x, \omega_y) = (320, 180)$ the size of video frames. Function $f_\omega(u, S) = max(min(u, \omega - \frac{S}{2}), \frac{S}{2})$ allows to have data inputted to our network within the boundaries of the region of interest. To avoid jittering within our cuboids of size $(W \times H \times T)$, we then applied a Gaussian filter using a kernel of size k_{size} with scale parameter $\sigma_{blur} = 0.3 * ((k_{size} - 1) * 0.5 - 1) + 0.8$ along the temporal dimension to average the center position. In our experiments the optimal kernel size was established to $\frac{1}{3}$ second which represents k_{size}= 41 frames at 120 fps.

2.3 Data Augmentation

For each stroke, we extract one video sample of size $(W \times H \times T)$. Without data augmentation, the T frames from the video stream are centrally extracted in the temporal and spatial dimensions according respectively to the duration of the stroke Δt and our ROI extraction.

For spatial augmentation, we apply random rotation in the range $\pm 10°$, a random translation in x and y direction respectively in range $\pm 0.1 * W$ and $\pm 0.1 * H$, and a random homothety in the range 1 ± 0.1. Transformations are applied and centered on the ROI.

To perform temporal augmentation, a clip of T successive frames is extracted, whose center is drawn from a Gaussian distribution centered on the stroke. The Gaussian distribution uses a standard deviation $\sigma = \frac{\Delta t - T}{L}$, with $L = 6$, which represents more than 99% of chance to be in the temporal boundaries of the stroke. This process is presented in Figure 4.2. However, if the frames are not in the temporal boundaries, another random draw is done until the condition is satisfied.

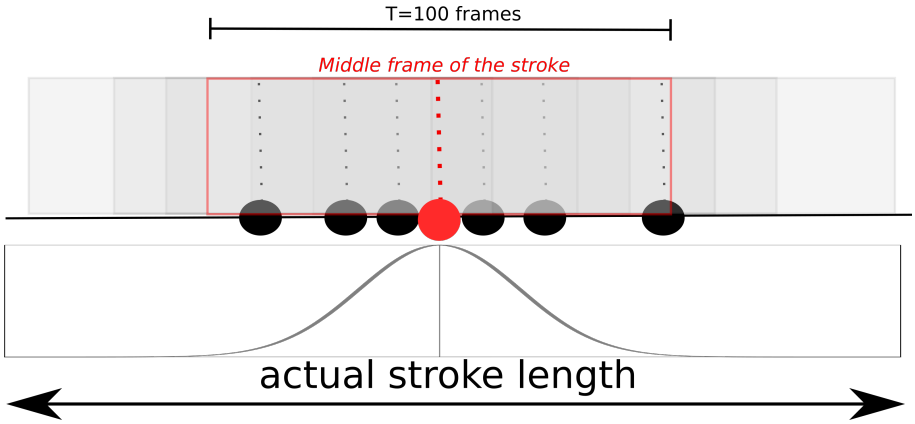


Figure 4.2 – Representation of 7 draws of the same stroke using temporal augmentation.

2.4 Training Step

Estimation of network parameters is fulfilled using Stochastic Gradient Descent with Nesterov Momentum Sutskever et al. (2013). It is computed according to equation 4.2:

$$\begin{aligned} v_{t+1} &= \mu v_t - lr \frac{\delta \mathcal{L}}{\delta \theta_t} \\ \theta_{t+1} &= \theta_t - v_{t+1} \end{aligned} \tag{4.2}$$

with v_0 initialized at zero, θ being parameters to be estimated, μ being the momentum coefficient, lr the learning rate, and $\mathcal{L}$ the objective function commonly called loss.

We use a momentum coefficient value of 0.5 and decrease it to 0.1 and 0.05 at epoch 1000 and 1500 respectively, as the momentum methods are known to oscillate at the beginning of the iterative process. We use a weight decay of 0.005. The maximum number of epochs is set to 2000. Cross-entropy loss is used as objective function. The batch size is relatively low for memory matter and is set to ten. The number of negative samples is chosen twice bigger than the mean of the number of strokes per class. The dataset is split into training, validation and testing sets with the respective proportions: 70%, 20% and 10% as represented in Table 4.1, with Train set denoted as "Tr" and Validation set denoted as "Val".

We use data augmentation on our training set for all models and evaluate them at each epoch with the accuracy on the validation dataset without augmentation. Models with the best accuracy are saved for the next evaluations on the test set.

2.5 Evaluation Methods

Classification Task:

To compare the performances of our model in the classification task, we use the **RGB-I3D** CNN introduced by Carreira and Zisserman (2017), which uses `inception_v1` architecture Szegedy et al. (2015), as our baseline and apply it to our dataset following their instructions for training. Their model needs to be trained with 115 000 iterations which represents 851 epochs in our case. In addition, the learning rate is decreased, changing from 0.1 to 0.01 and 0.001 respectively at iterations 97 000 and 108 000. Also, since our input data are twice smaller than theirs, the first max pooling layer of the RGB-I3D model has been discarded.

For this task, the goal is to recognize the class of an already localized stroke. To evaluate our models on the test set, four different methods have been used. The first one, simply referred as "Test", used also for the validation set, consists in classifying strokes by considering only T *frames temporally centered on each sample*. This method does not take into account the whole video segment duration and is based on the hypothesis that the main features are centered in time. The three other methods consider *all frames of the given sample*. For the those methods, we

Table 4.1 – Datasets distribution over the different splits and strokes duration.

	# **Samples**				# **Frames**		
Table tennis strokes	**Tr**	**Val**	**Test**	**Sum**	**Min**	**Max**	**Mean***
Serve Forehand Backspin	58	17	8	83	125	269	182 ± 35
Serve Forehand Loop	56	16	8	80	100	273	171 ± 51
Serve Forehand Sidespin	57	16	9	82	101	273	192 ± 39
Serve Forehand Topspin	67	19	9	95	100	273	184 ± 52
Serve Backhand Backspin	56	16	8	80	133	261	188 ± 31
Serve Backhand Loop	43	12	6	61	100	265	186 ± 42
Serve Backhand Sidespin	60	17	9	86	129	269	193 ± 33
Serve Backhand Topspin	57	16	8	81	100	273	175 ± 48
Off. Forehand Flip	31	9	5	45	113	269	186 ± 44
Off. Forehand Hit	45	13	6	64	100	233	158 ± 34
Off. Forehand Loop	23	7	3	33	101	277	177 ± 43
Off. Backhand Flip	25	7	3	35	100	265	195 ± 49
Off. Backhand Hit	28	8	4	40	100	173	134 ± 21
Off. Backhand Loop	21	6	3	30	100	229	155 ± 32
Def. Forehand Backspin	29	8	4	41	129	229	177 ± 25
Def. Forehand Block	8	2	2	12	100	137	115 ± 14
Def. Forehand Push	23	7	3	33	105	177	143 ± 19
Def. Backhand Backspin	22	6	3	31	121	233	189 ± 25
Def. Backhand Block	19	5	3	27	100	261	131 ± 37
Def. Backhand Push	6	2	1	9	121	229	155 ± 31
Non strokes samples	74	21	11	106	100	1255	246 ± 154
Total	808	230	116	1154	100	1255	182 ± 65

* in the form: mean value ± standard deviation

classify a stroke using a temporal sliding window with a time step of $\delta t = 0.1T$ frames. Class scores for each window are then obtained. Our first method then uses majority vote from the window decision to classify the whole segment and is referred as "TVote". Our second method uses the average score of the class scores among all the windows scores and is referred as "TAvg". Finally the last method weights the class scores using a temporal Gaussian and is referred as "TGauss". This method uses a kernel size "$ksize$" of the number of windows with scale parameter $\sigma = 0.3 * ((ksize - 1) * 0.5 - 1) + 0.8$ which is the default parameter of the OpenCv function used[1]. Performances for each method are shown in Table 4.2.

Detection by Classification:

The joint detection and classification in videos is done through the classification of video segments using a sliding window of size T with step one. This process is long and is performed only on our models. Hence we obtain a vector of probability scores P of size $T_{video} - T$ with T_{video} being the length of the video. To avoid border effects when classifying the whole video, we extrapolate P by simple copy of the first and last probability score respectively at the beginning and at the end of our probability vector. Different decisions have been experimented to integrate classification results along the time for smoothing the classification decision. The decision without temporal smoothing is denoted as "Gross". The majority vote and max average decision - which average the probabilities over the classes - use a window decision of size $1.5T$ and are denoted as "Vote" and "Average". Another decision is experimented which weights the classification probabilities over time using a Gaussian kernel of size $2T + 1$ with scale parameter $\sigma = 0.5T$ allowing greater consideration of close window classification probabilities while smoothing the decision. This last decision rule is denoted as "Gaussian". Because the detection may not be exact in time according to the crowdsourced annotations, the prediction is considered correct at the boundaries of strokes if it is classified as negative stroke or as one of the stroke that are overlapping. This overlap of label is set to 20% of the stroke duration. The performances are shown in Table 4.3.

To evaluate the performances of our method for detection and classifications in videos, we compare our predictions with the ground truth built from the crowdsourced annotations of `TTStroke-21` dataset. Since the videos are limited in the diversity of strokes, experiments for this task have been conducted with the whole dataset which incorporates strokes and negative samples that were in the training, validation and test sets.

[1] https://opencv.org/

3 Experiments and Results

The deep learning models have been trained using PyTorch framework on Graphics Processing Unit (GPU) NVIDIA Tesla P100. The size of the input data have been set to $(W \times H \times T) = (120 \times 120 \times 100)$ which results after the three convolutions and flattening process in a feature vector of size 216 000. T has been chosen with respect to the rapidity of strokes and represents the minimum stroke duration: 0.83 second as described in Chapter 3, section 4. We also made experiments by setting $T = 64$ to keep the same temporal parameters used in Carreira and Zisserman (2017) for better comparison with the baseline. W and H have been set according to the distance of the players to the camera, and thus to their visual appearance size in the frames.

3.1 Pure Classification Task

Table 4.2 – Performance comparison between RGB-I3D (Carreira and Zisserman, 2017) and RGB-STCNN (Ours).

		Accuracies in %					
Models	**T**	**Train**	**Val**	**Test**	**TVote**	**TAvg**	**TGauss**
RGB-I3D	64	86	40	40.5	9.5	9.5	10.3
RGB-I3D	100	98.3	72.6	69.8	84.5	84.5	84.5
RGB-STCNN	64	68	64.8	66.1	62.7	61.9	65.3
Gray-STCNN	100	97.7	75.7	26.1	25.4	24.6	24.6
RGB-STCNN*	100	97.7	75.7	70.7	68.1	69.8	68.1
RGB-STCNN	100	96.7	88.7	**89.8**	67.6	74.6	70.3

* trained without data augmentation

It can be observed in Table 4.2 that our RGB-STCNN model outperforms the RGB-I3D model Carreira and Zisserman (2017), both trained from scratch on our dataset, using $T = 64$ and $T = 100$. The maximum accuracy obtained on `TTStroke-21` test set using RGB modalities is 89.8% with RGB-STCNN model using the "Test" evaluation method and 100 frames, against 84.5% with RGB-I3D model Carreira and Zisserman (2017). The RGB-I3D model scores drops to 70.7% when the same model is trained without data augmentation. The models using only 64 frames fail to obtain good classification scores and in Varol et al. (2018) it is proven that greater number of frames can improve classification scores for long and similar actions referring to UCF101 dataset. However, in `TTStroke-21` actions are similar and temporally short but are recorded at 120 fps, leading to similar challenges.

Figures 4.3 and 4.4 show the accuracy and loss curves of the RGB-I3D models respectively with $T = 64$ and $T = 100$. The gap between the training and validation

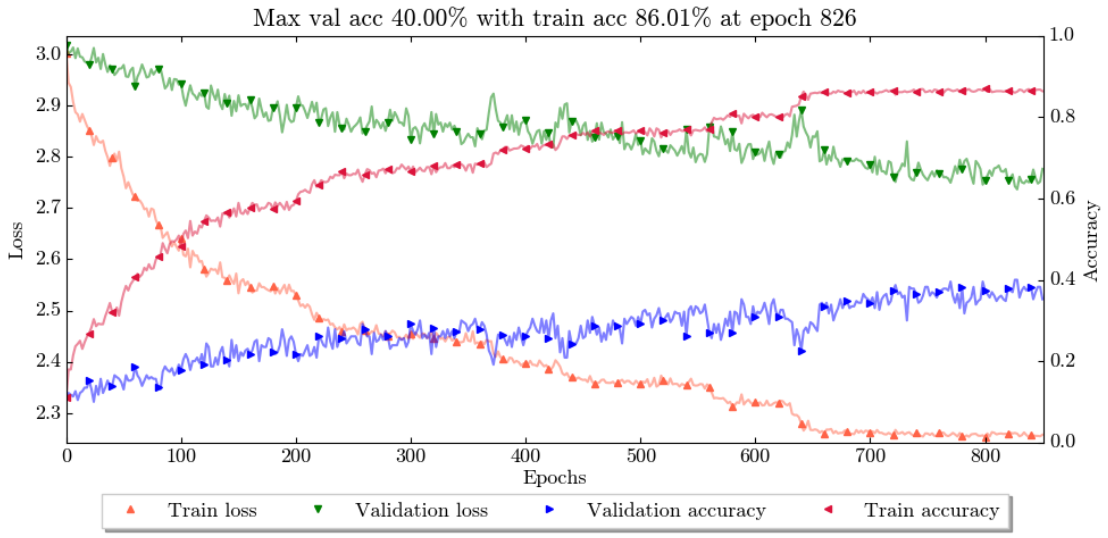


Figure 4.3 – Training process of the RGB-I3D model with $T = 64$.

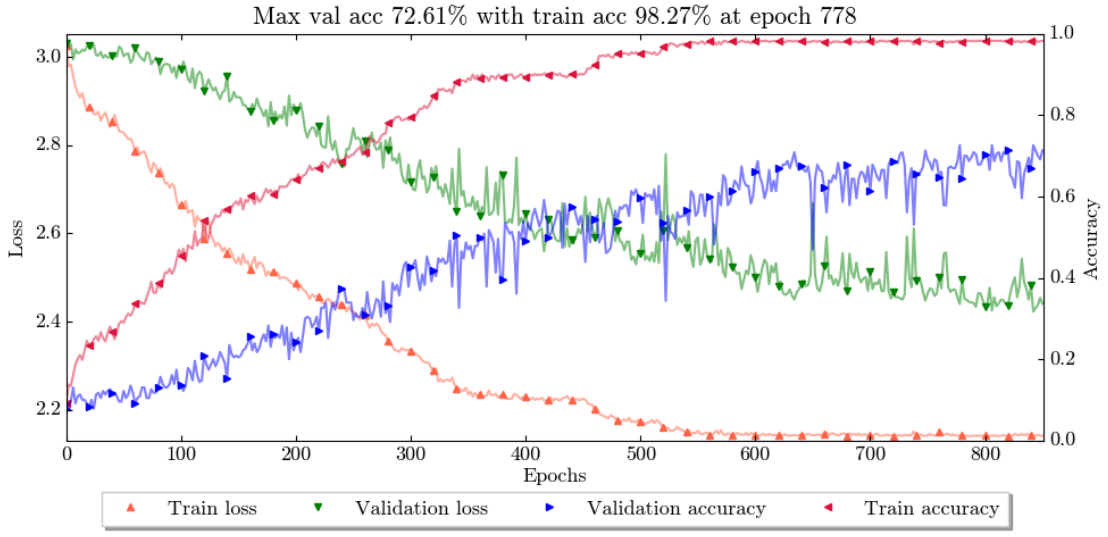


Figure 4.4 – Training process of the RGB-I3D model with $T = 100$.

accuracy for the I3D model, especially visible when $T = 64$, starts since the beginning of the training. A reason may be that this model is not meant to be trained on datasets with low number of samples, leading then to overfit the training set and fails to generalize the features extracted from the training set. The RGB-I3D model is much deeper than our model and has been evaluated on bigger datasets such as UCF101 and HMDB-51. Our dataset which focuses on low inter-classes classification, is also in general more challenging than the datasets used in their experiments and makes the task more difficult.

However the use of the temporal extension greatly increased the accuracy of the RGB-I3D model, trained with 100 frames, from 69.8% to 84.5% but failed for the model trained with $T = 64$ as reported in Table 4.2. The failure with the lower input size can be explained by the non-relevant features extracted at the beginning and the end of the stroke using a shorter temporal window. The temporal smoothing will thus have the tendency to classify stroke samples as Negative samples. This aspect is not observed on I3D-models certainly because of the greater receptive field of their model coming from their deepness, allowing to better combine temporal information.

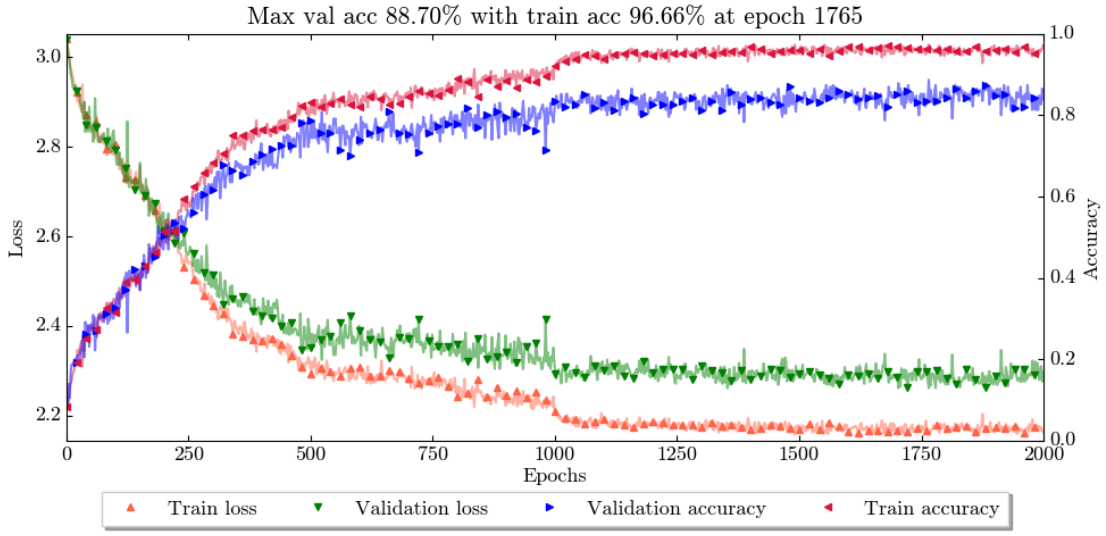


Figure 4.5 – Training process of the RGB-STCNN model with $T = 100$.

In contrary, the RGB-STCNN validation and training curves stay quite close to each other as presented in Figure 4.5. Surprisingly the prediction methods that took into account the temporal classification of the samples did not improve the scores and did even confuse the model (see Table 4.2). The discriminant features seem to be well centered on the stroke and other features, on the temporal borders, are not efficient.

Experiments using only gray scale images were also conducted to see how much color contributes to the classification. The model is denoted as "Gray-STCNN" in

the table. Hamel et al. (2016) present how much color is important to get the saliency of an image. In our case, it seems similar since one can observe much lower performances and difficulty for the model to converge. The parameters for training the model might not be adapted for this modality, but it can also be explained by the acquisition environment of our task. The table on which strokes are performed is blue, the racket is sided with two distinguishable colors: black and red, and the ball is either orange or white. By taking out the color information, those distinguishable colors will be similar to the background or the surroundings, and thus the discriminating features extraction harder to perform, leading to slower convergence and lower performance.

3.2 Analysis of Classification Results

To better understand the classification results of our RGB-STCNN model, we present in Figure 4.6 the confusion matrix obtained with our model trained using $T = 100$ and evaluated with the "Test" method. As it can be seen, some classes are entirely wrongly predicted. This is due to the lack of data in those classes in both training and test set. As indicated in Table 4.1, the "`Defensive Backhand Push`" class is poorly represented within the dataset. Moreover, since the annotations are crowdsourced, some strokes might be wrongly labeled, leading to mislearned classes. However the `TTStroke-21` dataset has been cleaned and reviewed many times to correct those mistakes.

We can also notice the limit of our RGB-STCNN model to predict the "Negative", non-stroke, classes. By visualizing the negative samples we actually notice that the player often misses the ball or is in position to perform the stroke but cannot finish it because the ball rebounds out of the table. The annotation is indeed correct since no actual stroke is performed but may lead the model to misunderstand the player intention. It does also explain the limits of the prediction methods to take into account the whole temporal information of the sample: features extracted at the border of the stroke are similar to the features of negative samples described earlier and lead the model to classify actual stroke as "Negative". This point is illustrated in Figure 4.7 showing the confusion matrix using the "TAvg" method for the same RGB-STCNN model.

3.3 Joint Stroke Detection and Classification Task

Joint stroke detection has been carried out with the RGB-STCNN model trained with and without data augmentation. When taking into account all labels (including negative ones), a maximum of 80.9% of accuracy is reached for detection and classification task using the average smoothing method and data augmentation (Table 4.3). We can notice how the temporal smoothing methods improves greatly the results. This underlines the instability of the scores obtained along the temporal dimension for this particular model.

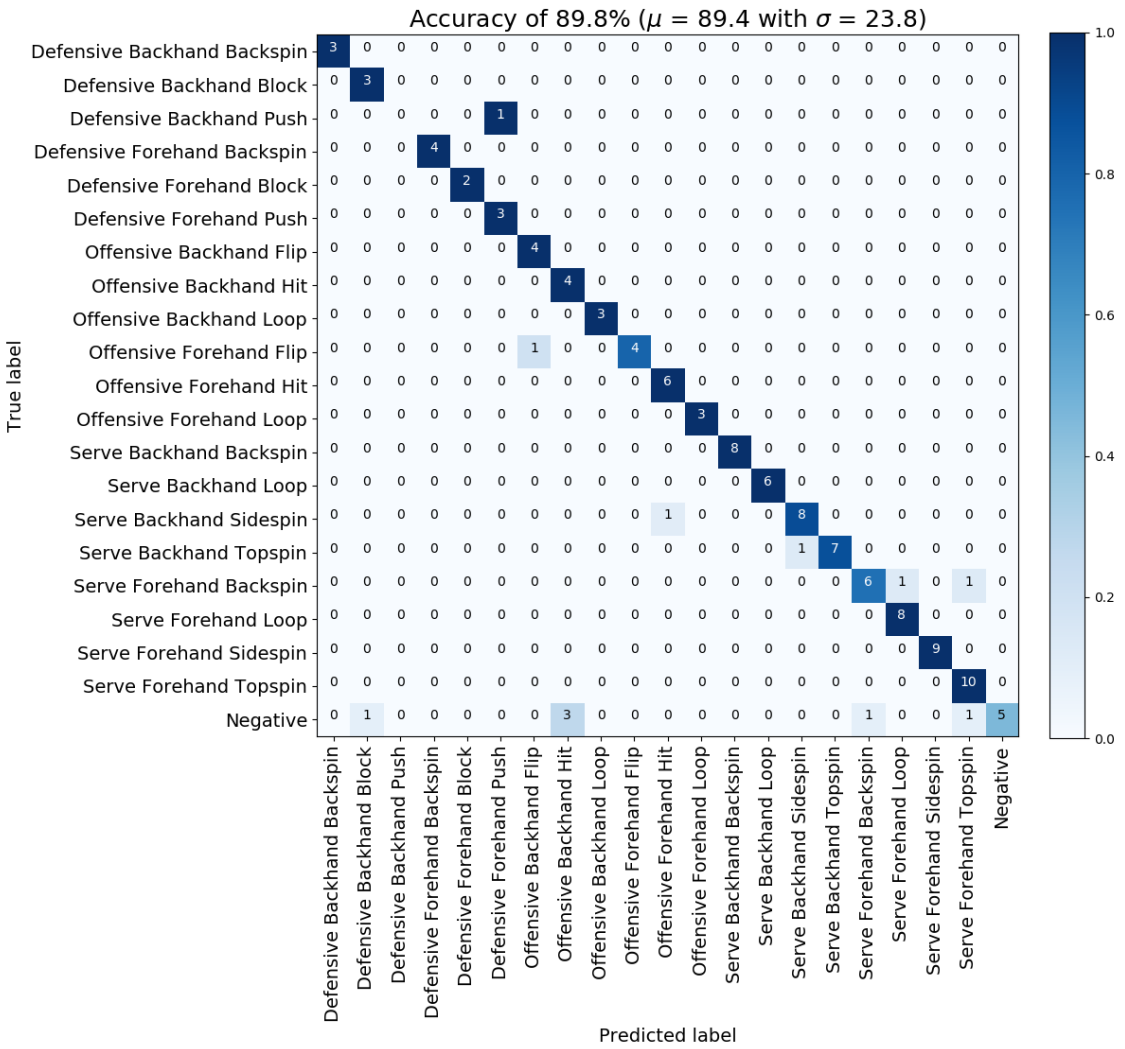


Figure 4.6 – Confusion Matrix of the RGB-STCNN model using "Test" method with $T = 100$.

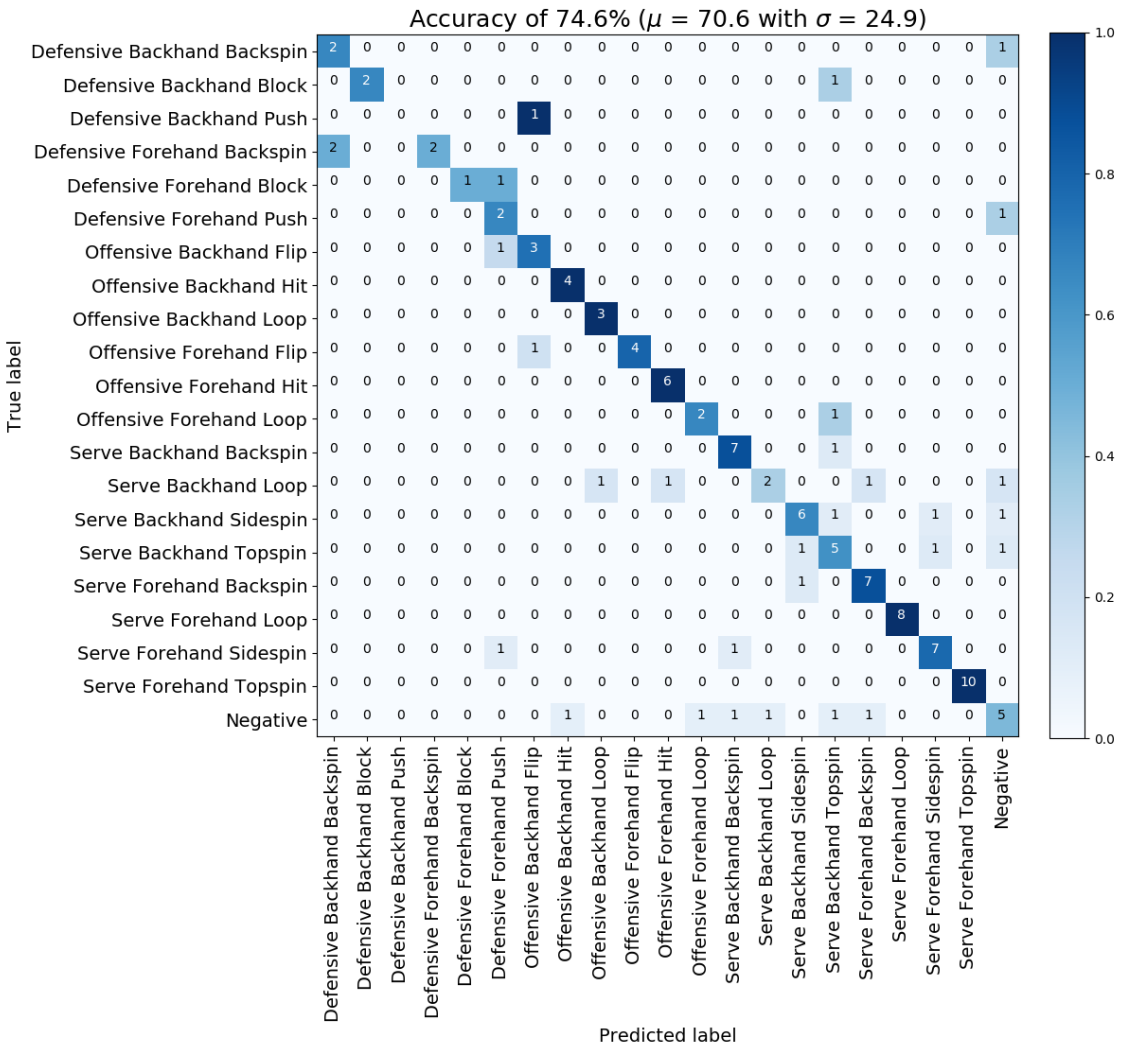


Figure 4.7 – Confusion Matrix of the RGB-STCNN model using "Avg" method with $T = 100$.

Table 4.3 – Performance of stroke detection and classification.

	Accuracies in %			
Models	**Gross**	**Vote**	**Average**	**Gaussian**
RGB-STCNN*	57.2	80	**80.9**	80.7
RGB-STCNN	57	80.1	80.8	80.2
without taking into account negative labels				
RGB-STCNN*	52.3	64.3	64.7	**67.9**
RGB-STCNN	41.5	44.8	46.2	49.1

* trained without data augmentation

A video can be mostly constituted of negative samples, which can make our performance evaluation biased. In Table 4.3 we also report results obtained without counting the "Negative" labels. Their overlaps with other strokes is also not considered which makes the evaluation method much more discriminant. Therefore we get much lower scores with a maximum accuracy of 67.9% using the Gaussian filtering method for the model trained without data augmentation. In this case, the data augmentation does not seem to be useful. Indeed, by taking out the negative portions and their overlaps with the strokes, it results in classification of well centered strokes. It thus makes sense that the RGB-STCNN trained without augmentation, which has been trained without temporal augmentation with RGB data centered on the strokes, obtains better results.

4 Conclusion

The goal of the presented of work is the challenging task of recognition of sport actions with weak inter-class variability in videos. To that aim, we have proposed an approach based on RGB data extracted from the video stream fed to a 3D Spatio-temporal Convolutional neural network - RGB-STCNN. The results are compared with a deeper neural network RGB-I3D (Carreira and Zisserman, 2017).

The superiority of our model on the classification task could be explained by the architecture of the I3D model which is too deep to be trained on our dataset, leading therefore to overfitting. Our model reaches the best accuracy with 89.8% on the test set and shows limits when using the negative labels. Despite its efficiency on the classification task, results are more questionable on the detection and classification task. Indeed the accuracy on this task reaches a maximum accuracy of 80.9% on the whole dataset which might look like a good performance. However when we do not consider non stroke parts of the videos, which are numerous, we reach only 67.9% of accuracy which enlightens the limits of our model. Better handling motion information using the optical flow is the core of the following chapter.

Chapter 5

Efficient Use of Optical Flow for Action Recognition

1 Introduction

Detecting and classifying human actions in videos is one of the current challenges in visual content analysis and mining. As drawn in the previous chapter, the use of RGB data only may not be enough to allow efficient detection and classification of actions in videos. In this chapter, we present a method for performing a fine-grained classification of sport actions with a Spatio-Temporal Convolutional Neural Network using optical flow data: `Flow-STCNN`. We compare different Optical flow methods and study their influence on the classification score. We also present different normalization methods of the optical flow that drastically impact results, boosting accuracy from 44% to 74.1% using the same number of iterations.

Motion information is obviously a crucial clue for recognizing actions in general and especially fine-grained ones (Shao et al., 2020). Table tennis strokes fall in this category and motivated our research using only motion information. Review of the state-of-the art shows that spatial features from RGB images are also needed to attain a reasonable accuracy for action classification, be it in sport (Varol et al., 2018) or in specific cultural content (Stoian et al., 2016). Static information mostly captures background characteristics of the action. Hence, it seems necessary to use both static data and temporal information. To efficiently fuse data of limited spatial extend with variable motion magnitude, an adequate normalization of motion has to be done. Accordingly, we compare different OF estimation methods and their normalization, and how they influence the training of our Flow Spatio Temporal Convolutional Neural Network (Flow-STCNN). Similarly to Chapter 4, we perform two tasks: classification only, and joint detection and classification. This chapter is related to some extent to our following publications: Martin et al. (2019d, 2021b, 2020c, 2019b).

The remainder of this chapter is organized as follows: Section 2 focuses on the comparison of different normalization of OF methods on the Sintel benchmark (Butler et al., 2012), and how they can be useful in our fine-grained classification context. Section 3 presents our Flow-STCNN model, the processing of the OF data before feeding the model and the classification methods. Results and performance

assessment for classification and joint classification and detection are presented in Section 4. Conclusion and prospects are drawn in Section 5.

2 Choice of the Optical Flow Estimator and Normalization

Optical flow estimation is of primarily importance in the analysis of spatio-temporal actions (Efros et al., 2003; Ng et al., 2018) and is often refereed as the dynamic or temporal stream in action recognition (Feichtenhofer et al., 2016; Simonyan and Zisserman, 2014). Indeed it encodes the temporal information of images based on the derivation of the intensity. Optical Flow is based on the constant brightness hypothesis of the images in video for each pixel over time (Horn and Schunck, 1981) and can be expressed such as:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{5.1}$$

with I the image, x and y the horizontal and vertical position of the pixel, t it position in time and dx, dy, and dt its displacement respectively in horizontal, vertical and temporal plan. High frame rate of videos allow us to consider that the displacements are small and by using Taylor series expansion to the first order we obtain:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\delta I}{\delta x}dx + \frac{\delta I}{\delta y}dy + \frac{\delta I}{\delta t}dt \tag{5.2}$$

And by joining equations 5.1 and 5.2 we can simplify the expression as following:

$$\frac{\delta I}{\delta x}dx + \frac{\delta I}{\delta y}dy + \frac{\delta I}{\delta t}dt = 0 \tag{5.3}$$

Then, by dividing the equation 5.3 by dt, we can rewrite the expression and obtain velocities in x and y direction such as:

$$\begin{aligned} \frac{\delta I}{\delta x}\frac{dx}{dt} + \frac{\delta I}{\delta y}\frac{dy}{dt} + \frac{\delta I}{\delta t} &= 0 \\ I_x v_x + I_y \delta y v_y + I_t &= 0 \end{aligned} \tag{5.4}$$

with $v_x = \dfrac{dx}{dt}$ and $v_y = \dfrac{dy}{dt}$ being respectively the velocity in x and y direction and are the components of the OF vector, and $I_x = \dfrac{\delta I}{\delta x}$, $I_y = \dfrac{\delta I}{\delta y}$ and $\dfrac{\delta I}{\delta t}$ being the derivative components of the intensity. Finally we have:

$$\begin{aligned} I_x v_x + I_y v_y &= -I_t \\ \nabla I \cdot V &= -I_t \end{aligned} \tag{5.5}$$

with $\nabla I = (I_x, I_y)$ the spatial gradient of the intensity and $V = (v_x, v_y)^T$ the optical flow vector.

2.1 Selection of the Optical Flow Estimator

We consider thereafter the following OF methods : Farneback (Farnebäck, 2003), Beyond Pixel (BP) (Liu, 2009), Dense Inverse Search (DIS) (Kroeger et al., 2016), TVL1 (Zach et al., 2007) and DeepFlow (Weinzaepfel et al., 2013). The quality of each method is evaluated with usual metrics such as Mean Squared Error (MSE) after motion compensation, also called interpolation error (Baker et al., 2011). This metric is based on the equation 5.1, stating that the image at $t+1$ can be reconstructed using image at t. This reconstructed image is called the *compensated image* and is a registration of the image at t using the motion vectors between the two images at t and $t+1$. Therefore, the MSE of motion compensation computation follows the equation:

$$\begin{aligned} MSE &= \frac{\sum_{x=k_W}^{W_I-k_W-1} \sum_{y=k_H}^{H_I-k_H-1} (I(x,y,t+1) - I(x+v_{x_{x,y}}, y+v_{y_{x,y}}, t))^2}{(W_I - 2k_W) * (H_I - 2k_H)} \\ &= \frac{\sum_{p\in\Omega} (I(p) - I_C(p))^2}{N_\Omega} \end{aligned} \tag{5.6}$$

with I and I_C the image and its compensated image of size $(W_I \times H_I)$, p the pixel defined in space Ω: $[k_W, W-k_W-1] \times [k_H, H-k_H-1]$ with k_W and k_H are arbitrary set to 20 pixels to avoid borders effects when doing the warping. k_W and k_H could also be set according to the size of the image and/or the motion amplitude but the results order of magnitude and ranking of the estimators were not impacted when we tested them with different values. $V = (v_x, v_y)^T$ being the computed optical flow, which has the same size of the image.

The popular Sintel benchmark introduced by Butler et al. (2012) is used. This dataset of synthetic videos has available reference optical flows, with some sequences containing strong aliasing effects and random texture. Bigger datasets exist in order to train models to predict OF (Mayer et al., 2016) such as FlowNet2 (Ilg et al., 2017). Since we are only interested in selecting the OF method, the Sintel benchmark is sufficient. The availability of the optical flow ground truth allows us to measure the efficiency of motion estimation methods in terms of angular error (AE) (Barron et al., 1994) and end-point error (EPE) (Otte and Nagel, 1994). These measures can be written in matrix form in equations 5.7 and 5.8 along with their respective average value average AE (aAE) and average EPE (aEPE):

$$
\begin{aligned}
AE &= \cos^{-1}\left(\frac{\epsilon + v_x^* \odot v_x + v_y^* \odot v_y}{\sqrt{\epsilon + v_x^{*2} + v_y^{*2}} + \sqrt{\epsilon + v_x^2 + v_y^2}}\right) \\
aAE &= \frac{\sum_{x=0}^{W_I-1} \sum_{y=0}^{H_I-1} AE_{x,y}}{\sqrt{W_I * H_I}}
\end{aligned}
\tag{5.7}
$$

with $\odot$ being the Hadamard product or *element-wise product*.

$$
\begin{aligned}
EPE &= \|V^* - V\|_2 \\
&= \sqrt{(v_x^* - v_x)^2 + (v_y^* - v_y)^2} \\
aEPE &= \frac{\sum_{x=0}^{W_I-1} \sum_{y=0}^{H_I-1} EPE_{x,y}}{W_I * H_I}
\end{aligned}
\tag{5.8}
$$

The metrics: aAE and aEPE are averaged for each frame, like the MSE metric, and are computed for the whole Sintel Benchmark dataset. To assess the quality of motion estimation in our context, we compute the MSE on a "difficult" sequence from `TTStroke-21`, which is a strong-motion sequence: an *Offensive Forehand Hit* stroke lasting 240 frames, and is incorporated in Table 5.1. Computation time, denoted as "Time", is measured on the *Offensive Forehand Hit* video segment, meaning 240 frames of size (320×180), using one thread on Intel(R) Xeon(R) Gold 5118 Central Processing Unit (CPU) @ 2.30GHz. All the metrics introduced are presented as their average over the whole dataset that we therefore denote average MSE (aMSE), average aAE (aaAE) and average aEPE (aaEPE), in Table 5.1 for each OF estimation method.

It is well-known that OF methods may have difficulties on flat areas (aperture problem) and can give noisy results on highly contrasted borders due to aliasing effects. A smaller MSE thus does not automatically yield better optical flow estimation. As several kinds of assessment are necessary, different normalization methods are considered for different computed OF to be used for classification.

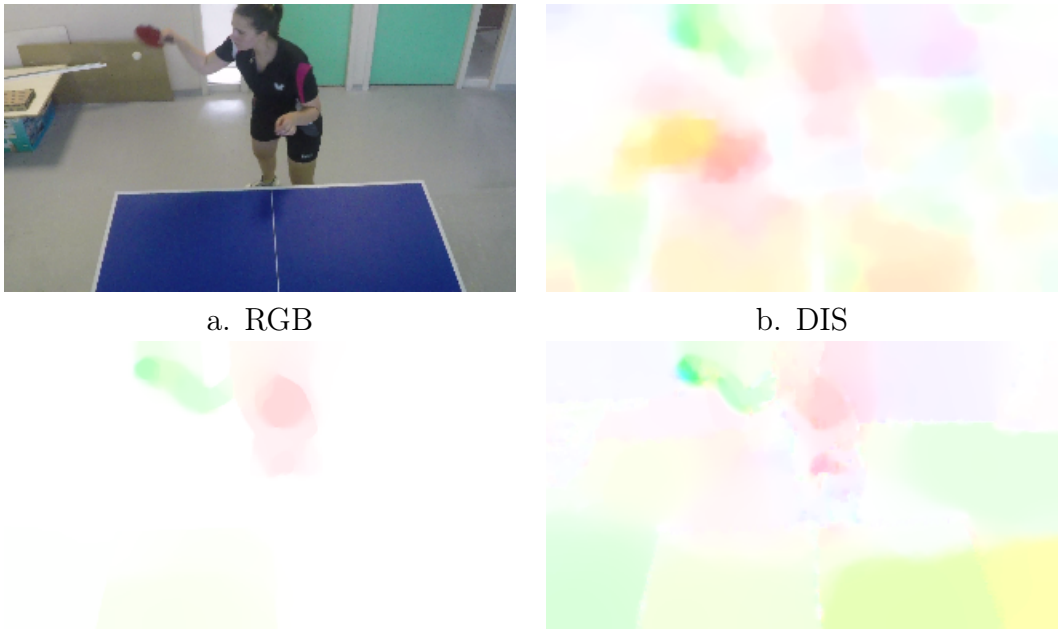
Performances of the OF estimator with respect to different metrics are shown in Table 5.1. It can be observed that the BP method (Liu, 2009) does not perform well on Sintel Benchmark as it is very sensitive to random textures. When considering the *Offensive Forehand Hit* stroke sequence and the aMSE metrics, the best method is DIS estimator with spatial propagation and preset parameters (denoted "`Medium`" in OpenCV). DIS method aimed at reducing time complexity but still yields competitive accuracy. However, aMSE is not a good metric for evaluating an OF estimator

Table 5.1 – Optical Flow methods comparison.

	Sintel Benchmark			Offensive Forehand Hit	
	aaEPE	aaAE	aMSE	aMSE	Time in s
Frame Diff	-	-	872.2 ± 1017.3	33.2 ± 12.1	-
Ground Truth	-	-	407.7 ± 778.4	-	-
Farneback	10.76 ± 18.15	$.694 \pm .328$	364.9 ± 771.6	20.9 ± 7.83	26.63
BP	6.44 ± 13.39	$.42 \pm .27$	316.5 ± 628.2	20.3 ± 3.91	617.09
DeepFlow	6.81 ± 15.27	$.374 \pm .259$	384 ± 807.6	24.1 ± 5.78	262.74
DeepFlow with matches	**2.64±5.65**	**.299±.183**	347.9 ± 711.2	24.9 ± 5.18	494.84
TVL1	9.26 ± 17.03	$.535 \pm .317$	423.3 ± 752.6	20.4 ± 4.08	617.9
DIS Medium	5.46 ± 10.88	$.461 \pm .287$	**289.7±539.6**	20.1 ± 3.92	22.72
DIS Medium†	4.82 ± 11.39	$.438 \pm .261$	318.1 ± 670.2	**19.8±4.61**	22.69
DIS Medium*	4.83 ± 9.80	$.429 \pm .265$	296.9 ± 559.2	20 ± 3.89	22.41
DIS Medium†*	4.83 ± 9.80	$.429 \pm 2.65$	296.9 ± 559.2	20 ± 3.89	22.19
DIS Fast	6 ± 11.5	$.515 \pm .307$	298.6 ± 526.1	24.2 ± 5.72	10.33
DIS Fast†	5.29 ± 11.13	$.492 \pm .279$	312.2 ± 605.5	23.8 ± 5.57	10.29
DIS Fast*	5.15 ± 9.84	$.482 \pm .281$	295.5 ± 523	24 ± 5.63	10.55
DIS Fast†*	4.94 ± 11.93	$.481 \pm .273$	320.9 ± 655	23.8 ± 5.57	10.70
DIS Ultrafast	7.08 ± 13.05	$.574 \pm .315$	307.8 ± 536	23.4 ± 5.58	1.45
DIS Ultrafast†	5.59 ± 10.91	$.528 \pm .28$	314.3 ± 601	22.9 ± 5.31	1.58
DIS Ultrafast*	5.91 ± 10.95	$.529 \pm .287$	296.7 ± 513.1	23 ± 5.37	1.72
DIS Ultrafast†*	5.16 ± 11.23	$.513 \pm .27$	323.3 ± 653.5	22.8 ± 5.21	1.87

* : with Temporal propagation † : with Spatial propagation

due to aliasing and texture effects, occlusions and illumination variations. Indeed, if the texture changes suddenly, or an occlusion appears, even with an accurate motion, the compensated image will be still much different from the image desired. Hence aMSE value on Sintel Benchmark computed with available ground truth OF is higher than that one supplied by the most of OF estimators. Its standard deviation is also very high showing the complexity of the Sintel dataset and limitation of aMSE metric. According to average angular and average end-point errors, the best estimator is DeepFlow, a variational approach for optical flow estimation, combined with matching algorithm (Weinzaepfel et al., 2013). As illustrated in Figure 5.1, DeepFlow method does not generate false movements on flat regions contrary to DIS Medium estimator with spatial propagation which performs the best with respect to MSE. Computation time is hundred times faster with DIS estimator than with DeepFLow or BP estimators. However, computation time is not here a crucial issue, because we are not interested in online predictions but in obtaining a good OF for fine-grained classification. Hence, we will focus in this chapter on DeepFlow OF estimator, which seems to predict accurate motion, and BP OF estimator as BP method was the reference used within our research team.



a. RGB b. DIS

c. DeepFlow d. BP

Figure 5.1 – Optical Flow estimators comparison. The OF values are visualized by converting $\mathbf{V} = (v_x, v_y)^T$ into an image in the color domain HSL where the Hue represents the polar angle of $\mathbf{V}$, the Saturation is set to one and the Lightness represents the amplitude of the motion.

Table 5.2 – Execution time when loading 1000 optical flow frames using several formats.

Extension	Files	Precision	size in MB	load time in s.
.png Boutell (1997)	2000	$int16$	**93.5**	3.22
.flo Baker et al. (2011)	1000	$float32$	461	.422
.npy Kern (2007)	1000	$int16$	231	.655
.npy Kern (2007)	1000	$float32$	461	**.317**
.npz SciPy community (2008)	1	$float32$	159	2.71

2.2 Storing the Computed Optical Flow

We also compared different methods for storing the optical flow in order to limit the disk usage and to have fast load of the data when we perform online data augmentation. In Table 5.2, we compare different formats according to different criteria (number of files, precision, size and loading time). The experiment was done with an Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz using one thread, and for 1000 optical flow frames. Once load, optical flow data are reshaped to get an array with direction components as channels.

The `png` format (Portable Network Graphics) (Boutell, 1997), allows to save gray scale images at 16 bits. We use this method to save each direction component of the optical flow $\mathbf{V} = (v_x, v_y)^T$ with v_x the horizontal movement and v_y the vertical movement. The values are shift and scaled before saving and need to be retransformed when loaded. Therefore we have two `png` files with very low usage disk space for each frame.

The `flo` format (Baker et al., 2011) stores 2-band float image for horizontal and vertical flow components for each frame and is commonly used for optical flow dataset. (Butler et al., 2012).

The `npy` format (Kern, 2007) is the standard binary file format in NumPy. It stores the shape of the array and the type of data. We tried using 16 bits integer and 32 bits float. In the 16 bits, values, similarly to `png`, are shift and scaled when saved - and are retransformed when loaded. This format type is similar to`flo` and is designed to be simple and portable.

The `npz` format is the standard format for multiple NumPy arrays. It is a zip file containing multiple .npy files. The gain in memory is interesting but loading time is increased.

The trade-off between disk space usage and loading time is different according to the type of applications. In our case, we are interested in training models with online data augmentation, which means data needs to be loaded several times. We thus decided to use the `npy` format which is the fastest but also the simplest method. As most of our codes use the NumPy library, NumPy arrays can be easily handled when doing online data augmentation.

3 Proposed Method for Action Classification

Our goal is to classify actions of a single table tennis player performing a series of strokes in training and match context. To limit computation cost, full HD video frames are resized to 320×180 pixels and OF is computed offline. A spatial ROI of size $(W \times H)$ is then extracted based on the foreground (Zivkovic and van der Heijden, 2006) OF values but using dilation of a mask and smoothed OF values. Our model is feed with 3D tensors of the same size $(W \times H \times T)$ than the OF. The model is tested for classification and joint detection and classification on the `TTStroke-21` dataset.

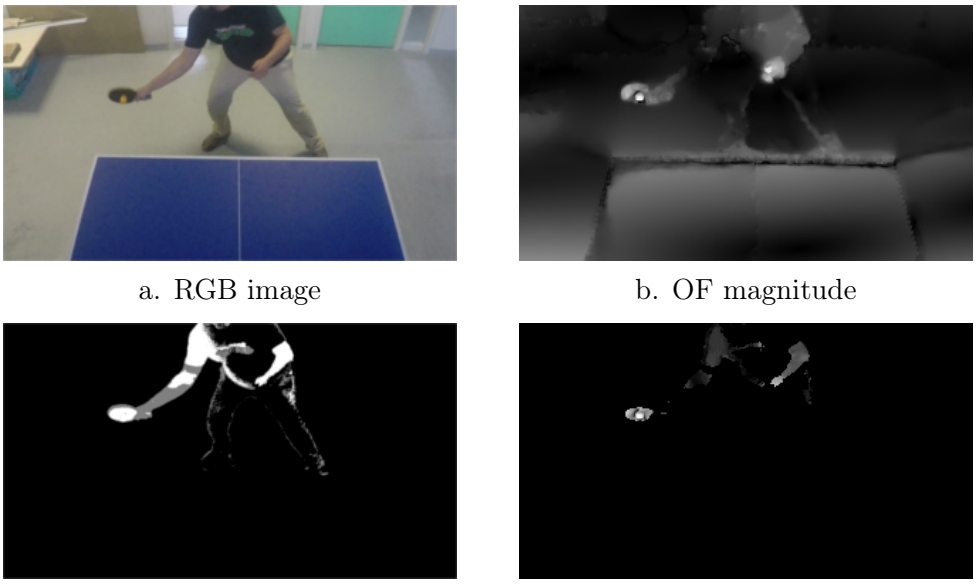
Optical Flow Filtering

Due to flickering caused by artificial light during recording sessions in sport halls, some artifacts appear on the computed OF. Those artefacts are visible in Figure 5.1, especially with the DIS and BP estimators, and on the OF magnitude in Figure 5.2. We tried to normalize the histogram but results were not relevant. We believed flickering could also be learned by the model but early results on non filtered OF were unsatisfying. To cope with that, only Regions-of-Interest (ROIs) were kept, using the Hadamard product between the foreground extracted with the method of Zivkovic and van der Heijden (2006) and the computed optical flow. After this step, parts of the OF which are not considered as background are kept. They mainly correspond to the rackets and body parts of the player which are in motion. Such method has also been applied by Chen et al. (2018b) but on RGB data for person search. The estimated foreground and the filtered OF magnitude are presented in Figure 5.2.

The foreground is not perfectly estimated still since the arm, even in motion, as color values similar to the floor, and is considered as shadows. Attempt to consider person segmentation with pretrained model on images (He et al., 2020a) were done but results were not very robust, since the temporal information was not used. New models trained on videos and more efficient were introduced very recently, but have not been tested yet to filter foreground motion.

Region-of-Interest Extraction

The ROI is estimated using the optical flow values as presented in Chapter 4, Section 2.2. The ROI center $\mathbf{X_{roi}} = (x_{roi}, y_{roi})$ is estimated from the maximum of the optical flow norm and the center of gravity of all pixels with non-null optical flow norm as follows:



a. RGB image b. OF magnitude

c. Estimated foreground d. OF magnitude after filtering

Figure 5.2 – Optical Flow filtering.

$$
\begin{aligned}
&\mathbf{X_{max}} = (x_{max}, y_{max}) = \underset{x,y}{argmax}(||\mathbf{V}||_1) \\
&\mathbf{X_g} = (x_g, y_g) = \frac{1}{\sum_{\mathbf{X}\in\Omega}\delta(\mathbf{X})}\sum_{\mathbf{X}\in\Omega}\mathbf{X}\delta(\mathbf{X}) \\
&\text{with} \quad \delta(\mathbf{X}) = \begin{cases} 1 & \text{if } ||\mathbf{V}(\mathbf{X})||_1 \neq 0 \\ 0 & \text{otherwise} \end{cases} \\
&x_{roi} = \alpha\ f_{\omega_x}(x_{max},\ W) + (1-\alpha)\ f_{\omega_x}(x_g,\ W) \\
&y_{roi} = \alpha\ f_{\omega_y}(y_{max},\ H) + (1-\alpha)\ f_{\omega_y}(x_g,\ H)
\end{aligned}
\tag{5.9}
$$

with parameters $\alpha = 0.6$, $\mathbf{\Omega} = (\omega_x, \omega_y) = (320, 180)$ being the size of video frames. Function $f_\omega(u, S) = max(min(u, \omega - \frac{S}{2}),\ \frac{S}{2})$ allows to have data inputted to our network within the boundaries of the region of interest. To avoid jittering within our cuboids of size $(W \times H \times T)$, we also apply a Gaussian filter using a kernel of size k_{size} with scale parameter $\sigma_{blur} = 0.3 * ((k_{size} - 1) * 0.5 - 1) + 0.8$ along the temporal dimension to average the center position.

3.1 Normalization

For normalizing the estimated motion $\mathbf{V} = (v_x, v_y)^T$, different approaches are possible. Four methods have been tested: the first one is to normalize each component of $\mathbf{V}$ by its maximum absolute value over the whole dataset. We reference this method as "`MAX`". The second method is to normalize each component of $\mathbf{V}$ by the mean μ and the standard deviation σ of the distribution over the whole dataset of frame

maximum absolute values. We reference this method as "NORMAL" and the operation is described in equation 5.10. In the two following equations v and v^N represent respectively one component of the OF $\mathbf{V}$ and its normalization.

$$\begin{aligned} v' &= \frac{v}{\mu + 3 \times \sigma} \\ v^N(i,j) &= \begin{cases} v'(i,j) & \text{if } |v'(i,j)| < 1 \\ SIGN(v'(i,j)) & \text{otherwise.} \end{cases} \end{aligned} \tag{5.10}$$

The third method, denoted "LOGMAX", is similar to the MAX normalization method. We use the same concept but using the absolutes log values of the OF shifted to one. The log values are then remapped according to their initial sign.

The fourth and last method that we reference as "LOGNORMAL", is similar to the NORMAL normalization. But as the previous method, we use the logarithm of the distribution over the whole dataset of frame maximum absolute values of a component on the log values shifted and resigned. The normalization is detailed in equation 5.11.

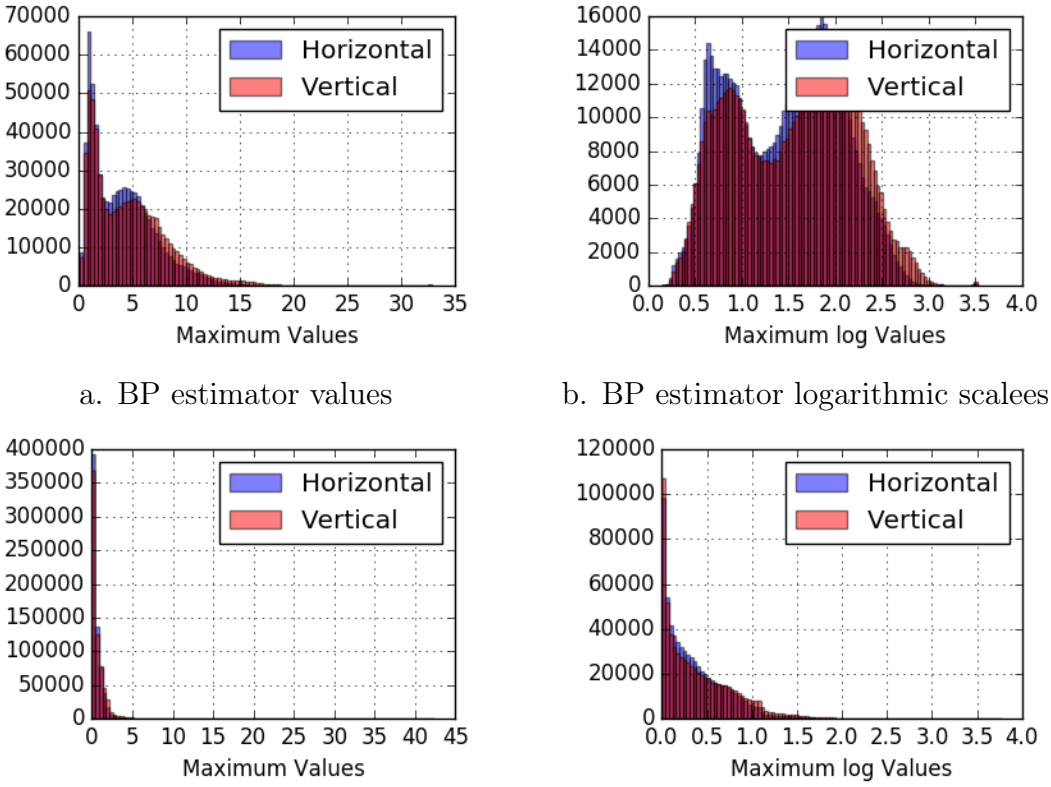
$$\begin{aligned} v_{log} &= log(|v| + 1) \\ v' &= \frac{v_{log} - (\mu_{log} - 3 \times \sigma_{log})}{6 \times \sigma_{log}} \\ v_n(i,j) &= \begin{cases} 0 & \text{if } v'(i,j) \leq 0 \\ SIGN(v(i,j)) \times v'(i,j) & \text{if } 0 < v'(i,j) < 1 \\ SIGN(v(i,j)) & \text{otherwise.} \end{cases} \end{aligned} \tag{5.11}$$

Obviously, the MAX method strongly reduces the magnitude of most motion vectors. However the NORMAL normalization method increases the magnitude of most vectors, while LOGMAX and LOGNORMAL should flatten the values distribution. The distribution of the maximum absolute motion values in each direction of BP and DeepFlow estimators, which are going to be used for classification in this chapter, and their logarithmic shifted to one are presented in Figure 5.3. We can notice how the shapes of the distributions are different depending on the two estimators.

A polar representation of the OF is also possible. The motion is then encoded through the amplitude of the motion and its direction (angle according to the vertical axis). Such representation may lead to interesting statistical structures as presented by Adato et al. (2011). However, after having performed tests using such representation by normalizing the amplitude with similar method than MAX and NORMAL, it led to lower classification scores and no further investigation was conducted.

3.2 Architecture of the Flow Spatio-Temporal Convolutional Neural Network

The Flow-STCNN architecture presented in Figure 5.4 is very similar to the RGB-STCNN presented previously. It is constituted of an individual branch of three



a. BP estimator values

b. BP estimator logarithmic scalees

c. DeepFlow estimator values

d. DeepFlow estimator logarithmic scales

Figure 5.3 – Optical Flow maximum absolute values distribution.

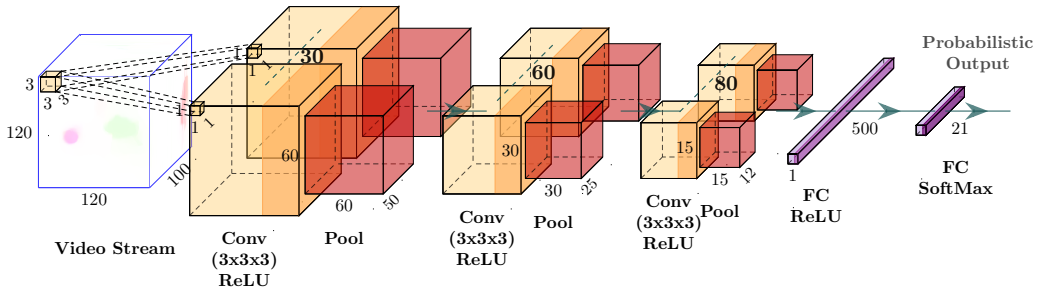


Figure 5.4 – The Flow-STCNN architecture.

3D convolutional layers with 30, 60, 80 filter response maps, followed by a fully connected layer of size 500. The branch takes the preliminary estimated OF as 3D tensor of size ($W \times H \times T$) with two channels encoding horizontal and vertical motion v_x, v_y. The 3D convolutional layers use $3 \times 3 \times 3$ space-time filters with a dense stride and padding set to one in each direction. The extracted features of size 500 are processed through a last fully connected layer of size 21 followed by a Softmax function for computing a classification score.

3.3 Model Training

The Flow-STCNN model is trained the same way than the RGB-STCNN . It uses the popular stochastic gradient descent optimization method with Nesterov momentum (Nesterov, 2004). We chose a constant learning rate of 0.01 for training the model. The other parameters are momentum of 0.5 which is decrease to 0.1 at epoch 1000, weight decays of 0.005, and a batch size of ten. Since we had to test different flow estimation methods and different normalization methods, we set the number of epochs to 500 when comparing the different methods and increased this number to 1500 for the best combination.

3.4 Performance Evaluation

Classification Task

To compare the performances of our model in the classification task, we use the **Flow-I3D** CNN introduced by Carreira and Zisserman (2017) as baseline. It is the same baseline than used in Chapter 4 with the RGB-STCNN model but using the Optical Flow data. It also uses `inception_v1` architecture (Szegedy et al., 2015), which is applied to our dataset following their instructions for training. Their model needs to be trained with more iterations than the RGB-I3D: 155 000 iterations are needed (against 115 000) which represents 1148 epochs with our training set from `TTStroke-21`. In addition, the learning rate is scheduled, changing from 0.1 to 0.01 and 0.001 respectively at iterations 97 000 and 108 000, and again to 0.01 and 0.001 respectively at iterations 140 000 and 150 000. The authors noticed that their Flow models required more training after an initial run of 115k iterations. Since our input data are twice smaller than theirs, the first max-pooling layer of the Flow-I3D model is discarded.

For this task, the goal is to recognize the class of an already localized stroke. To evaluate our models on the test set we use the same four different rules presented in Chapter 4, Section 2.5: “Test”, ‘TVote”, “TAvg” and “TGauss”.

Detection by Classification

The joint detection and classification in videos is done through the classification of segments of video using a sliding window of size T with step one. This process

is long and is performed only on our models performing well on the classification only task. The same method and rules as in Chapter 4, Section 2.5 are used for evaluating the performances: "Gross", "Vote", "Average" and "Gaussian".

To evaluate the performances of our method for detection and classifications in videos, we compare our predictions with the ground truth built from the crowd-sourced annotations of `TTStroke-21` dataset. Since the videos are limited in the diversity of strokes, experiments for this task have been conducted with the whole dataset which incorporates strokes and negative samples that were in the training, validation and test sets.

3.5 Data Augmentation

Data augmentation is a necessary step for training DNNs, particularly for fine-grained classification where data are more difficult to get. As for the RGB-STCNN, it is performed on the fly to save storage space. For spatial augmentation we apply random rotation with angle θ in the range $\pm 10°$, a random translation (t_x, t_y) in the range ± 0.1 in x and y directions, and a random homothety k in range 1 ± 0.1 on OF data V. For the latter, rotation needs special care for Flow data and is performed such as $V = R(\theta) * V$ with $R(\theta)$ being the rotation matrix of angle θ. Transformations are applied with respect to the center of the ROI. Finally we perform horizontal flip with probability of 0.5. Note that the flip on optical flow means also changing the sign of v_x.

4 Experiments and Results

Experiments are performed on the same splits of `TTStroke-21` (Chapter 4, Section 4.1). T is set to 100 frames, which represents 0.83 seconds at 120 fps. Our model is then fed with cuboids OF data of size $(W \times H \times T) = (120 \times 120 \times 100)$. Different normalization methods are tested and models are trained longer for the classification and joint classification and detection tasks.

4.1 Influence of Normalization Method on Classification

Table 5.3 shows the performances of the Flow-STCNN classifier trained with 500 epochs using BP and DeepFlow OF estimators with the four normalization methods presented earlier. It is a pure classification task, as temporal borders of each action are known.

From the table, we can observe that the best performances are obtained using the `NORMAL` method with BP flow estimator. As explained in Section 3.1, the `NORMAL` method spreads the low values but does not concentrated them as the logarithmic scaling does. It is also a way to get rid of the noise from the OF computation which leads to very high values and makes the `MAX` method less efficient. Corresponding

Table 5.3 – Performances of the Optical Flow normalization methods.

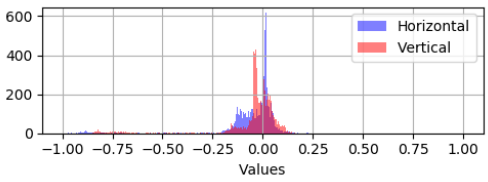
	Accuracies in %					
Normalization	**Train**	**Val**	**Test**	**TVote**	**TAvg**	**TGauss**
BP						
MAX	53.5	44.4	43.1	44	44	44.7
NORMAL	88.5	73.5	69.8	72.4	**74.1**	73.3
LOGMAX	50.3	50	50	50	51.7	46.6
LOGNORMAL	97.8	75.7	65.5	66.4	68.1	68.1
DeepFlow						
MAX	38.5	36.5	25	28.5	27.6	28.5
NORMAL	34	35.7	25.9	25.9	26.7	26.7
LOGMAX	49.5	48.3	43.1	37.9	39.7	40.5
LOGNORMAL	45.3	37	35.3	40.5	41.4	41.4

histograms of initial v_x, v_y values computed using BP method after filtering are presented in Figure 5.5 with NORMAL normalization (b) and LOGMAX normalization (c). As we can see, the flow values distribution with the LOGMAX normalization method is much more distributed than with the NORMAL normalization method. We might think it thus should lead to better performances but it actually gives more weight to regions that are not needed for classification. Experiments without filtering process were also conducted but led to lower performances.

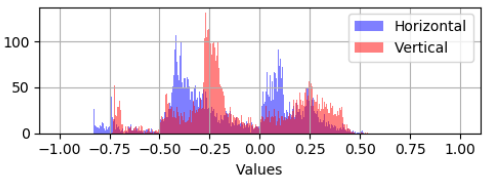
However our model trained with OF from DeepFlow estimator did not perform well. This actually can be explained by looking at the maximum value distribution of the method, Figure 5.3. We can notice how much the maximum values are more concentrated on the lowest range, even using logarithmic transformation. Furthermore, by looking back at the Figure 5.1, we can notice how DeepFlow is much smoother compared to BP. The visual result may seem correct, and the evaluation metrics too, but it actually leads to misdetect movements on little objects moving fast such as the ball while BP captures it. We thought that DeepFlow would have been a good estimator for classification according to the evaluation metrics presented in Section 2 but fast movement on little objects, such as the ball or racket, did not increase much the evaluation metrics but are fundamental for our fine-grained classification task.

The gap between the classification score across normalization method and flow estimator underlines the importance of the choices made when deciding which modality to use and how to process it. Even when training with a greater number of epochs, we observed that NORMAL normalization performs the best. According to these results we decided to use the NORMAL normalization method with BP OF estimator for the next trainings.

a. RGB image



b. BP OF with `NORMAL` method



c. BP OF with `LOGMAX` method

Figure 5.5 – Different OF normalization and their histogram.

4.2 Pure Classification Task

The results for the pure classification task are depicted in Table 5.4. The models were trained using $T = 100$ as we have done previously. Models using only 64 frames were not convincing.

Table 5.4 – Performance comparison between Flow-I3D (Carreira and Zisserman, 2017) and Flow-STCNN (Ours) on pure classification task.

	Accuracies in %					
Models	**Train**	**Val**	**Test**	**TVote**	**TAvg**	**TGauss**
Flow-I3D	98.8	74.8	73.3	**82.8**	**82.8**	**82.8**
Flow-STCNN*	91	78.7	67.2	71.6	70.7	70.7
Flow-STCNN	97.5	79.6	75.9	80.2	80.2	78.5

* trained without data augmentation

The use of data augmentation improves the accuracy of our Flow-STCNN model from 71.6% to 80.2% with "TVote" and "TAvg" evaluation methods. Convergence of our model is presented in Figure 5.6. Best model is saved at epoch 1 449 when its validation is the best. No strong overfitting is observed compared to our baselines convergences which is presented in Figure 5.7.

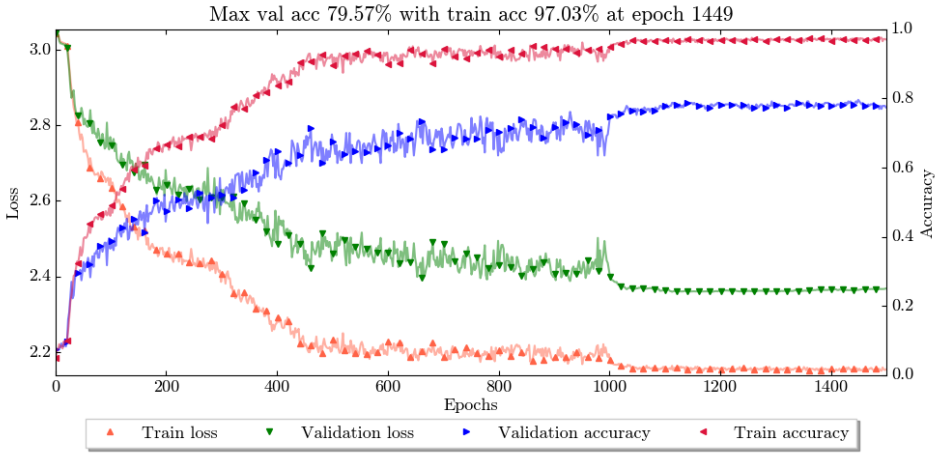


Figure 5.6 – Training process of the Flow-STCNN model with $T = 100$.

The different normalization methods did not have any impact on our baseline and here we show performances using the `MAX` normalization for the Flow-I3D model. This behavior can be explained by the use of batch normalization in their network after each inception_v1 modules (Szegedy et al., 2015) and the deepness of their network. Indeed it uses nine inception modules and four additional convolutional

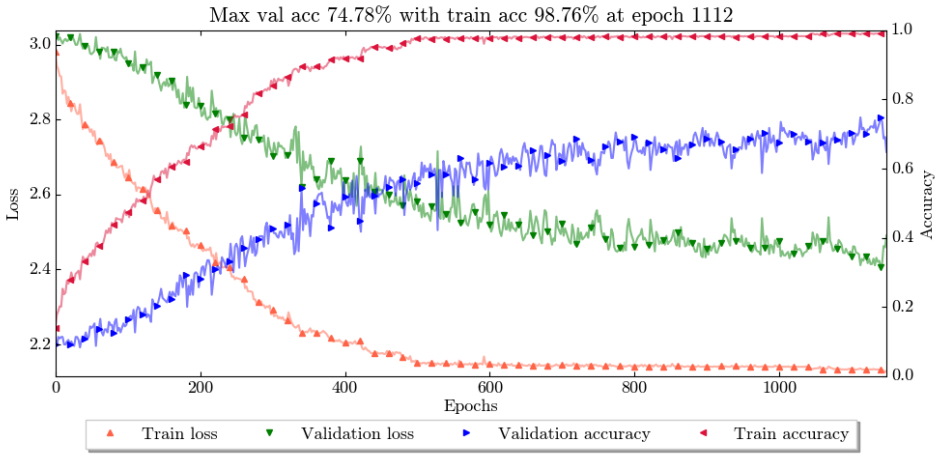


Figure 5.7 – Training process of the Flow-I3D model with $T = 100$.

layers against three convolutional layers for us. Their network, because of its complexity, might not need to pre-process the OF data before extracting features. In addition, their model beats ours to 2.6% in term of classification accuracy when using the temporal methods for classifying the video samples ("TVote", "TAvg" and "TGauss" method). However if we consider only center features ("Test" method) our model outperform theirs by also 2.6%.

As seen in Chapter 4 with the RGB-I3D model, the convergence is similar. Note that the training is longer when using the OF modalities as described in Section 3.3.

In Figure 5.7 we can observe a gap between the curves of the validation and train sets which let suggest an overfitting from the I3D model. Despite its better performances on the classification task, the curves of the Flow-STCNN model, Figure 5.6, shows a tinier gap between the validation and training curves.

4.3 Joint Stroke Detection and Classification Task

Same method is used that in Chapter 4 for evaluating the detection and classification task. Experiments have been conducted with the whole dataset which incorporates strokes and negative samples in the training, validation and test sets. Each model classifies the entire video with a temporal sliding window $T = 100$ frames with step one. Majority vote and average probability method use a sliding decision window $WD = 1.5T = 150$ also with step one. For the decision rule based on a Gaussian filter, a kernel of size $2T + 1 = 201$ and a scale parameter $\sigma = 0.5T = 50$ were used. As the detection may not be always precise in time because of errors in the crowdsourced annotations, a prediction is considered correct at the boundaries of actions (between two classes) if one of these classes is found. The maximum possible class overlap is set to 20% of the current stroke duration. Results are shown in Table 5.5 for our Flow-STCNN.

Table 5.5 – Performance of stroke detection and classification.

	Accuracies in %			
Models	**Gross**	**Vote**	**Average**	**Gaussian**
Flow-STCNN*	68.3	77.4	78.1	77.9
Flow-STCNN	70.3	80.5	80.9	**81**
without taking into account negative labels				
Flow-STCNN*	44.9	48.3	49.2	52.8
Flow-STCNN	50.4	55.4	59.2	**62.4**

* trained without data augmentation

The table shows the performance with and without using the data augmentation with and without considering the negative samples. As specified in the last chapter, the video may be mainly composed of negative samples, when players take a break or when they miss a ball, which motivates this analyse separation. Unlike the RGB-STCNN, the Flow-STCNN with data augmentation obtains better performances in both cases using the "Average" and "Gaussian" methods. Difference comes certainly from the type of data modality we are working with. Indeed the ball on RGB data might be not be totally visible since its color can be similar to the background. Hence, features coming from this source are less considered, especially when using data augmentation since the ball will even be less present. While when using optical flow with BP estimator, the ball is well detected and is characterized by higher values spatially and temporally in the tensor. Therefore strokes should be well detected with or without data augmentation. Hence, better classification of the stroke will then depend on features extracted before and after impact on the racket. Such features are more processed with the Flow-STCNN when trained using data augmentation explaining this behaviour. Also, this ability to capture better the features coming from the ball, and thus better classify the non-stroke segment, explains the slightly better performance of the Flow-STCNN (81%) compared with RGB-STCNN (80.9%) when negative samples are considered.

Also the drop of performance when not considering the negative labels is higher than with the RGB-STCNN model. It can be easily explained by the ability of the Flow-STCNN to well classify negative samples using the OF data, since motion is less prominent in this class compared to strokes samples. Which leads us to the conclusion that RGB and OF modalities should be used together to improve performances.

5 Conclusion

This chapter deals with fine-grained action recognition applied to table tennis. We apply our method on the `TTStroke-21` dataset using only estimated motion from the video stream. We have compared several optical flow methods in terms of aMSE, aAE and aEPE metrics for selecting the optimal one as input data of our Spatio-Temporal CNN - Flow-STCNN. We have proposed four normalization schemes and studied their influence with respect to the classification accuracy. Our choices led to an accuracy of 80.2% with models trained in a reasonable number of epochs. However our baseline, Carreira and Zisserman (2017), reaches a better accuracy: 82.8%. Nonetheless, our model performs the best when we consider only centered features. We also showed that for both pure classification, and classification and detection tasks, proposed data augmentation led to better scores. In addition, we believe that results can be improved by fusing the methods developed in Chapter 4. In the next chapter, we present our work on how to fuse RGB and OF modalities, and the work carried out in those two last chapters, for fine-grained action classification.

Chapter 6

Twin 3D Spatial-Temporal Convolutional Neural Network for Fine-Grained Action Recognition

1 Introduction

The two previous chapters were dedicated to fine-grained action classification using only one modality, either RGB data or OF data computed from the RGB images. Those modalities require only the acquisition of images using cameras and do not affect the performances of the players. Our work is focused on detection and recognition of strokes in table tennis using Computer Vision only. It is a first step in a wide research program which goal is to give tools for sport coaches to improve performances of young athletes using recorded videos of training and playing sessions. In order to reach the largest audience, recordings have to be performed by widespread and cheap video cameras, e.g. GoPro. We use a dataset specifically recorded in a sport faculty facility and continuously completed by students and teachers: `TTStroke-21`, to train and test our method. The `TTStroke-21` dataset is described in Chapter 3, Section 4.

This dataset is constituted of player-centred videos recorded in natural conditions without markers or sensors. It comprises 20 table tennis strokes and a rejection class. The problem is hence a typical research topic in the field of video indexing: for a given recording, we need to label the video by recognizing and temporally segmenting each stroke appearing in the whole video. As we have seen in Chapter 5, motion information is a crucial clue for recognizing table tennis strokes. A review of the state-of-the art in Chapters 1 and 2, shows that spatial features from RGB images are also needed to attain reasonable accuracies for action classification, be it in sport (Varol et al., 2018) or in specific cultural content (Stoian et al., 2016). Hence, it is necessary to use both multi-modal data and temporal information. Contrary to Safaei et al. (2018); Bilen et al. (2018) who use one channel of a 3D tensor for encoding temporal information, we are using 3D convolutions of video frames, similarly to the promising results obtained in by (Tran et al., 2015; Feichtenhofer et al., 2016; Carreira and Zisserman, 2017).

However, to efficiently fuse data of limited spatial localization of moving sports-

men with variable motion magnitude for each stroke, an adequate normalization has to be done (see Chapter 5). After this step, extracted features from the RGB and OF data need to be fused efficiently to obtain relevant output probabilities for class decision. In this chapter, we use the work presented in the two last chapters for fusing efficiently both modalities. We introduce a Twin 3D convolutional neural network model, so called Twin Spatio-Temporal Convolutional Neural Network (T-STCNN), to incorporate temporal features along with spatial ones. The stroke is predicted from RGB video frames and their estimated motion vector fields. This chapter is related to some extent to our following publications: Martin et al. (2019c, 2021b, 2020c, 2019b, 2018, 2019d).

The remainder of this chapter is organized as follows: Section 2 presents the T-STCNN model, data preparation and processing, and training and evaluation methods. Results are presented in Section 3 and compared with our baseline, the Two-Stream-I3D model (Carreira and Zisserman, 2017). Conclusion and perspectives are drawn in Section 4.

2 The Twin Spatio-Temporal Convolutional Neural Network Model

For efficiently fusing RGB and OF data, this Twin Convolutional Neural Network architecture is introduced in Section 2.1. Input data are filtered before feeding them to the model for training and testing. Data augmentation is also used in order decrease intra-class similarity and help the model to better generalize extracted features. Other fusion methods are tested and presented along with the evaluation methods in Section 2.6.

2.1 Architecture of the Twin Spatio-Temporal Convolutional Neural Network

The Twin Spatio-Temporal Convolutional Neural Network, denoted as "T-STCNN" is presented in Figure 6.1. It is a two stream network constituted of two branches. Each branch follows the same structure: three blocks constituted of a 3D convolutional layer using kernels of size $(3 \times 3 \times 3)$, with stride and padding one in all directions and "ReLU" as activation function, feeding a 3D Max-Pooling layer using kernels of size $(2 \times 2 \times 2)$ and floor function. From input to output, the convolutional layers use 30, 60 and 80 filters. Each branch ends with a fully connected layer of size 500. The two branches are combined using a bilinear transformation $(y = x_1^T A x_2 + b)$ with Softmax function to output a classification score of size 21 corresponding to the number of classes considered in our task. The "Twin" appellation comes from the fact that the same configuration is used on both branches. Unlike the Two-Stream I3D model Carreira and Zisserman (2017), our network is

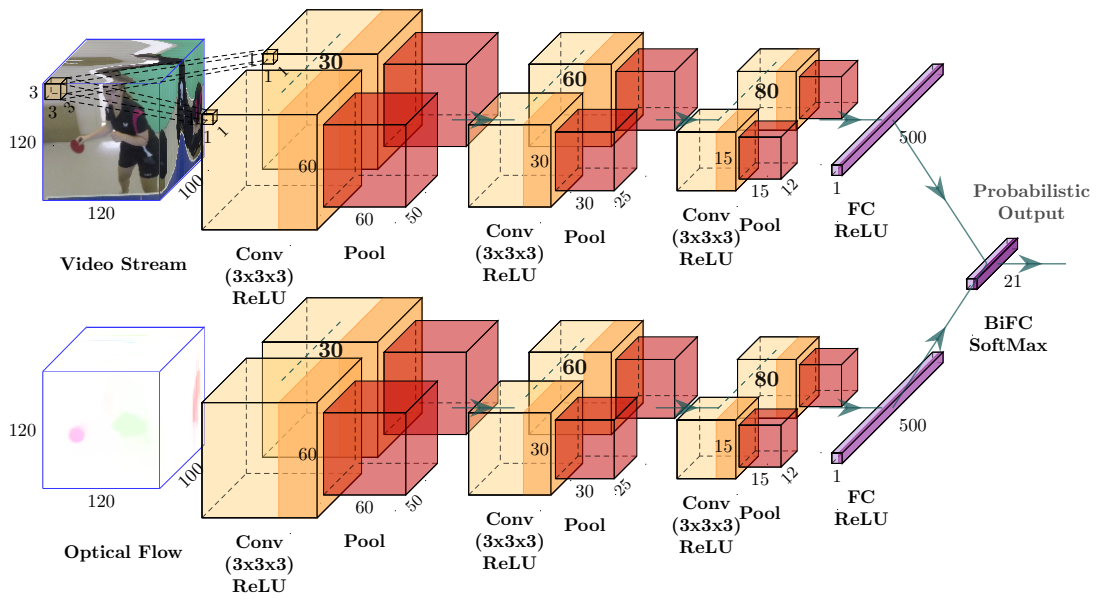


Figure 6.1 – Twin Spatio-Temporal Convolutional Neural Network - T-STCNN - architecture.

much shallower and do not simply add each streams predictions for classification; we fuse the features extracted from each stream before our last fully connected layer.

2.2 Input Data

Branches of the network take RGB images and optical flow field of size $(W \times H \times T)$ as inputs. The optical flow is computed using method BP estimator Liu (2009), based on iterative re-weighted least square solver. In the last chapter we have indeed shown its efficiency for classification purpose: even if the method of Liu (2009) is sensitive to flickering and not smooth on flat regions, it is able to capture fine details, such as the motion of the ball, in contrary to DeepFlow method Weinzaepfel et al. (2013). The extracted frames from the video size (1920×1080), are resized to (320×180) before computing the optical flow field.

Optical Flow Filtering and Region-of-Interests Extraction

The same filtering method presented in Chapter 5 is applied in this chapter. The flickering caused by artificial light during recording sessions and leading to artefacts are partially filtered using the Hadamard product between the foreground extracted with the method of Zivkovic and van der Heijden (2006) and the computed optical flow. Since RGB and Optical Flow modalities are processed together, we believe flickering could be learned and processed more easily by the model.

The ROI is estimated from the maximum of the optical flow norm and the center

of gravity of all pixels with non-null optical flow norm as described in previous chapter.

2.3 Data Normalization

Before feeding the data to the network, RGB values are mapped between [0,1] and Optical flow is mapped between [−1,1]. In the last chapter, different normalization methods were tested : `MAX`, `NORMAL`, `LOGMAX` and `LOGNORMAL`. One possible method is normalizing OF values by the `MAX` method which consists in dividing each OF channel, v_x and v_y by the absolute maximum of the OF for each direction on the whole dataset. Another consider normalisation method is the `NORMAL` normalization which normalizes each component of $\mathbf{V}$ by the mean μ and the standard deviation σ of the distribution over the whole dataset of frame maximum absolute values. The normalisation is presented in Section 6.1 with v and v^N referring respectively to one component of the OF $\mathbf{V}$ and its normalization.

$$\begin{aligned} v' &= \tfrac{v}{\mu+3\times\sigma} \\ v^N(i,j) &= \begin{cases} v'(i,j) & \text{if } |v'(i,j)| < 1 \\ SIGN(v'(i,j)) & \text{otherwise.} \end{cases} \end{aligned} \tag{6.1}$$

Results are provided in the following for both normalization methods with the T-STCNN model as they are representative of the span of performances of other methods.

2.4 Data Augmentation

Data augmentation is performed similarly than in Chapters 4 and 5. For each stroke, we extract one video sample of size $(W \times H \times T)$. Without data augmentation, the T frames from the RGB and Optical Flow are extracted at the centre of the temporal and spatial extends, according respectively to the duration of the stroke Δt and our ROI extraction.

Spatial augmentation is performed by applying random rotation in the range $\pm 10°$, random translation in x and y direction respectively in range $\pm 0.1 * W$ and $\pm 0.1 * H$, and random homothety in the range 1 ± 0.1. All transformations are applied and centered on the ROI center.

Temporal augmentation is performed extracting T successive frames following a normal distribution around the center of our stroke with standard deviation of $\sigma = \frac{\Delta t - T}{L}$, with $L = 6$, as presented in Figure 4.2. If the frames are not in the temporal boundaries of the annotated sample, another random draw is done until the condition is satisfied.

2.5 Training Step

The training process is similar to the one used for the RGB-STCNN and Flow-STCNN models previously presented. Estimation of network parameters is done using Stochastic Gradient Descent with Nesterov Momentum Sutskever et al. (2013). We use a momentum coefficient value of 0.5 and decrease it to 0.1 and 0.05 at epoch 1000 and 1500 respectively, as the momentum methods are known to oscillate at the beginning of the iterative process. We use a weight decay of 0.005. The maximum number of epochs is set to 2000. Cross-entropy loss is used as objective function and the batch size is set to ten. The number of negative samples is chosen twice bigger than the mean of the number of strokes per class. The `TTStroke-21` dataset is split into training, validation and testing sets with the respective proportions: 70%, 20% and 10% (Chapter 4, Table 4.1).

We use two different architectures: i) the Twin architecture introduced in Section 2.1 (referred as T-STCNN), and ii) a simple branch architecture similar to RGB-STCNN or Flow-STCNN using RGB images and OF concatenated together resulting in an input of a five channel tensor). Since the fusion of both modalities are performed before training, we refer to this last model as Early Fusion Spatio-Temporal Convolutional Neural Network (EF-STCNN). The Twin model uses a learning rate of 0.001 and the EF-STCNN a learning rate of 0.01.

We use data augmentation on our training set for all models and evaluate them at each epoch with the accuracy on the validation set without augmentation. Models with the best accuracy are saved for the next evaluations on the test set.

2.6 Evaluation Methods

To compare performances of our models, we use the Two-Stream I3D model Carreira and Zisserman (2017) as our baseline and apply it to our dataset following their instructions for training. The first max pooling layer has been discarded because of the size of our input data which are twice smaller than theirs. The RGB and Optical Flow models of I3D, RGB-I3D and Flow-I3D, are trained separately as presented in Chapter 4 and 5. RGB-I3D and Flow-I3D model are trained respectively with 115 000 and 155 000 iterations which represents 851 and 1148 epochs with our training set from `TTStroke-21`. The learning rate is scheduled, decreasing from 0.1 to 0.01 and 0.001 respectively at iterations 97 000 and 108 000 For the Flow-U3D model, the learning rate increases to 0.01 at iterations 140 000 and decreases to 0.001 at iteration 150 000. The reason is that the model using OF modality needs to be trained for a longer time, as precised in (Carreira and Zisserman, 2017).

The Two-Stream I3D model consists in performing a late fusion using both models to classify the stroke by summing up their obtained class scores. Finally, we also apply a late fusion operator such as summing scores of one-branch models RGB-STCNN and Flow-STCNN. We refer to it as Late Fusion Spatio-Temporal Convolutional Neural Network (LF-STCNN).

Same evaluation methods are used than in Chapter 4, Section 2.5 and are called "Test", "TVote", "TAvg" and "TGauss" for the classification task and performances are shown in Table 6.1. Evaluation of the performances for detection and classifications task in videos is done with "Gross", "Vote", "Average" and "Gaussian" methods and results are presented in Table 6.2.

3 Experiments and Results

Our deep learning models have been trained using PyTorch framework on GPU NVIDIA Tesla P100. The size of the input data have been set to $(W \times H \times T) = (120 \times 120 \times 100)$ which makes out the first fully connected layer of each branch to take as input a vector of size 216 000. Experiments are also conducted using a time window of 64 frames to keep default setting of our baseline Carreira and Zisserman (2017), which then lowers the size of the input vector of our first connected layer on each branch down to 144 000. We provide results according to the model type and by default the OF normalization method is `NORMAL`. We present some model performances which use the `MAX` normalization in Table 6.1.

3.1 Pure Classification Task

Performances of all the models presented are in Table 6.1.

Table 6.1 – Performance comparison between Two Stream-I3D, EF-STCNN, LF-STCNN and T-STCNN on classification.

		Accuracies in %					
Models	**T**	**Train**	**Val**	**Test**	**TVote**	**TAvg**	**TGauss**
Two-Stream I3D†	64	87.8	41.7	43.1	11.2	11.2	10.3
Two-Stream I3D†	100	99.2	76.2	75.9	84.5	87.1	86.2
T-STCNN	64	79.1	76.5	72	72	74.6	80.2
EF-STCNN†	100	88.4	84.4	73.3	74.1	75	75
EF-STCNN	100	90.8	84.8	82.2	81.4	83.9	83.9
LF-STCNN†	100	82.3	62.2	57.7	59.5	70.7	69.8
LF-STCNN	100	97	88.7	89.8	87.3	87.3	87.3
T-STCNN†	100	99.5	90.4	89	72	74.6	71.2
T-STCNN*	100	99.5	90.4	90.7	75.4	73.7	72.9
T-STCNN	100	95.8	87.8	**93.2**	91.5	90.7	91.5

* trained without data augmentation † : with `MAX` OF normalization

Best performance obtained for the classification task reaches 93.2% of accuracy on the train set using the T-STCNN model, trained with data augmentation, using `NORMAL` method for flow normalization and evaluated decision based on "Test"

method. Its convergence is show in Figure 6.2 and weights are saved at epoch 1784 when the model reaches its best performance on the validation set. Our model does not overfit the training dataset in contrast to the Two-Stream I3D model where the gap between validation and training performances is important.

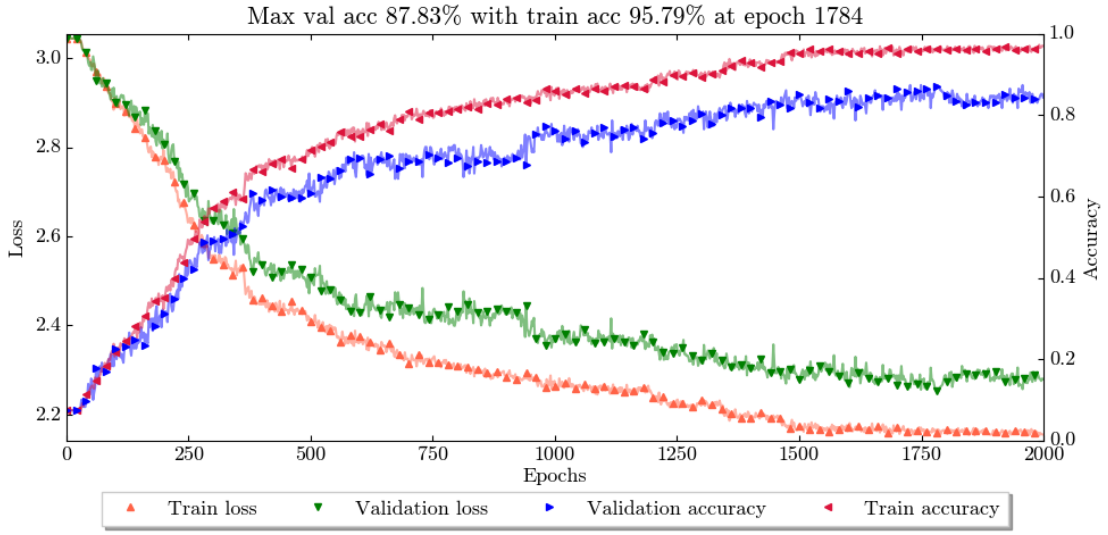


Figure 6.2 – Training process of the T-STCNN model with $T = 100$ and `NORMAL` OF normalization method.

In Figure 6.3, we can see that strokes "Defensive Forehand Block" and "Defensive Backhand Push" are misclassified. This is certainly due to their low representation in the dataset, keeping the model from learning their distinctive features. The use of the whole stroke segment for decision does not improve the results, but do not worsen them neither, in contrary to T-STCNN trained without data augmentation which drops from 90.7% down to 72.9% using "Test" and "TGauss" rule decision methods. Moreover, T-STCNN models using 100 frames are more efficient than the one using 64 frames only, which can be also noticed for I3D models (Carreira and Zisserman, 2017). It has been demonstrated that the use of longer temporal windows improves classification performances for long and similar actions Varol et al. (2018), which is our case when considering fast table tennis stokes at a frame rate of 120 per second.

The T-STCNN model with `MAX` flow normalization method gets also lower scores. Our late fusion approach, which considers the sum of the probabilistic output of the two best models RGB-STCNN and Flow-STCNN presented earlier, obtains the third best performance with an accuracy of 89.8% using "Test" decision. It also gets much better performance than the same model but using `MAX` flow normalization. Indeed the Flow-STCNN with `MAX` normalization misleads the RGB-STCNN which gets better performances than the fused model.

Early fusion using "MAX" method for flow normalization gets similar results

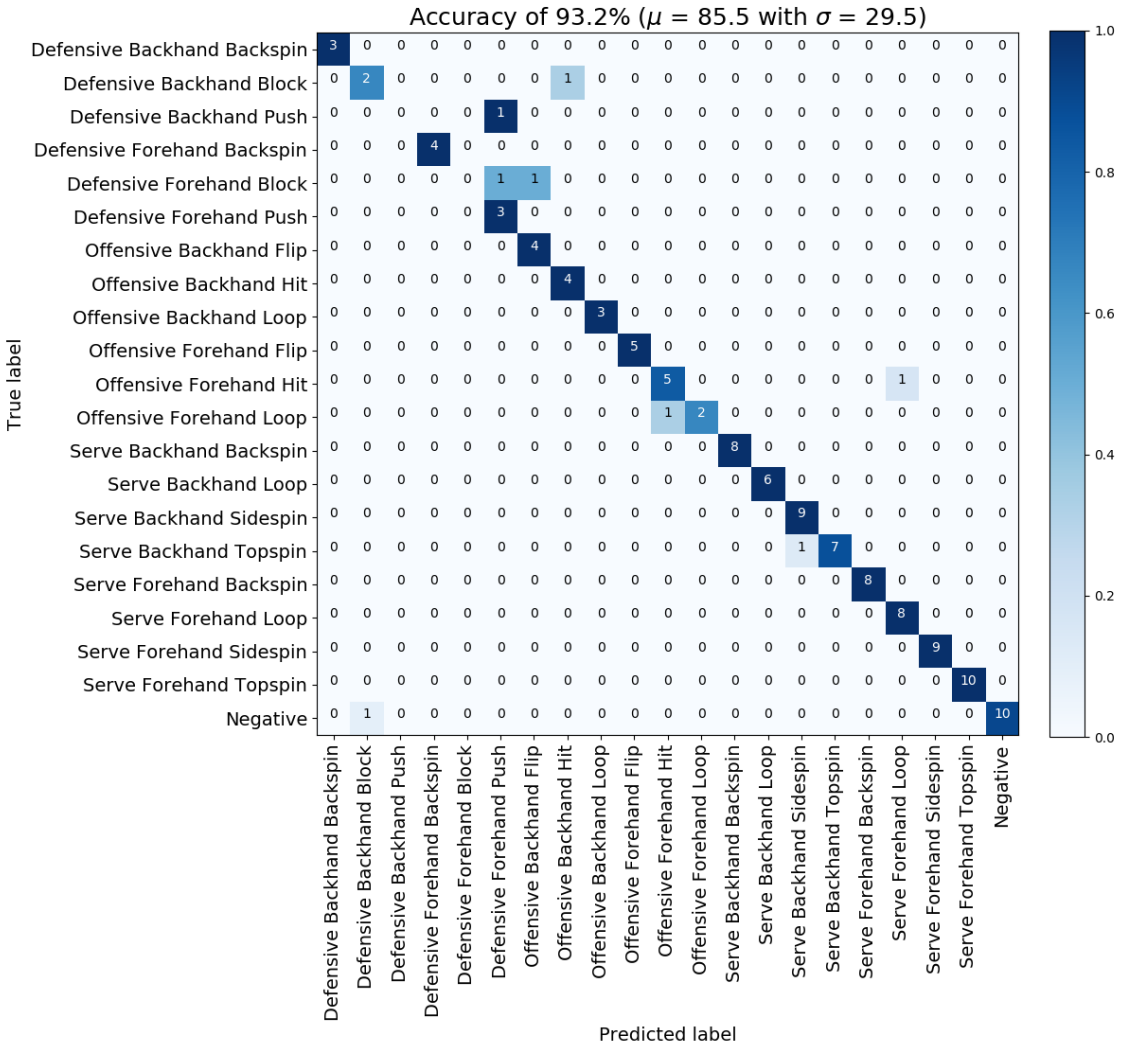


Figure 6.3 – Confusion Matrix of the T-STCNN model using "Test" method decision with $T = 100$.

than the RGB-STCNN, certainly because the flow values are not used since they are too small compared with normalized RGB values. However EF-STCNN with "NORMAL" flow normalization get much better performances with stable results when considering the temporal dimension with "TVote", "TAvg" and "TGauss" evaluation methods. Nevertheless, it seems that to get the best of the two modalities, they need to be processed in parallel. Indeed, concatening two modalities encoding different types of information, without pre-processing, may not be relevant which explains the better performances of the T-STCNN model. **We could also consider different convolution approaches to allow a smoother fusion through the model.**

Our baseline, the Two-Stream I3D model trained from scratch on `TTStroke-21` dataset, ranks fourth in the presented results. They reach an accuracy of 87.1 with the "TAvg" decision rule which is an improvement of more than 10% in accuracy compared to the "Test" rule which does not consider the whole video segment for decision making. They also get far better performances when considering 100 frames as input compared to 64 which fails to obtain accurate classification. One can notice that our T-STCNN with $T = 64$ gets much better performances than their model considering the same number of frames. The over fitting of their method on the train set is visible on the convergence of their single modality models, especially with $T = 64$, (Figure 4.3 from Chapter 4). To give a potential explanation of such overfitting, the Two-Stream I3D model being deeper than ours, it may over-fit our dataset which is more limited than UCF101 or HMDB51 datasets on which they report their results on. We also train their model from scratch with a different image resolution than for what their model was built for. This may have led to inconsistent tuning of the training parameters using the recommended learning rate and number of iterations. By looking at the confusion matrix of the Two-Stream I3D model using 100 frames in Figure 6.4, we can see that it fails to recognise well the "Defensive Forehand Block" strokes, as our T-STCNN model. This is certainly due by the low concentration of this particular stroke in the training set.

However, it classifies to much as "Defensive Backhand Push" while not classifying correctly its true label sample. This class is also not well represented in the dataset neither, and is similar to other "Backhand" strokes, explaining such behaviour. The low inter-class variability of `TTStroke-21` increases the difficulty of the task.

Two-Stream I3D network is composed of nine inception modules and four additional convolutional layers against two times three convolutional layers processed in parallel for our T-STCNN. Other experiments were conducted using deeper and shallower architecture by adding and removing 3D convolution layers in our T-STCNN. However the results were not convincing which corroborates the fact that I3D-models are too deep networks for our application and therefore comforted us to work with such network depth. In addition, our T-STCNN models, being much shallower than the I3D models, can be trained jointly with RGB and Optical flow streams; which certainly improves the following fusion step. Conversely, the Two Stream I3D model simply adds the predictions of each stream, which may explain

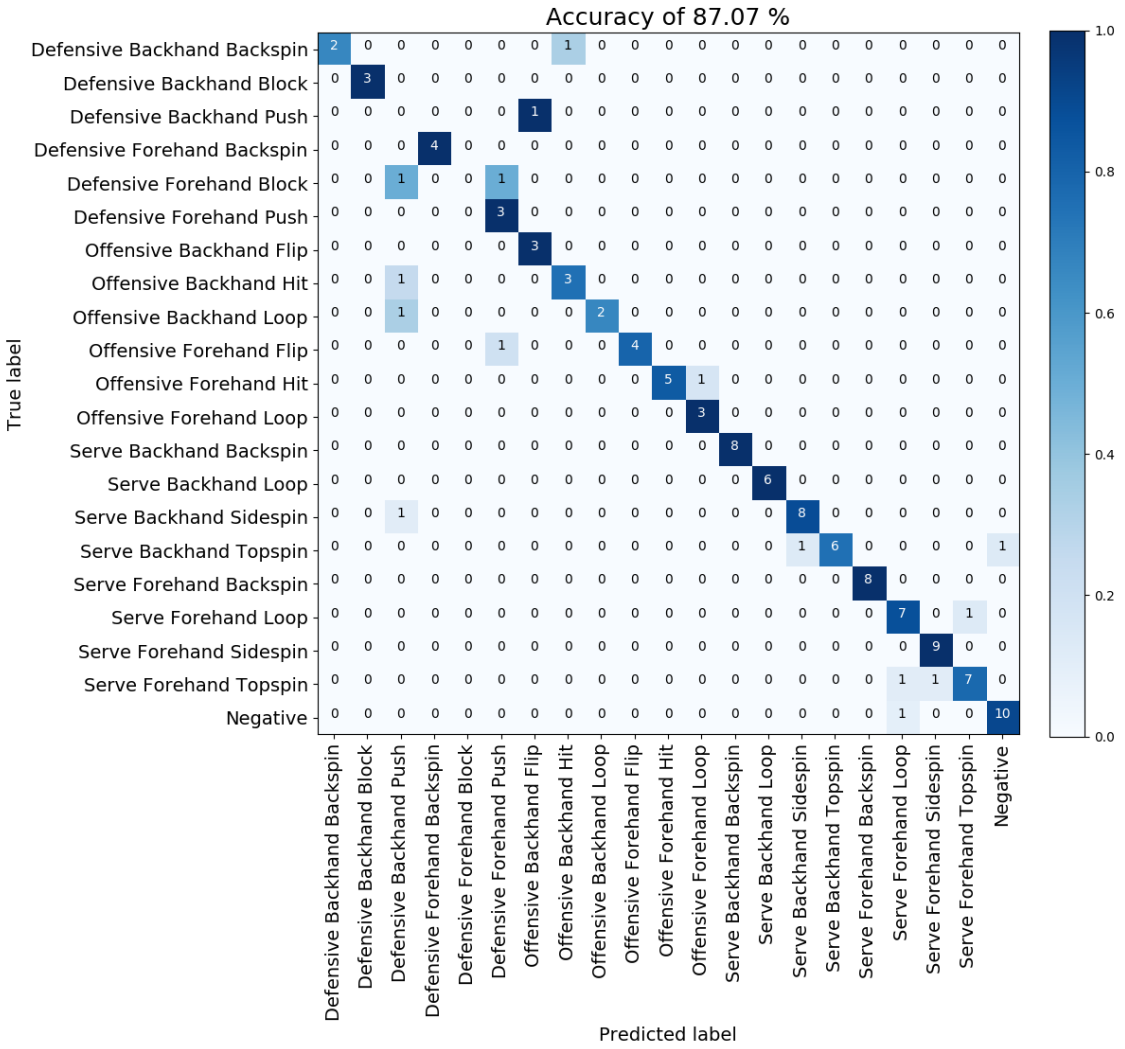


Figure 6.4 – Confusion Matrix of the Two-Stream I3D model using "TAvg" method decision with $T = 100$.

lower performances.

3.2 Joint Stroke Detection and Classification Task

Table 6.2, shows the accuracies obtained for our T-STCNN model with and without data augmentation. The table reports results with all labels, and also when not considering the negative labels. This second evaluation is motivated by the fact that most parts of a video are constituted of negative labels. Indeed, is considered as negative, all the portions between stokes: i.e when the player is getting ready, when the match or training session ends and when the player is resting. It makes the evaluation method much more discriminant since even the overlaps between annotation, which were considered correct when classified as either the previous or either the next label, are not considered.

Table 6.2 – Performance of stroke detection and classification.

	Accuracies in %			
Models	**Gross**	**Vote**	**Average**	**Gaussian**
T-STCNN*	62.8	81.8	82.3	**82.6**
T-STCNN	60.8	79.8	80.2	79.7
without taking into account negative labels				
T-STCNN*	45.4	50.2	50.8	54.8
T-STCNN	60.5	76.8	76.9	**78.4**

* trained without data augmentation

When considering all the classes, we reach a 82.6% of accuracy with the Twin model trained without data augmentation using the "Gaussian" filtering method for decision making. It is the best accuracy obtained for this task when comparing with RGB-STCNN and Flow-STCNN models. The other T-STCNN model using data augmentation is a bit behind with only 80.2% of accuracy with "Average" rule. We believe this gap can be explained by the training method used. Indeed, by looking at Table 6.1, we can notice that the accuracy on the training set for the model trained with data augmentation is lower than the one without, leaving more room for improvements. Better performance could certainly be obtained by training the T-STCNN model using data augmentation with more epochs.

We can notice, also in Table 6.2, that the performances of the T-STCNN model trained without data augmentation drops while the T-STCNN model with data augmentation stays stable (maximum accuracy of 78.4% using the "Gaussian" evaluation method). Such behaviour can be explained by the data augmentation process which robustifies the output probabilities and gives more clues to the model for distinguishing intra-stroke-similarity and inter-stroke-similarity . The T-STCNN model trained without data augmentation gets the worst score on this task compared to

the RGB-STCNN and Flow-STCNN (max scores of respectively 67.9% and 62.4% of accuracy). As seen in the previous chapter, the optical flow model, trained without augmentation, led to worse scores. In this case, the OF branch might influence the T-STCNN decision too much. On the other hand, the T-STCNN trained with data augmentation highly outperforms both the RGB-STCNN and Flow-STCNN models.

4 Conclusion and Perspectives

This chapter leverages the work of the last two chapters dedicated to fine-grained stroke action recognition in table tennis. We have proposed an approach based on a Twin spatio-temporal convolutional neural network architecture taking as input the RGB data and their estimated optical flow. The method has been evaluated on the `TTStroke-21` dataset, recorded in real-world conditions and annotated using a crowdsourced method with professionals of the table tennis field. Our method is compared with the I3D baseline, which was the the state of the art method during this work. Our T-STCNN model reaches a maximum accuracy of 93.2% on the test set against 87.1% for to the Two-Stream I3D model on the pure classification task. For the detection and classification task, our model performs best when considering all classes if it is not trained using data augmentation and reaches 82.6% of accuracy. However this evaluation is biased due to the high presence of negative segments. By discarding them, we notice a net superiority of our T-STCNN when trained with data augmentation which is able to reach 78.4% of accuracy against 67.9% and 62.4% respectively for RGB-STCNN and Flow-STCNN models. This leads us to the conclusion that RGB and OF data should be merged using a middle fusion approach when training the neural networks, and that the Twin model is a proper approach to do so.

Next chapter is dedicated to the analysis of the features extracted from the T-STCNN in order to better understand their contribution in the classification decision.

Chapter 7

Features Understanding in 3D Convolutional Neural Networks for Action Recognition in Videos

1 Introduction

Deep Convolutional Neural Networks are often used for action recognition in videos. It is our case in the present research. Predicting an action is similar to image classification for consecutive frames and makes final decision over that time frame. It becomes a challenging task when actions have low inter-class variability. Although Convolutional Neural Networks are performing impressively in different action recognition datasets such as DeepMind Kinetics Kay et al. (2017), UCF-101 Soomro et al. (2012) or AVA Gu et al. (2018), it is not quite understandable why they make the correct or incorrect classification decisions. It is often observed that these models make correct decision based on wrong reasons, such as focusing on scene background information rather than the actual actions of a subject. Our main objective is to perform fine-grained action recognition in table tennis with the aim of improving athletes' performances. From raw video of table tennis exercises, it is necessary to recognize actions properly before motion, posture and other performance indicators could be analyzed. The difficulties in indexing of such video content resides in the low inter-class variability of actions.

Our solution is in designing and deploying 3D CNNs such as presented in Varol et al. (2018); Carreira and Zisserman (2017); Wang et al. (2018b) and we show that they classify actions quite efficiently, but the interpretation of their decision making still remains open. The explanation of decisions is specifically important for the target users of Multimedia content, sport coaches in our case. Hence, this will help table tennis teachers to focus on particular strokes performed by students for post exercise analysis. It is needed to explain the decision making of the CNN, at the generalization step, as the user is interested why a particular image or video segment is assigned to a particular class by the model.

In this chapter we present a novel visual CNN features understanding technique. Its objective is to find salient features that play a key role in the decision making of the network. The technique uses only the features from the last convolutional layer

before the fully connected layers of a trained model and generates a binary feature importance map per channel. To reduce the contribution of relatively low magnitude features channels, the final importance map is generated as a weighted sum of all binary channel maps. The resulting map is propagated to the original frame thus highlighting the regions in them that contribute to the final decision. The method is fast and does not require gradient computation as many state-of-the-art methods do.

The attempts to explain the decision making by CNNs are numerous and this area of research is very active as it serves for increasing trust of the users into Deep Learning and AI in general. Hence the method Hassan et al. (2019) already focuses on explaining the decision by getting the images from the training set which contributed the most to the decision by comparing feature distances. Also, ROAR method Hooker et al. (2019) was recently introduced to measure the interpretability of the networks for image classification. The method is based on gradient computation: it actually modifies the training and test sets by removing the important features used for classification, retrain the network and analyses its performances. However such method is very expensive especially when it comes to videos. In this chapter we therefore focus on the visualization of the features and analyse their localisation when seeking for computational efficiency.

The proposed technique is applied to the T-STCNN designed for table tennis action recognition presented in the previous chapters and trained on `TTStroke-21` dataset. Features visualization is performed to the RGB and OF branches of the T-STCNN architecture to highlight contributions of pixels both in video frames and motion vectors into the final decision. Contrary to the popular visualization methods which are based on back-propagation with gradient computation, the proposed method uses only features value and global importance of features channels. The method is compared with the state-of-the-art gradient-based techniques such as Vanilla Gradient-based Back-propagation Simonyan et al. (2014), Guided Back-propagation Springenberg et al. (2015) and Grad-CAM Selvaraju et al. (2020). The method gives a better understanding of the decision and is similar to Vanilla gradient-based propagation which is the reference method in the field. The metrics show that generated maps are similar to those obtained with known Grad-CAM method and the method is faster than all our baselines.

This chapter is related to some extent to our following publications: Martin et al. (2020c, 2018); Fuad et al. (2020) and is the result of the collaboration with Kazi Ahmed Asif Fuad who did his internship in the scope of the Eramus Mundus master Image Processing and Computer Vision (IPCV) within our team. Kazi Ahmed Asif Fuad is the main author of the paper on which this chapter is built on (Fuad et al., 2020) and it gave a great extension and deeper insight to the project, which we believe deserves its place in this thesis.

The rest of the chapter is organized as follows. Section 2 discusses related works on Convolutional Neural Network features understanding techniques. In Section 3, the proposed algorithm is explained. In Section 4, results of different test cases

are compared between the proposed algorithm and existing algorithms. Finally, conclusion and discussions are drawn in Section 5.

2 Related Work

DNNs are commonly considered as black-boxes. Indeed they are composed of stacked layers of individually very simple functions whose parameters correspond to weights that are mainly depending on the training data. However, the global function that represents such a network is hardly understandable. Several works have been done to mitigate this issue by visually providing hints on the decision taken by the DNN Hohman et al. (2019).

Such visual applications often rely on a view that provides features importance in the input space (*i.e* which features positively voted for the final classification). In the variety of approaches to generate such views, we can distinguish two trends: i) methods on the basis of back-propagation and gradient computation and ii) methods based on back-tracing feature values. Another family of the methods is based on perturbations added to the input and measuring the deviations of the output thus explaining the contributions of pixels/regions in the input content (Fong and Vedaldi, 2017). In the follow-up we focus on methods by back-propagation and gradient computation as they have inspired our contribution and served as a benchmark for it.

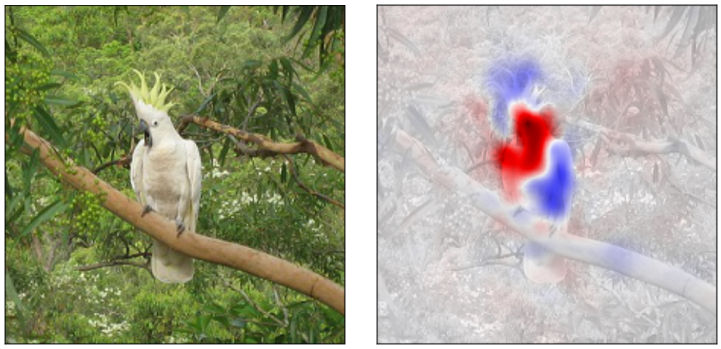
2.1 Methods Based on Back-Propagation and Gradient Computation

In the pioneering work Simonyan et al. (2014), the authors are interested in the so-called "class-saliency": what are the pixels which contribute the most into the decision of assigning an image to a particular class. A class score function $S_c(I)$ of a CNN is considered as a linear operation of convolution of an input image and a weight vector w. In reality the decision function of a CNN is highly non-linear. Hence the Taylor expansion is used and the derivatives, or gradient, of the weights with regard to the argument image I are computed. Strong derivatives of weights indicate pixels which contribute to the decision the most. This allows to build the so-called "class-saliency maps".

The Grad-Cam method Selvaraju et al. (2020) is based on the same principle. It generates a heatmap that highlights features of interest by back-propagating the gradient of the last layer until it reaches a convolution to compute the influence of the neurons on the prediction. The importance map is then upscaled to the initial image size in order to produce the heatmap.

The work carried out in Zintgraf et al. (2017) generates a heatmap that indicates in blue the input pixels that voted against the predicted class and in red those that voted for, as depicted in Figure 7.1. The method relies on difference analysis that

modifies the input space in order to detect how the prediction changes if the feature is unknown. Despite the gradient is not computed as in Simonyan et al. (2014), this is also a kind of "differential" approach.



a. Input image **b.** Output heatmap

Figure 7.1 – Example of decision visualization using prediction difference analysis (Zintgraf et al., 2017).

2.2 Methods Based on Back-Tracing Feature Values

The Guided Backpropagation uses the neuronal responses in high-level feature maps and propagates them back to the image thus finding pixels which contributed the most into the response of a single neuron Springenberg et al. (2015). Given high-level feature map, the "deconvnet" inverts the data flow of a CNN, going from neuron activation in the given layer down to an image. Typically, a single neuron is left non-zero in the high level feature map. Then the resulting reconstructed image shows the part of the input image that is most strongly activating this neuron, and hence the part that is most discriminative to it. The authors of Springenberg et al. (2015) work with fully CNNs which does not contain max pooling layers and thus the "tractability" of a neural response to the original pixel is possible.

Fully Layer-Wise Relevance Propagation (LRP) Montavon et al. (2019) also generates a heatmap of input features that supports the decision. The method relies on the concept of a relevance score per activation; the sum of all relevance scores of each layer is equal.

Li *et al.* Li et al. (2019b) generate salience relevant maps thanks to firstly LRP generated maps. This additional step allows to highlight parts of the image following human attention mechanisms by removing irrelevant parts highlighted by LRP.

VisualBackProp Bojarski et al. (2018) aims at visualizing the pixels at the origin of the decision in order to help to debug CNN in real time. The method can be used both during training and inference. The method is quite simple: the output of each ReLU layer is averaged, up-scaled to the resolution of the previous layer and multiplied by the previous layer. The operation identically repeated until the input layer.

Our proposed approach is in line of this group of methods: it relies only on the feature maps of the last convolutional layer following the philosophy that in Deep CNNs all feature layers except the FC layers are "feature extractors". Hence we only use the last one. Its features are the most relevant for the final decision. Thereon, by back-propagating the strong features, identification of the salient regions in the video frame is deduced. The method is presented in the next section.

3 Proposed Features Understanding Method

The method we propose is generic and can be applied to features understanding in networks for image classification such as 2D CNN, or as in our case to chunks of video frames in a 3D spatio-temporal convolutional neural network.

The core of the method we propose relies in the back-tracing of "strong" features from the last feature-layer, meaning the convolutional layer. From our perspective it "explains" the Network decisions at the generalization step.

At the generalization step, the chunk of input video frames to classify is forward-propagated through the trained network. Following the general philosophy of CNNs be they 2D or 3D, the convolutional layers act as features extractors and the last FC layers as classifiers. The upper convolutional layers are supposed to extract low level features Luo et al. (2016) from input data, while deeper we go into convolutional layers, higher level semantic features become prominent. Hence, we extract features from the last convolutional layer. The features are taken just before feeding the fully connected layers. The overview of our proposed algorithm is given in Figure 7.2. Here we show an illustration of the method for one video frame in RGB data without loss of generality.

The extraction of features from the last convolutional layer is realized after the activation function and max-pooling layers, as presented on the upper part of the figure. Binary feature maps are generated and importance weights are calculated. Importance map is then computed as a normalized linear combination with channel weights and visualized as a heat-map on the original image.

When our data that are chunks of video frames of size $(W \times H \times T) = (120 \times 120 \times 100)$, are pushed through the convolution and polling layers of the network, the input video frames become "feature frames". Their number is reduced by pooling in temporal dimension compared to the number of original video frames in the input chunk. By applying one filter we obtain its corresponding map. Finally, before fully connected layers we have $F = 80$ feature maps containing each $N = 12$ feature frames of size 15×15 in our case, with N being the temporal dimension. This size results from the three successive convolutional layers, using stride one, padding one in all directions with output dimension same as input dimension; which are connected to a max pooling layer using a stride of two in all directions and using floor function. A feature map can be considered as a "channel" of the feature maps. The total feature maps size is therefore (80×15). The proposed method is applied

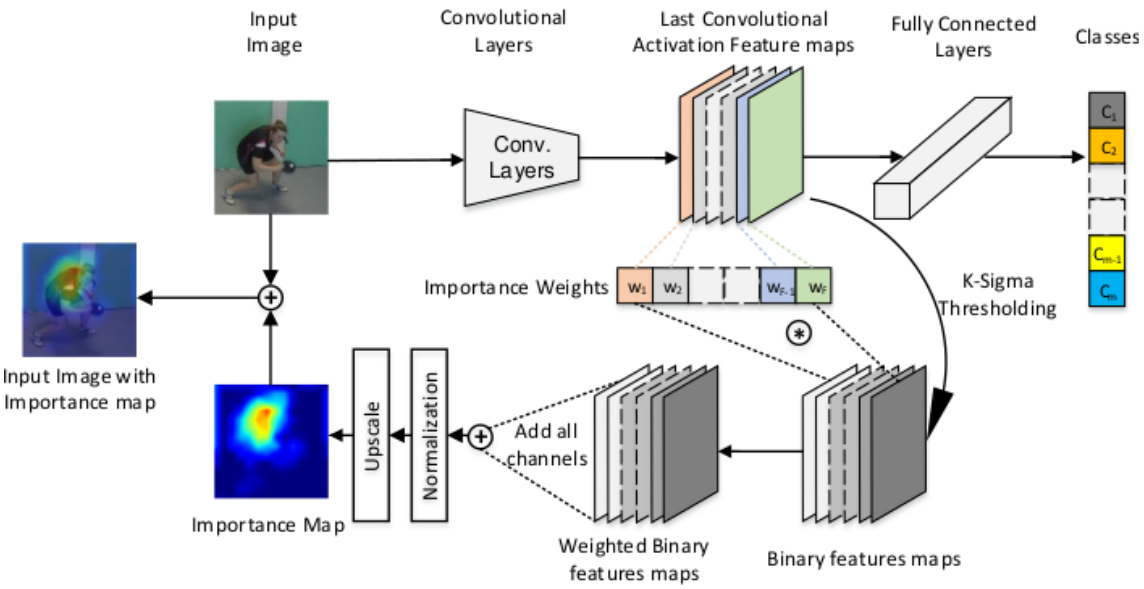


Figure 7.2 – Overview of the proposed visualization method.

to each feature frame and the resulting N importance maps M'_n with n ranging from one to N are back-projected on all input frames by tri-linear interpolation in space and time Kenwright (2015).

The second step generates a binary map for each channel of this features map in order to give an importance value for each features channel-by-channel. To detect the strongest features, we suppose that the features values in features maps follow Gaussian distributions. Obviously, the mean is positive as we take the features after a commonly used ReLu non-linearity that transforms negative values to zero, as depicted in equation 7.1.

$$ReLu(x) = max(0, x) \tag{7.1}$$

In the last convolutional layers features are "expressive" and only few of them are important. Following the Gaussian distribution hypothesis we are interested in the right queue of the distribution corresponding to "rare" and strong features. Hence we threshold the features maps accordingly to the K-sigma rule with different K values tested. For each channel c, the mean μ_c and standard deviation σ_c are calculated. Then a binary map per channel $\mathcal{B}_c$ is built which marks the strong features as described in equation 7.2:

$$\mathcal{B}_c(x_{i,c}) = \begin{cases} 1 & \text{if } x_{i,c} \geq \mu_c + K * \sigma_c \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

with $x_{i,c}$ the i^{th} feature value from the c^{th} feature map from last convolutional layer of the model. In our case i varies in the coordinate space $12 \times 15 \times 15$ which represents the feature map size and c in the interval $[1, F]$ representing one feature map.

During our experiments, we have verified that the histogram of each channel c shows that features distribution is similar to normal distribution although negative values are removed by the ReLU activation. Hence after thresholding, in each channel we have marked the strongest features through binary features maps as described in the bottom part of the Figure 7.2.

Next, not all features channels are globally significant for decision making. The number of convolutional filters in each layer is often chosen on the basis of preliminary experiments, full optimization of the network hyper parameters being too heavy. We propose to weight each temporal frame of each feature map using their mean value. The importance weight W of size F represented in Figure 7.2 , is it self composed of weight vectors $w_c = [\mu_1^c, ..., \mu_N^c]$ of size N, μ_i^c being the mean value of the temporal frame of the feature map. In our case, this importance weight vector is composed a $F = 80$ vectors of size $N = 12$.

Then, we compute the importance map M as a linear combination of all channel binary maps $\mathcal{B}_c$ using the channel weights μ_c and normalize it to $[0; 1]$ by using "Min-Max" feature scaling described in equation 7.3.

$$M' = \frac{M - min(M)}{max(M) - min(M)} \tag{7.3}$$

with, in our case, $min(M) > 0$ due to ReLU function and binarization using K-sigma rule. However this might change accordingly to the non-linear activation function chosen for the neural responses in the network.

Finally, the normalized importance map M' is up-scaled to the original image/video frame dimension ($W \times H \times T$) by a linear interpolation. Furthermore, the importance map M' is superimposed on the original image/video as a heat-map to visualize the spatial and temporal information which has contributed the most to the decision making.

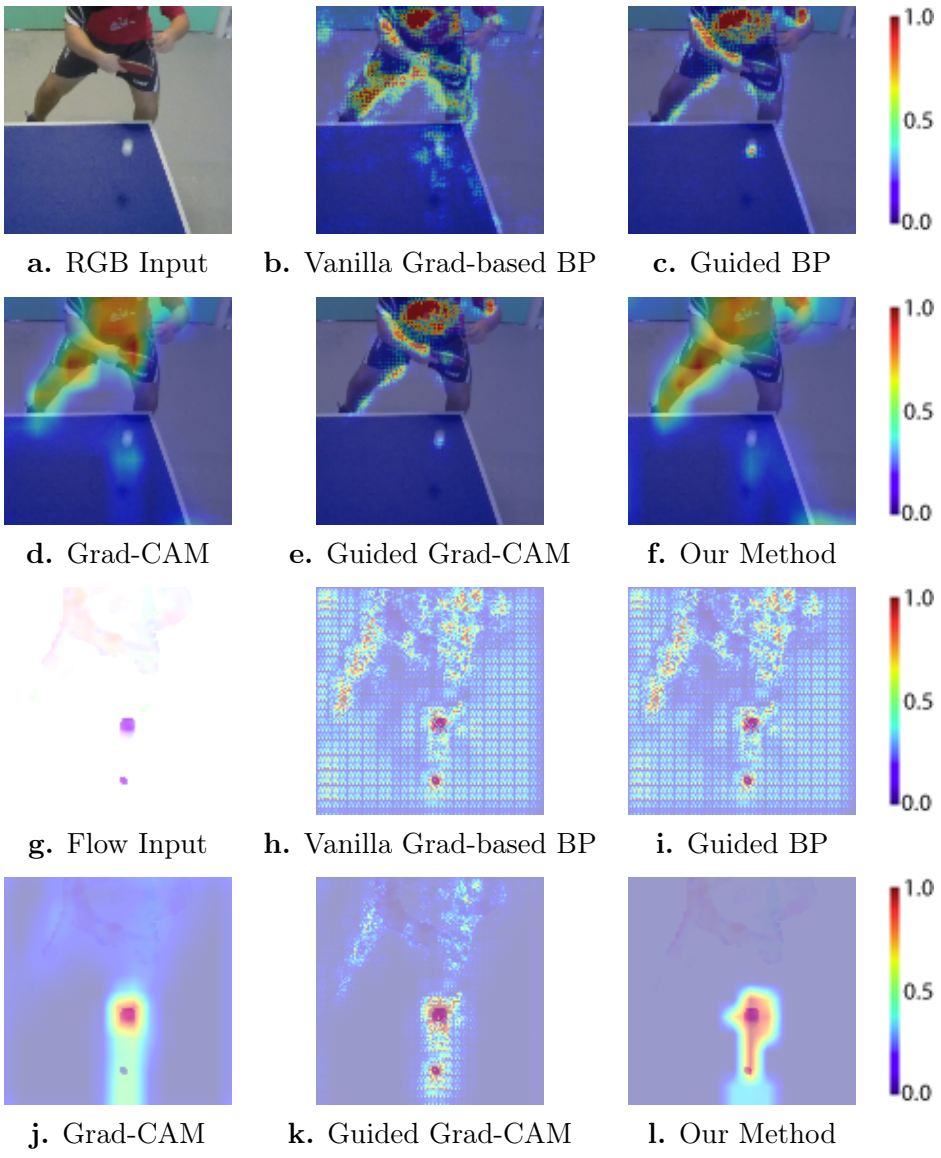
4 Experiments and Results

All experiments have been conducted using the T-STCNN model trained on `TTStroke-21` presented in Chapter 6. Our visualization results are visually compared with common methods in Section 4.1. Then comparison with our baselines is done with different metrics in Section 4.2. Finally computation times of all the methods are compared in Section 4.3.

4.1 Visual Analysis

Visualization was initiated with PyTorch Code from Ozbulak (2019). The authors of the codes for different visualization algorithms considered only Single Branch Convolutional Neural Networks in 2D. We extended it for Multi Branches Neural Networks and for application to 3D-CNN.

Figures 7.3 and 7.4 show visual results of our algorithm on the two branches of the T-STCNN and compare them with classical methods: Vanilla Grad-based BP Simonyan et al. (2014), Guided BP Springenberg et al. (2015), Grad-CAM and Guided Grad-CAM Selvaraju et al. (2020). Even if the data processed are in 3D, we only show one frame from the 100 frames of the video input for better visualization. The video results are available online[1].



a. RGB Input **b.** Vanilla Grad-based BP **c.** Guided BP

d. Grad-CAM **e.** Guided Grad-CAM **f.** Our Method

g. Flow Input **h.** Vanilla Grad-based BP **i.** Guided BP

j. Grad-CAM **k.** Guided Grad-CAM **l.** Our Method

Figure 7.3 – Different visualization algorithm outputs of the T-STCNN model for the class: "Defensive Backhand block". First two rows show the visualization for RGB input data and third and last row for Flow input data.

To keep the comparison uniform in the Figures 7.3, 7.4 and 7.5, all the impor-

[1] https://youtu.be/2yrG4vKxRTA

tance maps are scaled between zero and one using "Min-Max" feature scaling, see equation 7.3, and are visualized as heatmap over the input frame using *"jet"* color scale represented on the right of the figures. In Figures 7.3 and 7.4 the OF values are visualized by converting $V = (v_x, v_y)^T$ into an image in the color domain HSL where the Hue represents the angle created by v_x and v_y, the Saturation is set to one and the Lightness represents the amplitude of the motion.

The Figure 7.3 shows the table tennis stroke "Defensive Backhand block" with different visualization algorithms for both RGB and OF data input. From visual observation, we can see that Vanilla Gradient-Based Back-propagation Simonyan et al. (2014), Guided Back-propagation Springenberg et al. (2015) and Guided Grad-CAM Selvaraju et al. (2020) visualizations suffer from "discretization effect". On the other hand, continuous and smooth visualization has been obtained in Grad-CAM Selvaraju et al. (2020) and our method. From the RGB data and its visualizations in Figure 7.3, subfigures a to f, we can notice that Vanilla Gradient-Based Back-propagation focuses on all over the body and the table, but Guided approaches focus mostly on the upper body and the table tennis ball. In contrast, Grad-CAM and our method focus on the left leg and the hands which is coherent with human interpretation: they are the most important regions to classify a table tennis stroke. Regarding OF data, subfigures 7.3.g to l, our algorithm highlights the ball mainly. It makes sense since its OF values are very high and are characteristic to stroke presence. Also we select only the most prominent and "rare" features in the last convolutional layer, see Section 3, which do not leave much room for other features. The other methods behave similarly but Vanilla Gradient-Based and Guided BP give quite a noisy picture of "important" motion vectors on the body.

Figure 7.4 illustrates the features of a miss-classified sample. In this sample, for RGB data, Vanilla Gradient-Based Back-propagation Simonyan et al. (2014), Guided Back-propagation Springenberg et al. (2015) and Grad-CAM Selvaraju et al. (2020) algorithms highlight both the body and the table. But our algorithm emphasizes the body and the moving hand while Guided Grad-CAM Selvaraju et al. (2020) focuses on the side of the body. For OF data, all the algorithms focus on the moving parts mostly but Vanilla Gradient-Based Back-propagation Simonyan et al. (2014), Guided Back-propagation Springenberg et al. (2015) take the whole body into consideration. Also, all models are focusing in general on foreground rather than background, except for the little piece of wall in the upper right corner of the RGB image. Our method based on features computation from the model is focusing still on the player, even when no stroke is performed, therefore the model keeps focusing on the hand, which is a good sign for stroke recognition. However, we can suppose that, because the hand is moving to reach the ball on the table, the model falsely recognizes the sample to contain a stroke. This could be avoided by considering more such negative samples in the training process of the T-STCNN. Still, this negative sample is challenging because in most table tennis context, no box, containing the table tennis ball, should be on the table and in such view of the camera.

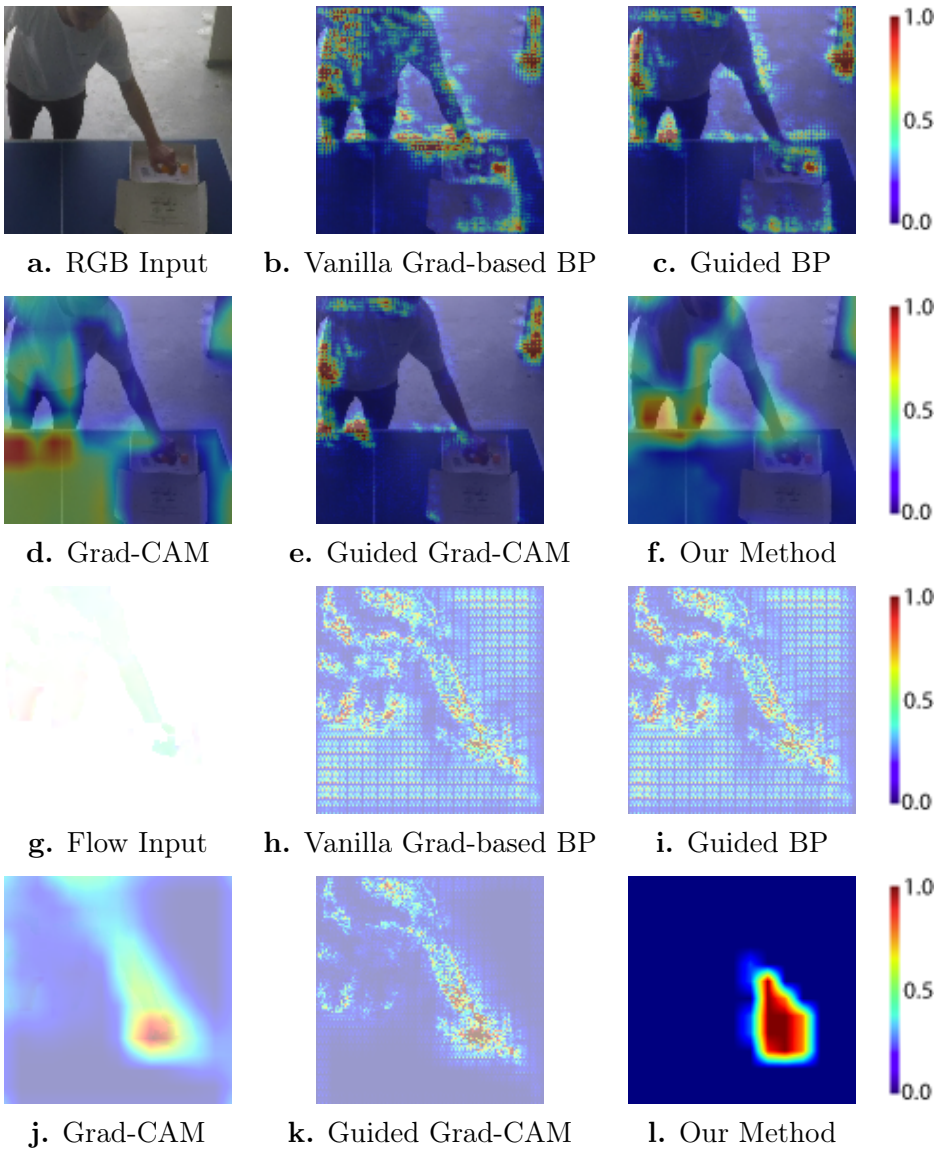


Figure 7.4 – Different visualization algorithm outputs of the T-STCNN model for the "Negative' class. First two rows show the visualization for RGB input data and third and last row for Flow input data.

We also conduct experiments to check if the choice of the features from the last convolutional layer is justified as we discussed in Section 3. The Figure 7.5 illustrates a typical visualization of the features extracted from the different convolutional layers output of our T-STCNN from the RGB branch, on the same "Offensive Backhand block" that in Figure 7.3, and using our method with different K values. Here We also use the popular 2-sigma and 3-sigma rules supposing Gaussian distribution of features, i.e. $K = 2$ and $K = 3$. The choice of k-value influences the binary mask that we computed before weighting as described in equation7.2.

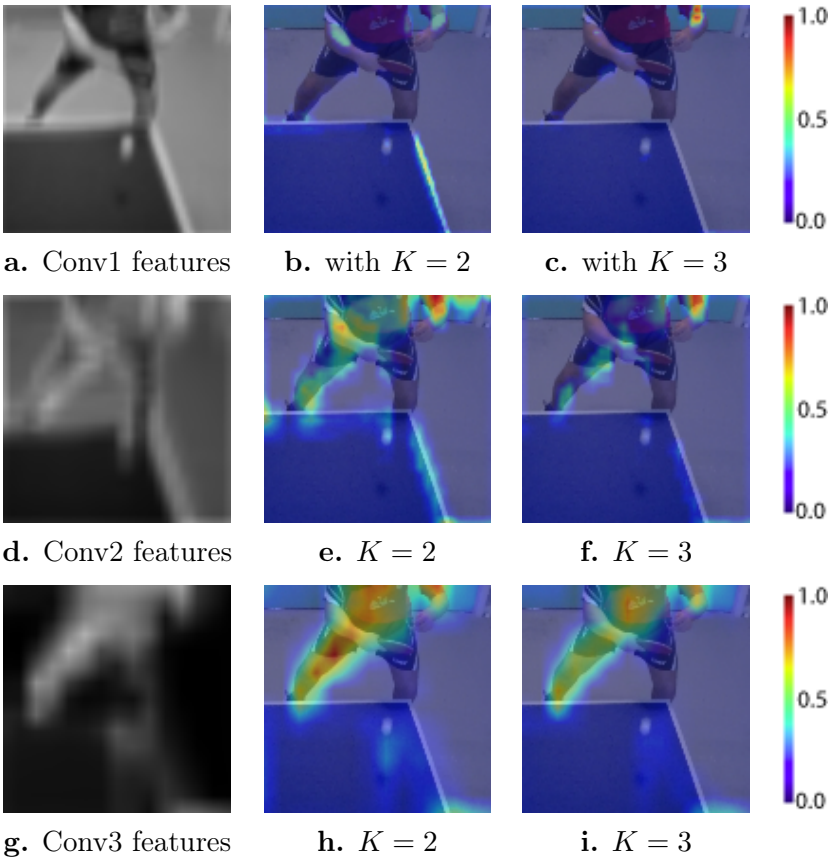


Figure 7.5 – Features visualization from different convolutional layers with different K values.

From Figure 7.5, we notice that the features from the third convolutional layer, third row, are the most coherent with the motion of the person and the ball position; and that the 3-sigma rule is more interesting as it allows to filter-out contrasted features on static objects and concentrates on changing details captured thanks to the 3D convolutions. The objective is visual evaluation of explanations which requires a large user study and discussion with professionals of the field which is in the perspective of our work. Nevertheless, we can quantify the similarity of resulting visualization maps obtained by our method with our baseline methods.

4.2 Metric-Based Comparison of the Methods

To compare our method with the baselines, we use bench-marking strategy using metrics for visual attention prediction. Several comparison metrics for normalized predicted heat-maps are used to compare with the reference map. Here we will compute the so-called "Similarity" and the "Pearson correlation coefficient" metrics Bylinskii et al. (2019). We perform the comparison on the normalized importance map M' re-scaled to the resolution of input frames as described in Section 3.

Similarity Metric

The similarity Bylinskii et al. (2019), is a popular metric to perform a simple comparison between importance maps. The importance maps are considered as distributions and the metric measures the intersection between two distributions. Given two importance maps M_1 and M_2 where independently the sum of their values equal to one, similarity metric is:

$$Similarity(M_1, M_2) = \sum_i min(M_{1i}, M_{2i}) \tag{7.4}$$

where, if iterating over the discrete pixel i, we have $\sum_i M_{1i} = \sum_i M_{2i} = 1$. Two completely overlapping importance maps will result in maximal similarity of one and the similarity will be zero when there is no overlapping at all. As different visualization techniques are being compared, a metric was required for partial matches and similarity metric is adapted for their assessment Bylinskii et al. (2019).

Pearson's Correlation Coefficient Metric Pearson Correlation Coefficient (PCC) Bylinskii et al. (2019); Judd et al. (2009) measures how two maps are correlated or depend on each other: it is close to one when two variables are perfectly correlated and zero when they are not at all. For two importance maps M_1 and M_2, PCC is:

$$PCC(M_1, M_2) = \frac{Cov(M_1, M_2)}{\sigma_{M_1}\sigma_{M_2}} \tag{7.5}$$

where $Cov(M_1, M_2)$ is the covariance of M_1 and M_2.

There has been many sanity checks done on visualization algorithms. In Adebayo et al. (2018), they suggest Vanilla Gradient-Based Back-propagation is more effective than other visualization algorithms. Hence we take the importance maps obtained with Vanilla Gradient-Based Back-propagation as reference for comparison. In Table 7.1, complete metric evaluation on the test set of `TTStroke-21` dataset is provided. The test set is composed of 116 instances over 21 classes. We calculated both mean and standard deviation to observe the deviation for different instances. In our observation on the test cases, similarity and PCC metrics are coherent for all the samples and algorithms for both RGB and OF data.

Table 7.1 – Comparison of Vanilla Gradient-based Back-propagation (VaGrBp) and of our method (Ours) with Guided Back-propagation (GuBp), Grad-CAM (GrC) and Guided Grad-CAM (GuGrC).

	RGB		Flow	
Methods	Similarity	PCC	Similarity	PCC
GuBp vs VaGrBp	**.77±.01**	**.75±.02**	**.99±.01**	**.99±.01**
GrC vs VaGrBp	.69 ± .04	.61 ± .08	.66 ± .02	.54 ± .06
GuGrC vs VaGrBp	.73 ± .02	.70 ± .03	.79 ± .02	.80 ± .03
Ours vs VaGrBp	.70 ± .03	.63 ± .05	.70 ± .01	.61 ± .02
Ours vs GuBp	.70 ± .03	.64 ± .05	.70 ± .01	.61 ± .02
Ours vs GrC	.70 ± .05	.63 ± .10	.72 ± .06	.69 ± .12
Ours vs GuGrC	.68 ± .03	.63 ± .05	.77 ± .02	.71 ± .02

Comparing all the state-of-the-art methods between them, we notice that Vanilla Gradient-based Back-propagation and Guided Back-propagation have highest similarity and PCC. These results are obvious as both algorithms rely on almost the same principle of using first convolutional layer gradients except on how they treat the gradients. Vanilla plots both positive and negative gradients whereas Guided Back-propagation plots only positive gradients. For Guided Grad-CAM, Grad-CAM output is multiplied with Guided Back-propagation output. Hence, Vanilla Gradient-based Back-propagation has higher Similarity and PCC with Guided Grad-CAM. In contrast to Vanilla Gradient-based Back-propagation Simonyan et al. (2014) and Guided Back-propagation, Grad-CAM uses last convolutional or deep convolutional layer gradients. Grad-CAM has similarity and PCC 69% and 61% respectively for RGB data but for OF data, it has similarity 66% and CC 54% which is nearly 30% and 40% less than compared to Guided Back Propagation.

Comparing our algorithm with state-of-the-art methods, we get 70% of similarity and 63% of PCC with Vanilla Gradient-Based propagation and Guided Back-propagation for RGB data. Compared to Grad-CAM, our algorithm has 5% of similarity and 15% of PCC more with respect to Vanilla Gradient-Based Visualization for OF data. With Grad-CAM, our algorithm yields the most similar balanced results: 70% similarity for RGB and and 72% for Optical Flow data. Finally, with Guided Grad-Cam, our algorithm attains slightly higher metric values: 77% of similarity and 71% of PCC on OF data. This can be explained by the use of the same features which come from the last convolutional layer.

4.3 Computational Analysis

All the algorithms have been developed using Python based PyTorch and Numpy libraries. Similar functions were used for calculating closely related functionality of different algorithms to make computational analysis uniform. Average time for each

instance visualization of different algorithms is given in Table 7.2.

Table 7.2 – Computation time for the different visualization techniques.

Visualization techniques	Time in second
Vanilla Back-propagation Simonyan et al. (2014)	$5.12 \pm .23$
Guided Back-propagation Springenberg et al. (2015)	$7.75 \pm .43$
Grad-CAM Selvaraju et al. (2020)	$4.9 \pm .21$
Guided Grad-CAM Selvaraju et al. (2020)	$11.37 \pm .62$
Proposed method*	**2.91 ± .13**

Training of model was performed on GPU but visualization was performed on CPU only. Computation time was measured in Intel(R) Xeon(R) Gold 5118 CPU @2.3GHz and Intel(R) Core(TM) i9 9900 CPU @3.1GHz. In both cases, similar results and ranking were obtained. In Table 7.2, computation time is provided only for Intel(R) Xeon(R) Gold 5118 CPU @2.3GHz. Average computation time was calculated on the 118 samples of the test set, each one having 100 frames. The OF data were computed beforehand. From Table 7.2, it is clear that our algorithm is the fastest among all the visualizations method since it does not contain a time-consuming gradient back-propagation. Our algorithm is almost two times faster than Vanilla Gradient-Based visualization and Grad-CAM.

5 Conclusion

In this chapter we have proposed a new method for explanation of CNN decisions by interpretation of visual features of CNNs in classification tasks. The method is generic and applicable both to 2D CNNs, typically used for image classification, and 3D CNNs for video action recognition. The method is based on selection of important features from last convolutional layer, the use of channels importance and back-projection of the feature importance maps to the original input.

We have shown that the method gives comprehensive results both on RGB input and on optical flow in our T-STCNN dedicated to table tennis stroke classification using `TTStroke-21` dataset. We have analyzed the extracted features using this method. It gave us some insights about the decision making. Our model manages to focus on the proper part of the video frames as experts in the field are analysing in the stroke classification: ball, player position, body parts and racket. Our T-STCNN however fails when it comes to situations that the model is not used to such as the box containing the ball on the table.

We also compared our method to the known features visualization methods with the help of classical similarity metrics used for saliency/importance maps. The method gives very much similar results in terms of Similarity and Pearson Correlation coefficient with regard to the Vanilla Gradient Back-propagation, is the

most similar to Grad-CAM remaining faster than all considered base-line methods. Evaluation of the methods would be more accurate if gaze fixation density maps of observers performing video action recognition tasks were available. Also, recent work Tomsett et al. (2020) developed metrics for saliency images and it would be of great interest to apply it to videos.

This work motivated us to go deeper into the attention mechanism to help the model focusing on the meaningful part of our inputs. Next chapter is dedicated to such work, in which we introduce 3D attention blocks within the T-STCNN architecture in order to increase its performances.

Part III

Extension of Architectures for Action Recognition

Abstract

In this third and last part, the previous models are modified in order to improve the classification performances. These modifications are performed by incorporating attention mechanisms in the models architecture. The attention mechanism is achieved through attention layers. The attention blocks are inspired from the 2D existing attention blocks and are extrapolated to the temporal dimension. The features of the attention blocks are analyzed to assess their efficiency. The convergence of the models using attention mechanism are discussed. Obtained results are compared with the I3D baseline and previous performances.

Keywords

Action classification, Attention mechanism, Residual block, Deep learning, Batch normalization

Summary

Chapter 8

3D Attention Mechanism for Fine-Grained Action Classification

1 Introduction

In this chapter we introduce 3D attention modules in the T-STCNN architecture and examine their impact on classification efficiency. The use of attention blocks in the network speeds up the training step and improves the classification scores of considered models. We visualize the impact on the attention-based features and notice correlation with player movements and spatial position. Score comparison between state-of-the-art action classification methods and proposed approach using attentional blocks is performed on the T-STCNN corpus. Proposed model with attention blocks outperforms our baseline and most of the previous model described in the last chapters.

Recognition of similar actions belongs to the fine-grained classification problem, and is a current issue. In sport for instance, such as table tennis or gymnastics (Shao et al., 2020), exercises are filmed in the same environment and movements can be quite similar. Hence the recognition problem becomes harder: the classifier cannot be helped by background information where the action is performed. To be efficient, a classifier has to focus on meaningful regions and changes in the video. This is a subject of a recent trend in Deep Learning, which is the introduction of "attention mechanisms", coming originally from Natural Language Processing (NLP) (Vaswani et al., 2017). The latter are designed to reinforce the contribution of meaningful features and channels into the decision and thus to increase the target accuracy. Recently, we proposed a comparative study of these attention mechanisms inherent to convolutional networks, as described by Obeso et al. (2019). The selection of the most relevant characteristics in different layers is very similar to the human attention mechanisms measured in psycho-visual experiments as explained in Chapter 7. While these attention mechanisms in 2D networks have been intensively studied (Wang et al., 2017a), this question remains to be further explored for a spatio-temporal content analysis using 3D CNNs.

Attention mechanisms for action recognition have been recently introduced in LSTM Liu et al. (2017) in an approach based on the analysis of joints of a human skeleton. In 3D CNNs, both global channel attention and spatial attention maps for

different feature layers have been proposed Cai and Hu (2020). We also follow this trend and design attentional blocks for our T-STCNN model. In this chapter, we propose spatio-temporal attention mechanisms in 3D convolution networks. They are applied for the recognition of challenging similar actions that are table tennis strokes. This chapter is related to some extent to our following publications: Martin et al. (2021b, 2020c, 2021a, 2020b).

The rest of the chapter is organized as follows: in Section 2, works using attention mechanisms are presented. The Section 3 presents the proposed method with attentional mechanisms and details the attention block. The results are drawn in Section 4 through feature analysis and classification performances. The conclusion and perspectives are given in Section 5.

2 State of the Art on Attention Mechanisms

Attention mechanism can be assimilated to saliency: the region where a person focus to perform a certain task. This saliency can be used for complementary information for classification methods. For example, Wang et al. (2017c) use attention from gaze fixation as spatial segmentation for food classification from images on the UPMC Food dataset (Wang et al., 2015) (based itself on ETHZFood101 dataset (Bossard et al., 2014)). Similarly, González-Díaz et al. (2019) use the same modality for grasping and object recognition. Tang et al. (2018) too focus on saliency prediction from images using U-net architecture. Moreover, the importance of color information is proven by Hamel et al. (2016) when it comes to saliency, suggesting that attention should be fed with color channels for improving efficiency. Note that perception of color can also differ from people to people for various reason (Iriguchi et al., 2018), which can bring inconsistency in ground truth saliency.

In this section, we present a brief state of the art on attention mechanisms introduced in convolutional neural networks for the classification of images and videos. One can distinguish two classes: 2D attention models, which concern images, and 3D models (2D +T) concerning videos. Although such a separation may seem artificial as the same principles govern the design of the models in both cases, we prefer to treat the spatio-temporal content separately.

2.1 2D Attention Models

One of the pioneering works introducing the use of an attention model in neural networks for image classification is presented in Hu et al. (2020). The authors are interested in the contribution of feature channels along convolutional layers into decision making. The attention model here is "global": a channel weighting mechanism is introduced by "attention blocks". The processing consists of three steps: i) synthesis (*squeeze*), ii) excitation (*excitation*) and iii) feature scaling (*scale*). A block thus is, for each channel, a small network of neurons that learns a weighting

coefficient. The next layers of the network ingest the characteristic channels thus weighted. This global weighting has been used as a basis for the authors of Chen et al. (2018d) who propose "double attention" blocks, i.e. ensuring a global and spatial weighting of the characteristics in convolution network layers.

The authors of Wang et al. (2017a); Dhingra and Kunz (2019) use the principles of residual neural networks to propose "residual" learning of the attention masks incorporated in the convolution layers. Their experiments on CIFAR data bases Krizhevsky (2009) show that on CIFAR-10 the residual attention network with depth of 452 has the best error rate compared to all the basic residual networks (3.90%). The authors propose the incorporation of attention mechanisms in both forward (*forward*) and backward (*backward*) runs. This is also the approach we had in Obeso et al. (2019), but by selecting important characteristics and not by weighting features and channels. Note that when minimizing the objective function by gradient descent, the attention mechanisms are implicitly introduced via the derivative calculation where the weighted characteristics are used. The authors of Wang et al. (2017a) report that this use in back propagation makes the training data robust to noise. This is also our approach in this chapter.

Other works such as Zagoruyko and Komodakis (2017) propose "Teacher-student" networks where the "Teacher" network is the one that learns attention and guides the student network for the image classification task. In our approach we also use a kind of attention transfer as in our architecture the attention branch and trunck branch will join together for selection of important features. In Huang et al. (2018), attention mechanism is coupled with LSTM to learn the correlation between different data modalities such as text and image and therefore leads to better embedding.

2.2 3D Attention Models

We focus here on the contribution of "3D" (2D+T) spatio-temporal attention models in deep networks for the action recognition problem.

Attention mechanism is used in Liu et al. (2017) on joint skeleton and coupled with LSTM for 3D action recognition task. They report better accuracy with attention mechanism than without. They also propose a recurrent attention mechanism on their model which strengthens the attention effect but might not lead to better performances if iterated too many times. Lei et al. (2019) consider attention on the channels of aggregated temporal features extracted from videos on appearance and motion streams. Similarly, Du et al. (2018) construct a spatial attention model for each image by introducing feature pyramids. The temporal extension is obtained by a simple aggregation of the attention maps estimated for each of the K images of the pyramid extended to the spatio-temporal domain. Motion information is not taken into account. We differ from this approach by introducing attention blocks in our twin network at the level of the two branches: RGB and the OF.

In Zhao and Snoek (2019), motion information, via the optical flow, acts as the attention map for locating actions in the video. The authors introduce the "motion

condition" layer to train the network on RGB appearance components conditional to this optical flow based map. The motion weighting layer allows to modify the spatial characteristics in the convolution layers. We find here the philosophy of using motion as an indicator of areas of interest Manerba et al. (2008). Once more, the difference of our approach consists in introducing attention blocks in the two branches (RGB and OF) of the T-STCNN.

In Long et al. (2018), attention clusters over the temporal dimension are used on image features extracted using Inception-ResNet-v2 Szegedy et al. (2016) for RGB and Flow modalities. The Inception-ResNet-v2 models are pretrained on ImagetNet Russakovsky et al. (2015) and are fine-tuned for the optical flow model. A third branch processes the audio signal using VGG-16 Simonyan and Zisserman (2015) on extracted spectrogram samples and is processed similarly to an image Cheng et al. (2016).

3D attention blocks have also recently been introduced in 3D ResNet type networks for the recognition of 3D hand gestures from videos Dhingra and Kunz (2019) or from action recognition dataset Cai and Hu (2020) such as HMDB-51 Kuehne et al. (2011), UCF-101 Soomro et al. (2012) and Kinetics Kay et al. (2017). The authors of Dhingra and Kunz (2019) build on the work of Wang et al. (2017a), and propose a convolution network using the RGB image for feature extraction, and another coupled network to determine a soft attention mask with the derivable Sigmoid function. The values of the extracted mask are then combined with the characteristics extracted from the RGB array. As in Du et al. (2018), the authors do not use the motion information explicitly.

Also Temporary-linked Multi-input Attention model (TMA) is introduced in Bolaños et al. (2018). The method is based on feature extractor using CNN with a BLSTM Graves et al. (2013) to capture the temporal relationships. Their method is applied to egocentric video description and outperformed the classical encoder-decoder methods.

Very recently, in Kalfaoglu et al. (2020) BERT are introduced. The use of this new type of layer in the popular 3D CNNs as I3D Carreira and Zisserman (2017), RentNeXt Xie et al. (2017), SlowFast Feichtenhofer et al. (2019) and R(2+1)D Tran et al. (2018) improve their performances. They reach the state of the art for HMDB51 and UCF101 dataset using R(2+1)D architecture with BERT layers after the 3D convolutions layers using pretraining on IG65M Ghadiyaram et al. (2019a).

Hence, our approach differs from current methods in the literature in the following:

- we introduce the 3D attention blocks into the two video streams: the branch containing the spatial information (RGB) and the branch containing the temporal information (OF).

- movement (OF) plays the discriminating role in our fine-grained classification context, our problem being to recognize actions and not only to locate them.

3 3D Attention Mechanism in Twin Space-Time Networks

We first present our Twin Spatio-Temporal Convolutional Neural Network used for classification, introduced in Chapter 6, using attention mechanism, and then detail the 3D attention and residual blocks and tests performed.

3.1 The Twin Spatio-Temporal Convolutional Neural Network

In order to perform action classification in videos, we use a two stream convolutional neural network (*twin*) with attention mechanism. Its architecture without attention blocks and results on `TTStroke-21` are described in Chapter 6. Its architecture with attention mechanism is presented in Figure 8.1. The difference from other Two stream networks Simonyan and Zisserman (2014); Feichtenhofer et al. (2016); Chen et al. (2018c) lies in: i) the symmetries of our network, ii) the input 4D data type (horizontal, vertical, temporal and channel) and iii) the final fusion step with a bilinear layer at the end of our two branches.

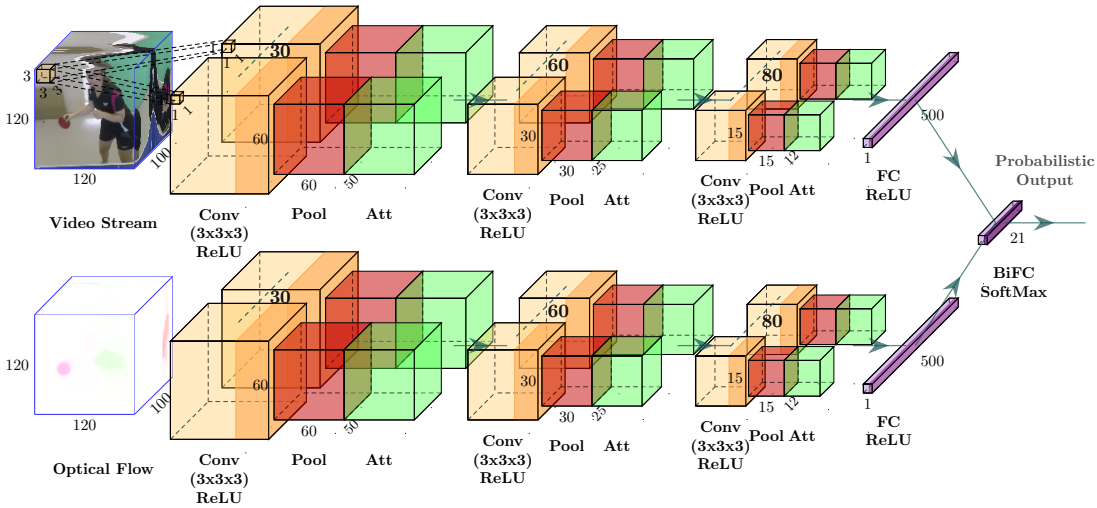


Figure 8.1 – Twin Spatio-Temporal Convolutional Neural Network with attention mechanism. The number of filters for each convolution are indicated above them.

Our T-STCNN with attention mechanism consists of two individual branches: one branch takes as input the values of the RGB images of the sequence, the other branch uses the optical flow estimated by the method of Liu (2009). It thus allows to incorporate both spatial and temporal features. The played stroke is predicted from the RGB images of the sequence and the estimated motion vectors $\mathbf{v} = (v_x, v_y)^T$.

Each branch consists of three convolutional layers comprising successively 30, 60 and 80 3D filters, followed by a fully connected layer of size 500. The 3D convolutional layers use space-time filters of size $3 \times 3 \times 3$. The two branches are merged through a final bilinear fully connected layer of size 21, followed by a Softmax function to obtain an output class membership probability.

Learning Phase

Learning of our `T-STCNN` network is done by SGD with Nesterov momentum Sutskever et al. (2013). In order to avoid overfitting, data augmentation is performed in the spatial domain using rotations, homotheties and scale transformations. Data augmentation is also performed in the time domain in order to add variability around the temporal boundaries of the played stroke. RGB data are normalized by 255 while the OF are computed using BP estimator and normalized using the "NORMAL" normalization. Transformations are discussed in details for each modality in Chapters 4 and 5.

T-STCNN with Residual and Attention Mechanisms

To test the efficiency of residual and attention blocks, an ablation study was performed: we first added residual block after the max pooling layers starting from the first max pooling layer until reaching all the max pooling layers. We did the same with attention blocks so to see the impact of each type of blocks and the impact of their number in the network. To analyse the contribution on each stream, we also experimented using separated branches of the network and with separate training, RGB branch denoted as RGB-STCNN and Flow branch denoted as Flow-STCNN. The T-STCNN with 3 attention blocks is presented in Figure 8.1.

3.2 3D Attention Block

3D attention block, inspired by the work carried out in 2D Wang et al. (2017a), takes as input a 4D data block of size $(N \times W \times H \times T)$ as illustrated in Figure 8.2.

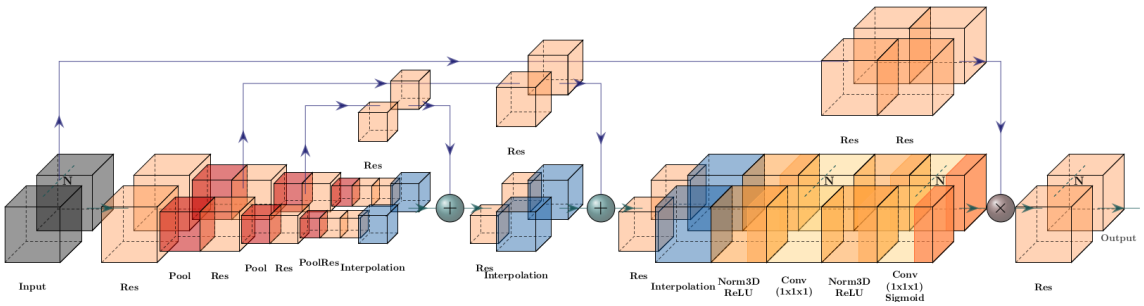


Figure 8.2 – 3D attention block architecture.

In this block, all convolution presented uses the same number of filters, N, to maintain the dimension of the processed data. Our input data are processed by a first 3D residual block, denoted as "*res*", presented in Section 3.3. Our network then splits in two branches: the trunk branch consisting of 2 successive 3D residual blocks (equation 8.1) and the soft floating mask branch (lowest position in Figure 8.2), described in equations 8.2, 8.3, 8.5. Its role is to accentuate the features generated by the trunk branch. Those two branches are merged as described in equation 8.6.

$$branch_{trunk}(.) = res(res(.)) \tag{8.1}$$

The soft mask branch is constituted of several 3D residual blocks followed by Max Pooling layers, denoted as "*MaxP*". It increases the reception field of convolutions using a bottom-up architecture, denoted as $f_{bu}(.) = res(MaxP(.))$. The lowest resolution is obtained after three Max Pooling steps.

$$\begin{aligned} x_1 &= f_{bu}(res(Input)) \\ x_2 &= f_{bu}(x_1)) \\ x_3 &= f_{bu}(x_2)) \end{aligned} \tag{8.2}$$

The information is then extended by a symmetrical top-down architecture, $f_{td}(.) = Inter(res(.))$, to project the input features of each resolution level. "*Inter*" denotes the trilinear interpolations Kenwright (2015) used for up-sampling. Two skipped connections are used for collecting information at different scales.

$$\begin{aligned} y_1 &= f_{td}(x_3) + res(x_2) \\ y_2 &= f_{td}(y_1) + res(x_1) \\ y_3 &= f_{td}(y_2) \end{aligned} \tag{8.3}$$

The soft mask branch is then composed of 2 successive layers. Each includes a 3D batch normalization, denoted as $F_n(.)$ as described by equation 8.8, followed by a ReLU activation function and a convolution layer with kernel sizes $(1 \times 1 \times 1)$. This is expressed by equation 8.4:

$$f_{conv}(.) = conv(ReLU(F_n(.))) \tag{8.4}$$

It ends with a sigmoid function, denoted as "Sig", to scale values between zero and one. These two layers are depicted on the right of the lowest branch in Figure 8.2 and are expressed by equation 8.5.

$$branch_{fmask}(Input) = Sig(f_{conv}(f_{conv}(y3))) \tag{8.5}$$

The output of our trunk branch is then multiplied term by term by $(1 \oplus branch_{fmask}(Input))$ where $branch_{fmask}(Input)$ is the output of the mask branch. The result is then processed by the last 3D residual block $res(.)$ which ends the attention block, see equation 8.6.

$$y = res(branch_{trunk}(Input) \odot (1 + branch_{fmask}(Input))) \tag{8.6}$$

Here the $\odot$ is an element-wise multiplication and $+$ is a classic addition of a scalar to each vector component.

3.3 3D Residual Block

Implemented 3D residual block inspired by the work carried out in 2D in He et al. (2016), takes as input a 4D data block of size $(N \times W \times H \times T)$ representing respectively the number of channels, the two spatial dimensions and the temporal dimension. The architecture of the block is represented in Figure 8.3.

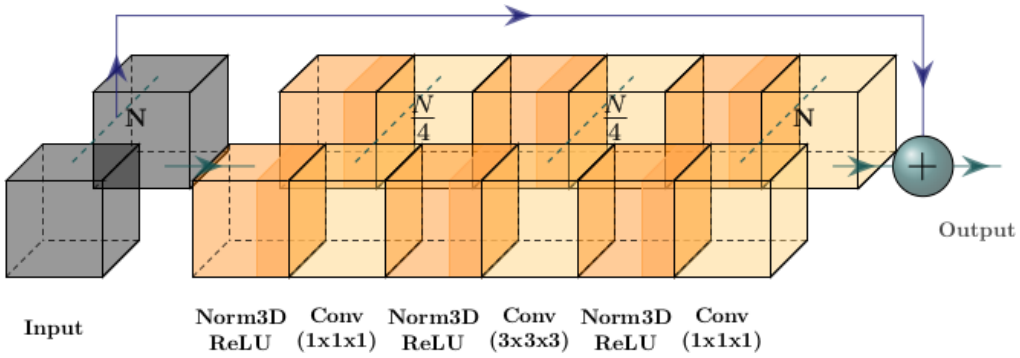


Figure 8.3 – 3D residual block architecture.

Input data are then processed by 3 successive layers $f_{conv_i}, i = 1, ..., 3$ (eq. 8.5). The result of residual block is the sum of the output of these three successive layers and our input data:

$$res(x) = f_{conv_3}(f_{conv_2}(f_{conv_1}(x))) + x \tag{8.7}$$

Here, the first layer f_{conv_1} uses $\frac{N}{4}$ convolution filters of size $(1 \times 1 \times 1)$, the second layer f_{conv_2} uses $\frac{N}{4}$ convolution filters of size $(3 \times 3 \times 3)$. Finally, the third layer f_{conv_3} employs N convolution filters of size $(1 \times 1 \times 1)$.

The 3D batch normalization, described in Ioffe and Szegedy (2015), is performed channel by channel over the batch of data. If we have $x = (x_1, x_2, ..., x_{N_{channels}})$, then the normalization is $F_n(x) = (f_n(x_1), f_n(x_2), ..., f_n(x_{N_{channels}}))$ with:

$$f_n(x_i) = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} * \gamma_i + \beta_i \tag{8.8}$$

with $i = 1, ...N_{channels}$, μ_i and σ_i the mean and standard deviation vectors of x_i computed over the batch, γ_i and β_i learnable parameters per channel and the division by $\sqrt{\sigma_i^2 + \epsilon}$ is element-wise. Here, $N_{channels} = N$ or $\frac{N}{4}$, depending on the normalization position in the residual block.

4 Experiments and Results

To assess the efficiency of the attention block for capturing qualitative features, we compare the classification results of the model with and without attention blocks on the `TTStroke-21` dataset. We also compare our model with the Two-Stream I3D model as done in the previous chapters.

4.1 Visualizing the Impact of the Attention Mechanism on Features

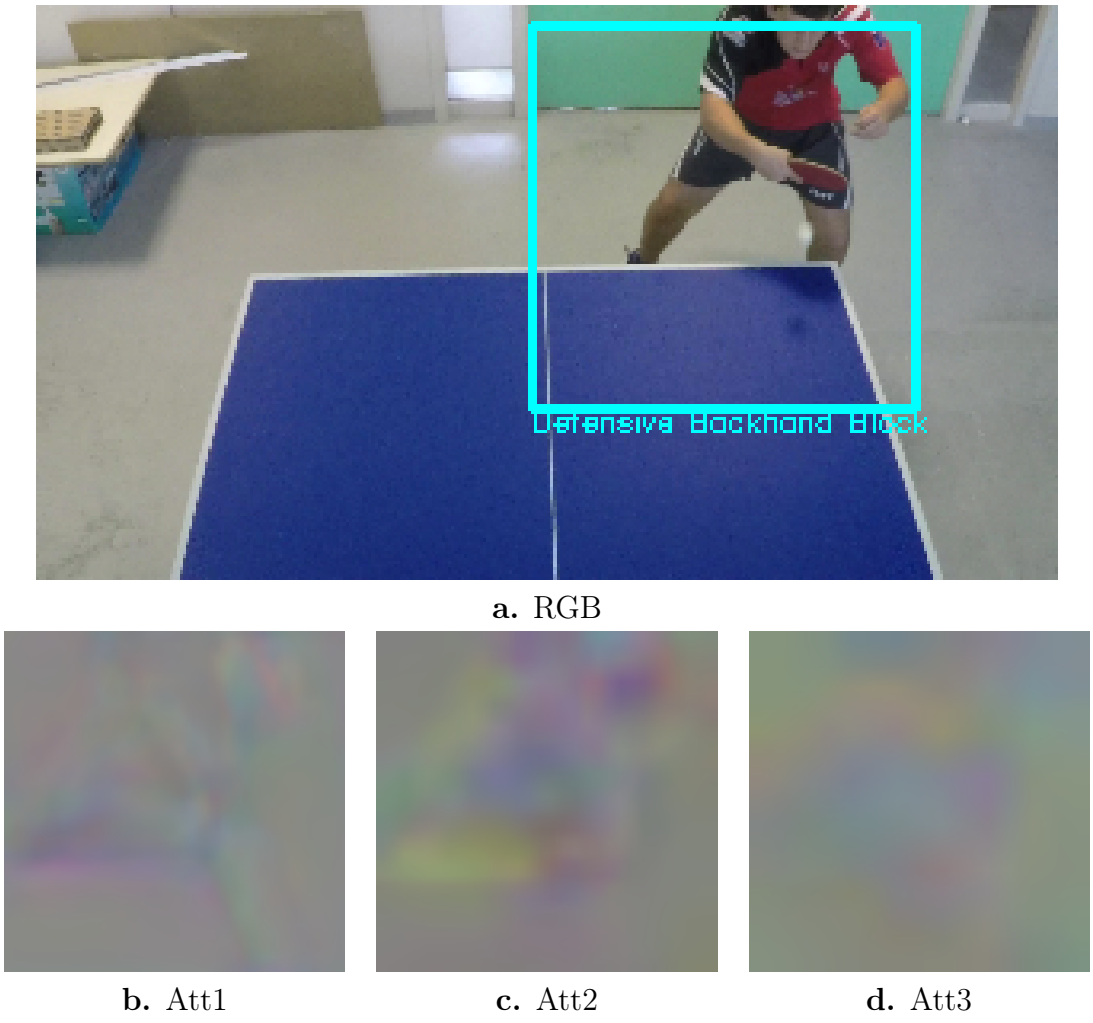
The attention block highlights features that contribute the most to the classification. In this way, the model can learn faster meaningful features in the classification task. Figure 8.4 shows outputs of floating mask branch of each attention block for a RGB image input to the T-STCNN model. Feature values range from zero to one, but are normalized using min-max normalization and resized for better visualization.

RGB input is of size $(100 \times 120 \times 120)$, each dimension representing respectively time, width and height. These parameters were fixed experimentally as a function of video resolution, frame-rate and stroke speed. The output size of the soft mask branch of attention blocks decreases by a factor two. Figure 8.4b, c and d represents 3 channels at a specific time. It can be noticed how the network focuses on the table's edges, on the player and even on the ball (more visible in Figure 8.4c). To classify a stroke, it is important to observe the posture of the player but also his/her position with respect to the table. The ball position and trajectory can also be of high importance to classify the stroke. On the whole training set, output values of the soft mask branch range between 0.35 and 0.65, meaning no features are totally left out or overrated, on the contrary.

4.2 Convergence of the Models

Conducted experiments required to change the values of the hyperparameters used in the previous chapters. Indeed, the number of parameters to train, which depends on the number of attention or residual blocks, greatly increased compared to our first experiments without attention mechanisms (see Table 8.1).

We compared our results with the I3D models which contains around 25 times more parameters to train compared to our models with attention blocks. Their model uses inception modules introduced in Szegedy et al. (2015) which are combination of different 3D convolutional layers using different filter sizes and concatenating their output. We trained their model according to their instructions with RGB data and optical flow trained separately. The training process differs according to the type of data: a larger number of iterations is required for optical flow, with a specific scheduled learning rate. The output of the two models on the test set can be combined together to improve the performances as shown in Table 8.3. We train their model using a time window of $T = 100$ frames which we have selected



a. RGB

b. Att1 **c.** Att2 **d.** Att3

Figure 8.4 – Visualisation of soft mask branch output from each attention block of the RGB branch of the T-STCNN model. The RGB input segmented from the original frame and its class is represented in light blue on **a**.

Table 8.1 – Number of parameters* to learn according to the architecture of the model.

Models	without attention blocks	with 3 attention blocks
RGB-STCNN	180 800	498 420
Flow-STCNN	179 990	497 610
T-STCNN	360 790	996 030
Carreira and Zisserman (2017) models		
RGB-I3D	∼ 12.5M	
Flow-I3D	∼ 12.5M	
Two-Stream I3D	∼ 25M	

* parameters of the fully connected layers are not considered.

after several experiments. Comparisons with $T = 64$ is also conducted in previous chapters.

Furthermore, the type of model trained (RGB-STCNN, Flow-STCNN, or T-STCNN) also influences the training process. Since different combinations and number of blocks were tested, the learning rate during training had to be adapted. A learning rate scheduler wasused, which reduced and increased the learning rate when the observed metric reached a plateau. Weights and state of the model were saved when it was performing the best and we re-loaded when the learning rate was changed. This allowed to re-start the optimization process from the past state with a new step-size in the gradient descent optimizer.

We started training with a learning rate of 0.01. A number of epochs: *patience*, set to 50, was considered before updating the learning rate, unless the performance drastically dropped (in our case: 0.7 of the best validation accuracy obtained).

The metric of interest was the training loss: if its average on the last 25 epochs was greater than its average on the 35 epochs before, the process was re-started from the past state and the learning rate divided by ten until reaching 10^{-5}. After this step, the learning rate was set back to 0.01 and process continued. These numbers of epochs were set empirically after preliminary experiments. This techniques differs from decreasing only by step (Zagoruyko and Komodakis, 2016) since we might re-increase the learning rate no amelioration is observed, similarly to work of Loshchilov and Hutter (2017) with warm restart technique.

It is worth mentioning that convergence is slower when using this method of learning rate re-scheduler with our past architectures introduced in Chapters 4, 5 and 6. Here, this strategy is however efficient to adapt and find adequate learning rates during training for different architecture configurations.

When comparing models with and without attention blocks as illustrated in 8.5, it can be noticed that our training process requires less epochs to adapt to our models with attention blocks. The convergence is faster and after the same number of epochs (500) models with attention blocks outperform models without attention.

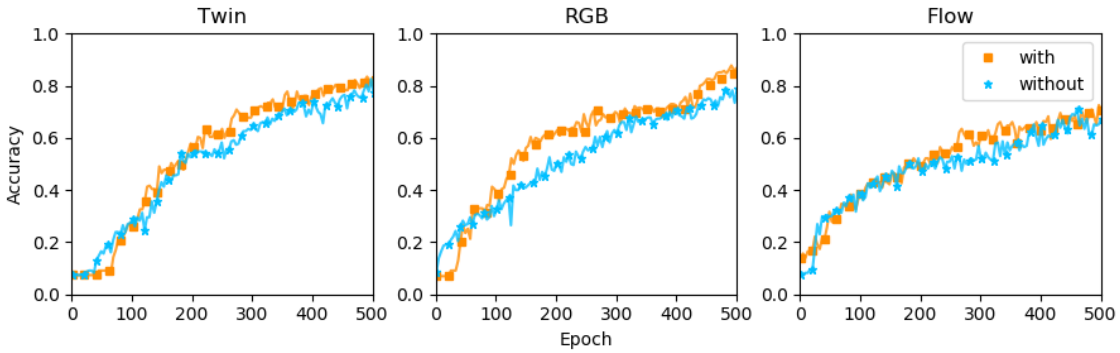


Figure 8.5 – Evolution of the validation accuracy for the different models with and without attention blocks.

Table 8.2, depicts these performances of our different models compared with I3D ones.

Table 8.2 – Comparison of the classification performances after 500 epochs for each model.

	Accuracies in %		
Models	Train	Validation	Gap
I3D-RGB	95.1	59.6	35.5
RGB-STCNN	83.9	78.7	**5.2**
RGB-STCNN with attention	96.3	**87.8**	8.5
I3D-Flow	97.7	55.7	42
Flow-STCNN	93	71.8	21.2
Flow-STCNN with attention	87.4	**72.6**	**14.8**
T-STCNN	88.9	82.6	**6.3**
T-STCNN with attention	92.7	**83.5**	9.2

The gap between validation accuracy and train accuracy is also reported in Table 8.2. At 500 epochs, the I3D models already overfit the training data. We can also notice that this gap increases for the RGB-STCNN and T-STCNN models when using attention mechanism, which might lead to limitation with greater number of epochs. However the Flow-STCNN model does ameliorate validation accuracy, and reduces the gap performances between Validation and Train sets . We think the OF data benefits a lot from the batch normalization in the attention mechanism, which corroborates our hypothesis in Chapter 5 when comparing with Flow-I3D model.

Analysing results further in Table 8.2, the T-STCNN model performs worse than the RGB-STCNN model and still needs training as presented Section 4.3. Its slower convergence can be due to the increased number of parameters to train. It also certainly comes from the batch size used during training, which had to be decreased from ten to five because of resource limitations.

4.3 Performances on Pure Classification Task

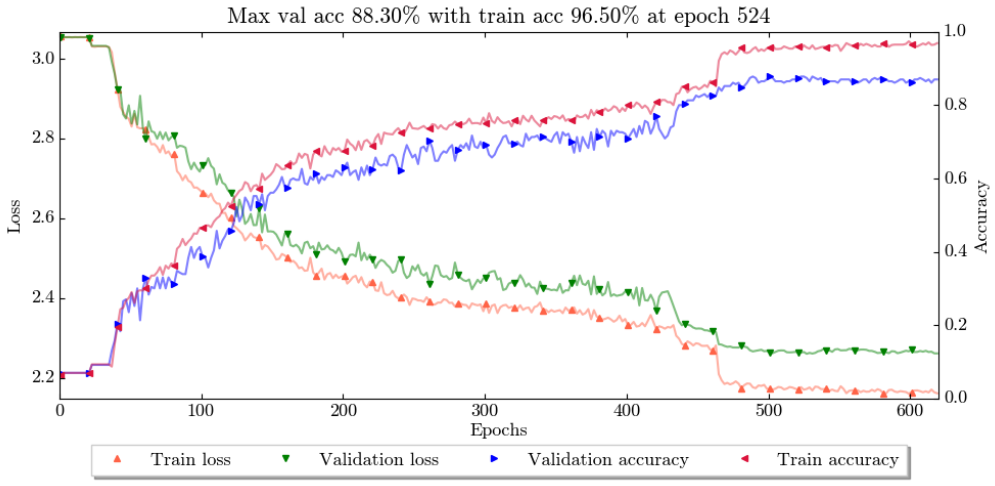
Our implemented attention blocks have shown to lead to faster convergence. In Table 8.3, we compare the models in term of accuracy for the pure classification task. In order to have an overall view, comparison is done with the models using three attention blocks (one after each max pooling layer) with the models presented in previous chapters.

The first classification scores obtained with the models using attention mechanism were surprisingly rather low, especially for the joint detection and classification task. The reason of such limited performances may be the use of the batch normalization in the attention blocks. It might benefit the convergence process, which are shown for each model in Figure 8.6; but it can also limit the generalization of the extracted features.

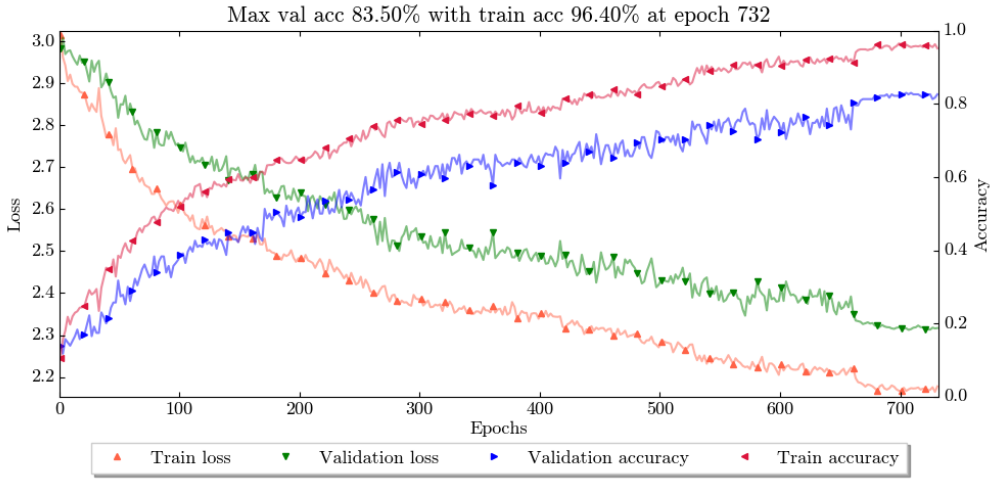
Table 8.3 – Comparison of the classification performances for models using attention mechanism after convergence in terms of accuracy.

		Accuracies in %					
Models	**Epochs**	**Train**	**Val**	**Test**	**TVote**	**TAvg**	**TGauss**
RGB-I3D	778	98.3	72.6	69.8	84.5	84.5	84.5
RGB-STCNN	1665	96.7	88.7	89.8	67.6	74.6	70.3
RGB-STCNN with Attention	524	96.5	88.3	92.4 93.2*	93.2 94.9*	94.1 95.8*	92.4 **96.6***
Flow-I3D	1112	98.8	74.8	73.3	82.8	82.8	82.8
Flow-STCNN	1449	97.5	79.6	75.9	80.2	80.2	78.5
Flow-STCNN with Attention	732	96.4	83.5	85.6 90.7*	66.1 71.2*	71.2 69.5*	66.1 70.3*
Two-Stream I3D	-	99.2	76.2	75.9	84.5	87.1	86.2
LF-STCNN	-	97	88.7	89.8	87.3	87.3	87.3
LF-STCNN with attention	-	97	88.7	90.7 94.9*	90.7 93.2*	92.4 94.1*	92.4 94.1*
T-STCNN	1784	95.8	87.8	93.2	91.5	90.7	91.5
T-STCNN with Attention	591	97.3	87.8	92.4 95.8*	71.2 77.1*	72 78*	72 77.1*

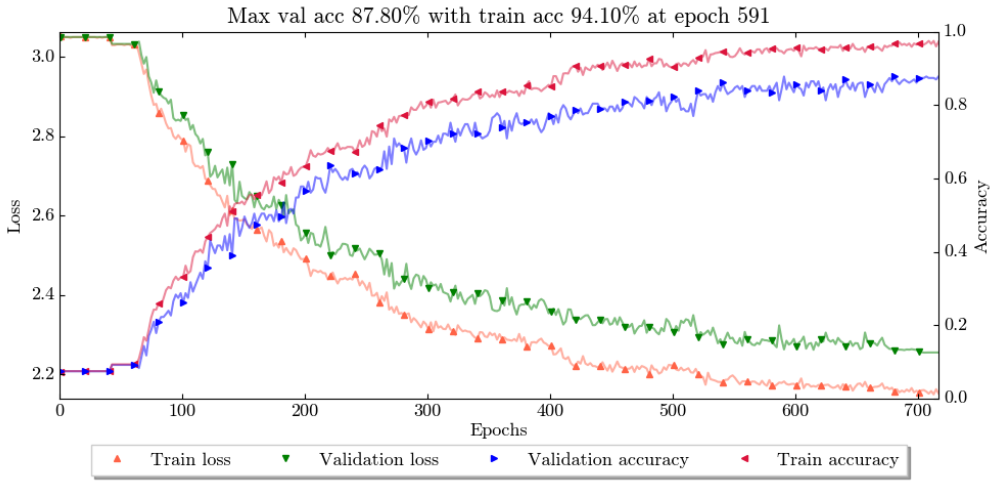
* with classical batch normalization



a. RGB-STCNN



b. Flow-STCNN



c. T-STCNN

Figure 8.6 – Training process of the different models using attention mechanism.

Batch normalization indeed computes statistics over input data of the model. By doing so, the model learns the value distribution at each layer during the training step. Estimated mean and standard deviation are then used during inference. It results that if the model encounters the same stroke but with slower motion, normalized motion vectors values will be too low to be classified as a stroke. However, if the motion is well distributed over the strokes, the statistic method should wield good performances. By not normalising by statistics mean and sigma during inference, classical mean and standard deviation are computed over the batch. We decided to perform tests using the two options: with statistics normalization and with classical batch normalization. The results classical batch normalization are shown in the second line of the models using attention mechanism. The results are therefore depending also on the batch-size used and the order of the test samples during inference. We keep the same order of the data for each model, and use a batch size of 5 for the classification and the joint detection and classification tasks, which results are described in Section 4.4.

The best results are obtained with the RGB-STCNN model using "TGauss" rule decision during inference. The Twin model comes only second using the classical "Test" decision. This is certainly due to the OF which do not adapt well to the temporal rule decision as it can be noticed when analysing the Flow-STCNN scores. Also, the T-STCNN model, due to its greater number of parameters, had to be trained with a lower batch size (five) compared to the single branch models (ten). This factor can also lead to lower performances since the model learns from a lower number of different samples at each iteration. From the visualization analysis, Figure 8.4, and results in Table 8.3, it can be argued that since the attention mechanism learns to focus on areas where RGB data are changing with respect to time, its contribution is lesser for models fed with temporal information such as the optical flow.

Overall, we can notice better performances with the models using attention blocks with using classical batch normalization. However the models using OF modality for classification are less stable when considering the whole stroke for classification. Indeed the scores drop of 20% of accuracy for the Flow-STCNN and T-STCNN models. On the other hand, the deterioration of performance with temporal smoothing is no longer observed on the RGB-STCNN model, certainly due to the increased receptive field generated by the incorporation of attention blocks. The fusion of the RGB-STCNN and Flow-STCNN, both with attention blocks, do not perform as good as the RGB-STCNN model alone. It may be because the Flow-STCNN misleads the decision. When considering the whole stroke duration, the performances remains nevertheless the same, proving a better stability of the model.

Furthermore, by analyzing the two best performances in the confusion matrix, Figure 8.7 for the RGB-STCNN model and Figure 8.8 for the T-STCNN model, we can notice the difficulty to classify under represented classes that are "Defensive Forehand Block" and "Defensive Backhand Push". Same behaviour was observed in

previous chapters. This may be solved by adding video samples to under represented stroke classes in the `TTStroke-21` dataset.

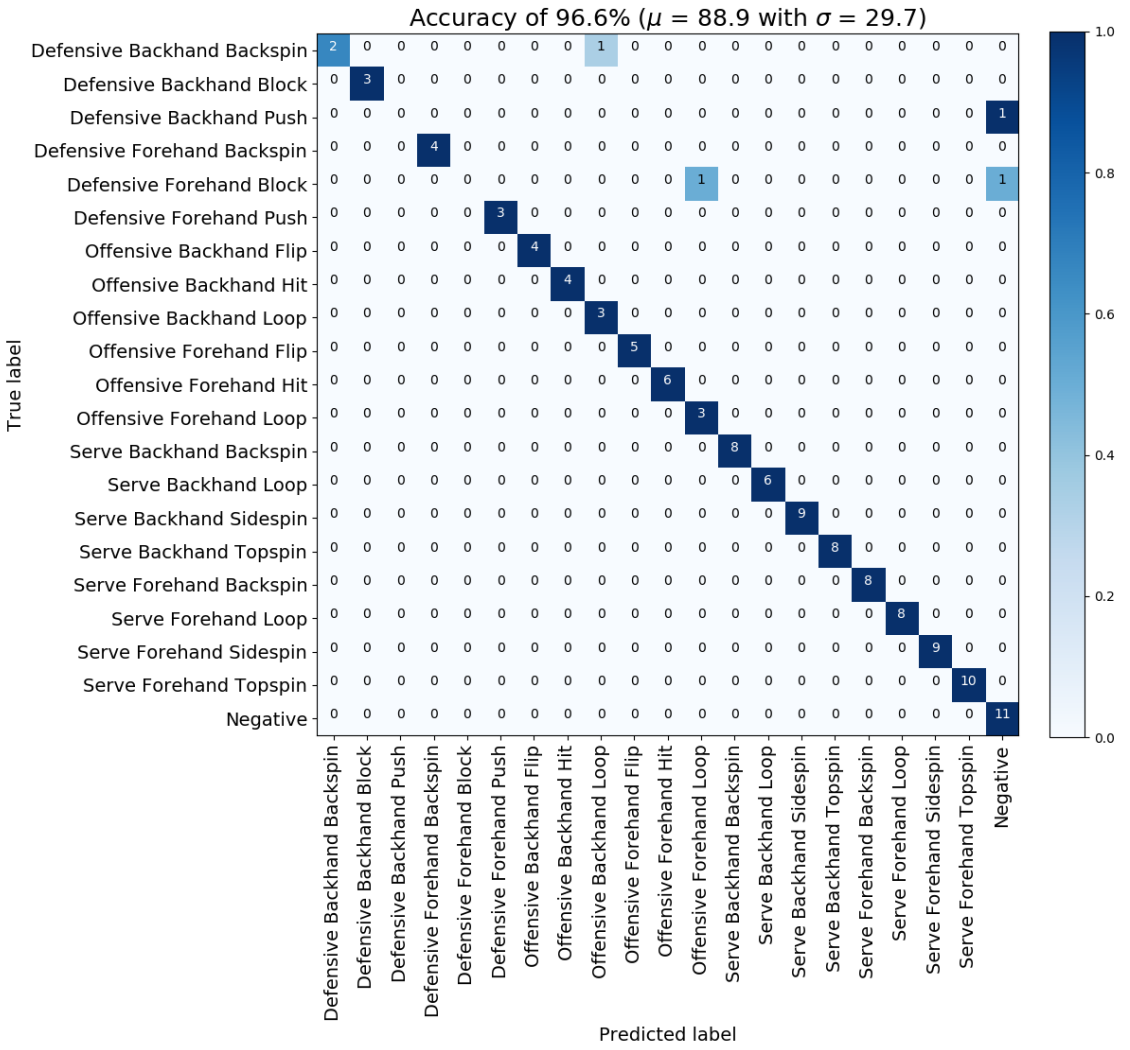


Figure 8.7 – Confusion Matrix of the RGB-STCNN model with attention mechanism with classical batch normalization using “Gauss” method decision.

Finally, by using the attention mechanism, we observe faster convergence in term of epoch (or iterations), and an improvement in performances compared to the previous models and our baseline RGB-I3D, Flow-I3D and Two-Stream I3D.

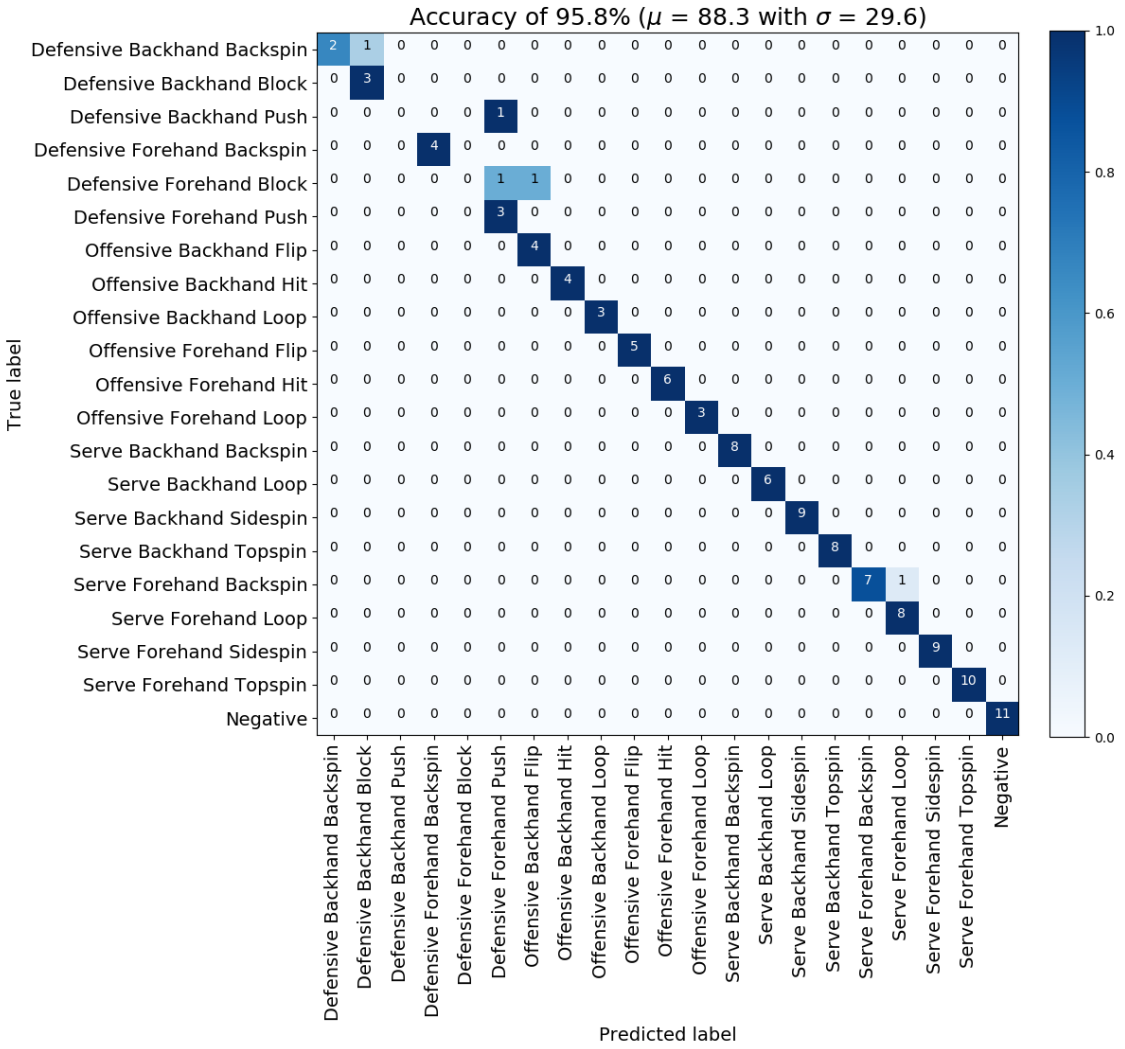


Figure 8.8 – Confusion Matrix of the T-STCNN model with attention mechanism with classical batch normalization using classic "Test" method decision.

4.4 Performances on Joint Stroke Detection and Classification Task

In Table 8.4, we show the accuracies obtained for each model with the attention mechanism on the joint stroke detection and classification task. We also performed extra tests with the LF-STCNN to see if the models using only one modality could benefit from each others. As previous chapters, results are reported when all labels are considered, and also when negative labels are left aside. With our choice to be flexible at the temporal border of the stroke, the decision is considered correct when classified as either the previous or either the next label. By not taking into account negative labels, overlaps are not considered and the evaluation may be more discriminant.

To have an overall view of the performances for each model, results from previous chapters are also reported.

Results for this task are mitigated. Best performance are obtained with the RGB-STCNN model with classical batch normalization, using the "Average" decision and reach 86.4% of accuracy. However this score drop to 74.8% when not considering the negative labels. On the other hand, best performance when not considering negative labels are obtained with the RGB-STCNN model but when running the statistics normalization, using the "Gaussian" decision and reaches 84.6% of accuracy. This score drops to 63.6% when considering negative samples.

Furthermore, the models using OF data are strongly affected by the attention mechanism and their performances are unstable. It affects too the fusion step, either late fusion or middle fusion (T-STCNN). However the T-STCNN seems to suffer less from this behaviour than the LF-STCNN.

Additionally, the Twin model with attention mechanism during inference is the slowest to perform classification. Still it can process 100 frames (830 ms), using one GPU in 82 ms (against 52 ms without attention). Therefore, real-time processing would be possible with the proper equipment, by classifying incoming stream every 10 frames. However, it would imply to have instant optical flow data. The RGB-STCNN, which takes 48 ms with attention blocks and 39 ms without to classify one sample is thus more fitted for such application.

Finally, the most stable results are obtained using the T-STCNN without attention mechanism, which obtains 79.7% and 78.4% accuracy respectively with and without considering the negative class using the "Gaussian" decision rule. Moreover, the mean average precision, widely used in segmentation problem (Everingham et al., 2010), also highlights the superiority of this model: 0.574 for this model against 0.564 for the RGB-STCNN model running the statistics normalization using the "Gaussian" decision.

Table 8.4 – Performance of stroke detection and classification.

	Accuracies in %			
Models	**Gross**	**Vote**	**Average**	**Gaussian**
RGB-STCNN	57	80.1	80.8	80.2
RGB-STCNN with Attention	43.8 70.1*	63.3 85.9*	64.7 **86.4***	63.6 86.1*
Flow-STCNN	70.3	80.5	80.9	81
Flow-STCNN with Attention	10.7 69.3*	20.1 78.4*	21.5 79.2*	21.1 79.8*
LF-STCNN with Attention	15.2 69.2*	26.9 78.5*	28.5 79.2*	28 79.8*
T-STCNN	60.8	79.8	80.2	79.7
T-STCNN with Attention	31 72.9*	46.8 82.1*	47.7 82.3*	47.3 83*
without taking into account negative labels				
RGB-STCNN	41.5	44.8	46.2	49.1
RGB-STCNN with Attention	65.4 66.9*	80.4 74.3*	81.9 74.8*	**84.6** 77.6*
Flow-STCNN	50.4	55.4	59.2	62.4
Flow-STCNN with Attention	40 33.8*	52.9 20.9*	55.8 22.9*	58.6 26.5*
LF-STCNN with Attention	32.6 33.8*	61.2 21.9*	64.5 23.5*	67.2 27.1*
T-STCNN	60.5	76.8	76.9	78.4
T-STCNN with Attention	45.2 45.6*	63.8 35.1*	65.6 35*	67.9 39.4*

* with classical batch normalization

5 Conclusion

In this chapter, we have extended the work carried out in 2D He et al. (2016); Wang et al. (2017a) to implement 3D residual blocks and 3D attention blocks. We have incorporated them in our Spatio-Temporal Convolutional Neural Networks for fine-grained action recognition in video on `TTStroke-21`.

According to the visualization of the soft mask branch, it is safe to say that the attention blocks focus on meaningful features such as motion, body parts, position of the player with respect to the table, rackets and ball. We also noticed a greater efficiency of the attention mechanism on RGB data.

We have shown that 3D attention blocks enable faster convergence of the models in terms of epochs and lead to better performances too. Liu et al. (2019a) draw similar conclusions when using recurrent 3D attention for object recognition using 3D shape. In our case, model with attention mechanism outperform the models introduced in Part II. However the amount of parameters to learn and the size of the network increase, which force us to decrease the batch size for training the T-STCNN model. Even if the convergence is faster in term of epochs, each epoch takes more time: from 85 to 200 seconds with T-STCNN respectively without and with attention blocks.

However, as presented in Chapter 2, "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale" propose a transformer for image classification based on attention mechanism. The latter does not use any convolution and is fast to train. Their results let us question on the use of CNNs and of attention mechanisms which increase exponentially the number of weights to train, thus requiring bigger computing resources for training. Those limitations narrow the applications and reduces their performances for video classification.

This Part III was dedicated to the modifications we could bring to the already introduced networks, in order to increase their performance. We choose to take the path of attention mechanism. The performances increased for the classification task, but at the cost of the stability of the performances on the joint detection and classification task. Other ways could be considered for improving the already good performances of our models. They are not yet implemented, but are presented in the conclusion of this manuscript.

General Conclusion and Perspectives

Conclusion

In this thesis manuscript, we have presented our contributions for fine-grained action recognition with the case study of stroke classification in table tennis. Our motivation is to develop tools for athletes in order to improve their performance.

In the first part of this work, Chapters 1 and 2 presented the state-of-the-art methods for action classification and the best performances were obtained using 3D Convolutional Neural Networks. Furthermore, methods considering both modalities, RGB and Optical Flow, proved their superiority, especially when models were trained from scratch. Chapter 3 described the many datasets used by the scientific community to benchmark the action classification methods. Their evolution in terms of classes, acquisition process and number of videos were highlighted. The same chapter introduced the `TTStroke-21` dataset, a dataset dedicated to the recognition of table tennis strokes. With regards to the state-of-the-art methods, we proposed a method based on Spatio-temporal Convolutional neural Networks, in order to capture efficiently the temporal evolution of the different strokes. We also chose to use the seminal work of Carreira and Zisserman (2017), as our baseline, and trained their model on our dataset.

Part II introduced our Spatio-Temporal Convolutional Neural Network models. Chapter 4 and Chapter 5 introduced the use of a single modality in the proposed model, respectively RGB stream and optical flow stream. Chapter 6 presented a Twin architecture to fuse both modalities in the network. An ablation study was preformed to analyse the contribution of the different normalization methods, the data augmentation process, colour information, the different modalities and the different fusion methods. Performances were analysed in terms of accuracy for two distinct tasks: pure classification, and joint detection and classification. The best performances were obtained with the Twin Spatio-Temporal Convolutional Neural Network (T-STCNN). The features obtained with this last model were analysed using a new feature understanding method based on back propagation of strong features. The observation of these features showed correlation between learned characteristics and image regions where a trained eye, such as a table tennis professional, would focus on to classify a stroke.

Part III focused on improvements of the presented models for increasing their classification performances. This is investigated through the incorporation of attention mechanisms in their architecture. The attention mechanism proved its efficiency by stressing meaningful characteristics for classification, boosting the convergence process and increasing classification performances. However, a lack of stability can appear, depending on the normalization method used in the attention blocs.

Without attention mechanism, the best performances were obtained with the T-STCNN. When considering attention mechanism, best performances were obtained with the RGB-STCNN using attention blocks. Despite improved results, inconstancy in performances was noticed. This instability was important on the OF modality, thus impacting the whole Twin model. Ultimately, the most stable model

performing well on both classification and detection+classification tasks, remains the T-STCNN without attention mechanism.

Perspectives

The presented work, despite an ablation study and comparison with another method, can still be conducted deeper by analysing further possible combinations of the network architectures. For instance, the attention mechanism may be better incorporated in the network to avoid instability issues. This work can also be extended in different ways. The use of another dataset to pre-train our model could be performed, or comparisons of performances using other datasets too. Furthermore, the state-of-the-art has evolved and new methods could be applied to `TTStroke-21`, such as optical flow methods for the characterisation of the data (Ilg et al., 2017), or the type of architecture for classification, such as a transformer model (Vaswani et al., 2017) or an architecture using Multi-Head Attention layers (Kalfaoglu et al., 2020). In addition, `TTStroke-21` has been continuously enriched, with new videos at different frame rates and viewpoint:further experiments could be conducted to test the robustness of the proposed model on such augmented dataset. This task will be performed through the Sport Task of MediaEval workshop which is conducted every year.

The aim of this thesis work is to develop a method to help students or athletes in their training. Fine-grained stroke classification is one step towards the understanding of player performance. In the scope of the CRISP project, Calandre et al. (2021) focus on modelling the ball kinematic parameters, such as speed, rotation and trajectory. This could lead to better understanding and evaluation of the strokes performed. New methods are also interested in semantic segmentations in table tennis games (Voeikov et al. (2020)[1]). This segmentation might also help the classification process and the performance evaluation. To give an insight, a semantic segmentation using the open source model of He et al. (2020a) is shown in Figure 8.9. Even if globally efficient, the method is not specifically designed for table tennis and one can observe wrongly annotated parts in the images. Furthermore, as no temporal information is used, results are inconsistent over time, which leaves a big room for improvement.

Moreover, we have been confronted, during the classification process, with the reliability of the ground truth that we have collected. Thus, the building by *crowdsourcing* of the `TTStroke-21` dataset induced a variability in the expertise of the annotators, as they could be students but also teachers or high level players. This process generates annotation noise on the labels (type of stroke played), but also on the temporal boundaries (location of the stroke). So far, we have manually filtered out cases of mismatched annotations. However, this issue is part of the domain of *weak supervision*, which is currently a very active topic (Chesneau, 2018; Ratner et

[1] https://lab.osai.ai/datasets/openttgames/

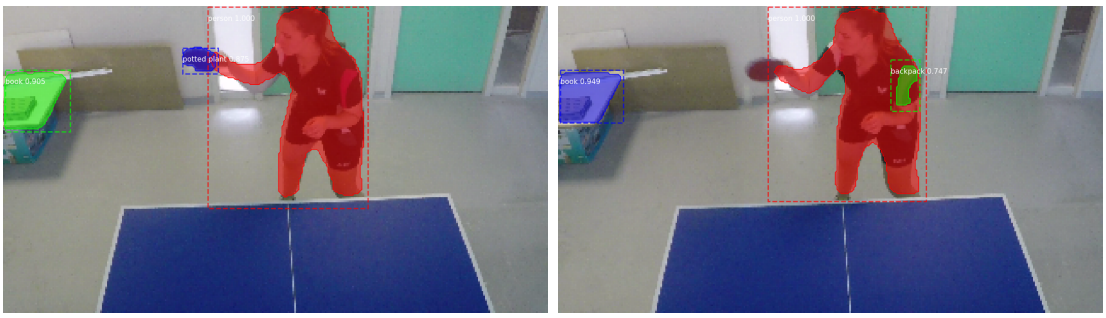


Figure 8.9 – Semantic segmentation of two successive frames of a `TTStore-21` sample (He et al., 2020a).

al., 2019). This is a perspective that we would like to develop, especially since the dataset is continuously enriched.

Another perspective for fine-grained sport gesture analysis is the use of human joints, also called "pose". On the one hand, pose can be used for action classification (Luvizon et al., 2018; Choutas et al., 2018; Zheng et al., 2020); but it can also be used to give qualitative measures for performance analysis of a sport (Morel et al., 2017; Einfalt et al., 2018). However, in our case, qualitative measures extracted from 2D joint skeletons might lead to limited results because of the fine-grained aspect of table tennis strokes. A more promising approach would be to leverage a 3D model of the human joints. For that reason, another interesting perspective would be to combine depth and pose, both computed from the RGB stream, here again to avoid wearable sensors in the acquisition process. Figure 8.10 presents preliminary results of such possible combination.

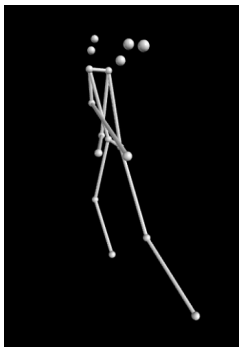
This model is still temporally unstable, because both, the depth and pose, are not perfectly estimated. Consideration of the temporal information in order to refine the results might lead to a more robust pose estimation. With such a model, one could extract qualitative measures of a gesture performed by a player, by comparing it with a reference stroke for instance.

To conclude, this work had a special focus on table tennis but presented methods are generic. The same protocol could be extended to other sports. Although, it requires to have a dedicated dataset for proper training of the algorithm. The proposed annotation platform can be extended to different sport activities. Likewise, the presented Spatio-Temporal Convolutional Neural Network can be adapted to match the number of classes of the considered sport.

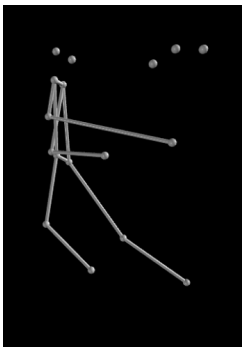
This manuscript and its related publications are contributions to fine-grained action classification, and we hope that other artificial intelligence methods can be built from this work. If there are still steps to take, we are optimistic about the fact that the expansion of computer vision will transform, in a near future, key areas of performance analysis in sport to improve the experience of sportsmen and women in their practice.

a. Estimated pose from RGB image (Newell et al., 2016)

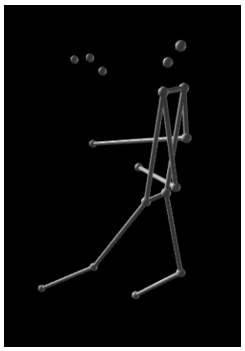
b. Estimated depth from RGB image (Ramamonjisoa and Lepetit, 2019)



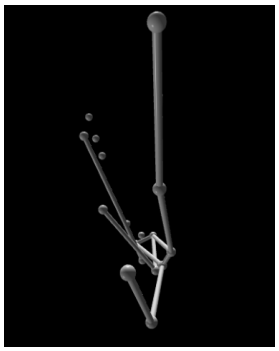
c. Front 3D view



d. Side 3D view



e. Back sided 3D view



f. Underneath 3D view

Figure 8.10 – 3D skeleton visualization from the combination of the pose and the depth estimated from a single RGB image.

Appendix

Appendix A

Publications Related to the Thesis

International Journal

[Martin et al. 2020] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks. In: *Multim. Tools Appl.* 79 (2020), no. 27-28, pp. 20429–20447. – URL https://doi.org/10.1007/s11042-020-08917-3

Book Chapter

[Martin et al. 2021] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Zemmari, Akka : Multi-faceted Deep Learning: Models and Data. In: *Springer* (2021)

International Conferences

[Fuad et al. 2020] Fuad, Kazi Ahmed A. ; Martin, Pierre-Etienne ; Giot, Romai ; Bourqui, Romain ; Benois-Pineau, Jenny ; Zemmari, Akka : Feature Understanding in 3D CNNs for Actions Recognition in Video. In: *Tenth International Conference on Image Processing Theory, Tools and Applications, IPTA 2020, Paris, France, November 9-12, 2020*, URL https://doi.org/1, 2020, pp. 1–6

[Martin et al. 2019a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud : Fine-Grained Action Detection and Classification in Table Tennis with Siamese Spatio-Temporal Convolutional Neural Network. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*, URL https://doi.org/10.1109/ICIP.2019.8803382, 2019, pp. 3027–3028

[Martin et al. 2018] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis. In: *2018 International Conference on Content-Based Multimedia Indexing, CBMI 2018, La Rochelle, France,*

September 4-6, 2018, URL https://doi.org/10.1109/CBMI.2018.8516488, 2018, pp. 1–6

[Martin et al. 2019b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Optimal Choice of Motion Estimation Methods for Fine-Grained Action Classification with 3D Convolutional Networks. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*, URL https://doi.org/10.1109/ICIP.2019.8803780, 2019, pp. 554–558

[Martin et al. 2021] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : 3D attention mechanisms in Twin Spatio-Temporal Convolutional Neural Networks. Application to action classification in videos of table tennis games. In: *25th International Conference on Pattern Recognition (ICPR2020) - MiCo Milano Congress Center, Italy, 10-15 January 2021*, IEEE Computer Society, 2021

International Workshop

[Martin et al. 2019a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Mascarilla, Laurent ; Calandre, Jordan ; Morlier, Julien : Sports Video Annotation: Detection of Strokes in Table Tennis Task for MediaEval 2019. In: *Working Notes Proceedings of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-30 October 2019*, URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_6.pdf, 2019

[Martin et al. 2020a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Mascarilla, Laurent ; Calandre, Jordan ; Morlier, Julien : Sports Video Classification: Classification of Strokes in Table Tennis for MediaEval 2020. In: *Proc. of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, 2020

[Martin et al. 2019b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Morlier, Julien : Siamese Spatio-Temporal Convolutional Neural Network for Stroke Classification in Table Tennis Games. In: *Working Notes Proceedings of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-30 October 2019*, URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_58.pdf, 2019

[Martin et al. 2020b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Morlier, Julien : Classification of Strokes in Table Tennis with a Three Stream Spatio-Temporal CNN for MediaEval 2020. In: *Proc. of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, 2020

Appendix B

Sports Video Classification: Classification of Strokes in Table Tennis for MediaEval 2019 and 2020

Abstract

Fine-grained action classification has raised new challenges compared to classical action classification problem. In contrast with classical action recognition datasets which comprise a wide variety of diverse actions, we focus on one sport which is table tennis. Sport video analysis is a very popular research topic, due to the variety of application areas, ranging from multimedia intelligent devices with user-tailored digests, up to analysis of athletes' performances. Running since 2019 as a part of MediaEval, we offer a task which consists in classifying table tennis strokes from videos recorded in natural conditions at the University of Bordeaux. The aim is to build tools for teachers, coaches and players to analyse table tennis games. Such tools could lead to an automatic profiling of the player and the training session could then be adapted for improving more efficiently the sportsmen and sportswomen skills.

1 Introduction

Action detection and classification is one of the main challenges in visual content analysis and mining (Stoian et al., 2016). Over the last few years, the number of datasets for action classification has drastically increased in terms of video content, resolution, localization and number of classes. However the latest research shows that classification performed using deep neural networks often focuses on the whole scene and the background and not on the action itself.

Sport video analysis has been a very popular research topic, due to the variety of application areas, ranging from multimedia intelligent devices with user-tailored digests, up to analysis of athletes' performance (Einfalt et al., 2018). The Sport Video Classification project was initiated between the Faculty of Sports STAPS of the University of Bordeaux and the computer science laboratories LaBRI of the University of Bordeaux and MIA of La Rochelle University. This work is supported

by the New Aquitania Region through CRISP project - ComputeR vIsion for Sport Performance and the MIRES federation. The goal of this project is to develop artificial intelligence and multimedia indexing methods for the recognition of table tennis sport activities. The ultimate goal is to evaluate the performance of athletes, with a particular focus on students, in order to develop optimal training strategies. To that aim, a video corpus named `TTStroke-21` was recorded with volunteered players. These data represent are of great scientific interest for the Multimedia community participating in the MediaEval campaign.

Several datasets such as UCF101 (Soomro et al., 2012), HMDB51 (Kuehne et al., 2011) and AVA (Gu et al., 2018) have been used for many years as benchmarks for action classification methods. In the work of Liu and Hu (2019), spatio-temporal dependencies are learned from the video using only RGB images for classification. This method is promising but scores are still below the multi-modal methods of I3D (Carreira and Zisserman, 2017). More recently, datasets have been enriched, like JHMDB (Jhuang et al., 2013) and Kinetics (Smaira et al., 2020) or fused like AVA_Kinetics (Li et al., 2020). Some also focus on the intra-class dissimilarity such as the Something-Something dataset. Others, such as the Olympic Sports dataset (Niebles et al., 2010), focus on sport actions only. However those datasets are not dedicated to a specific sport and its associated rules. Few datasets focus on fine-grained classification. We can cite FineGym recently introduced by Shao et al. (2020), which focuses on gymnastic videos. Similarly, our dataset `TTStroke-21` (Martin et al., 2020c) focuses on table tennis strokes.

`TTStroke-21` is manually annotated by professional players or teachers of table tennis, making the annotation process longer, but more temporally and qualitatively accurate. Classification methods such as I3D models or LTC model (Varol et al., 2018) performing well on UCF101 dataset inspired the work done by Martin et al. (2018, 2020c) which introduces a T-STCNN - Twin Spatio Temporal Convolutional Neural Network. Here, the video stream and derived computed optical flow are passed through the branches of the T-STCNN. Martin et al. (2019d) also investigated the normalization of the flow in order to improve the classification score. They also introduce an attention block to improve the performances and speed of convergence (Martin et al., 2021a). The inter-similarity of actions - strokes - in `TTStroke-21` makes the classification task challenging and the multi-modal method seemed to improve performances. To better understand learned features and classification process taking place in the T-STCNN, we also developed a new visualization technique (Fuad et al., 2020).

Recent work focusing on Table Tennis Wang et al. (2020) tries to get the tactics of the players based on their performances during matches using a Markov chain model. Liu et al. (2019b); Xia et al. (2020); Tabrizi et al. (2020) are interested on stroke recognition using sensors. Voeikov et al. (2020) focus on segmentation of the player, ball coordinates and event detection while Wu and Koike (2020); Lin et al. (2020) focus solely on the trajectory of the ball.

In Section 2, we first introduce the specific conditions of usage of this data then

describe `TTStroke-21` and the task respectively in Sections 3 and 4. The evaluation method is explained in Section 5. Supplementary notes are shared in Section 6. More information may be found on the dedicated GitHub web page[1].

2 Specific Conditions of Usage

`TTStroke-21` is constituted of videos with players playing table tennis in natural conditions. Even if we are using an automatic tool for blurring players' faces, some faces are misdetected on few frames and thus some players remain identifiable. In order to respect the personal data and privacy of the players, this dataset is subject to an usage agreement, referred to as *Special Conditions*. These *Special Conditions* apply to the use of videos, referred to as Images, generated in the framework of the program Sports video classification: classification of strokes in table tennis, for the implementation of the MediaEval program. They correspond to the specific usage agreement referred to in the *Usage agreement for the MediaEval 2020 Research Collections*, signed between the User and the University of Delft. The full and complete acceptance, without any reservation, of these *Special Conditions* is a mandatory prerequisite for the provision of the Images as part of the MediaEval 2020 evaluation campaign. A complete reading of these conditions is necessary and requires the user, for example, to obscure the faces (blurring, black banner, etc.) in the video before use in any publication and to destroy the data by October 1st, 2021.

3 Dataset Description

In the MediaEval 2019 and 2020 campaign, we released a subset of the `TTStroke-21` dataset which has been specifically recorded in a sport faculty facility using a lightweight equipment, such as GoPro cameras. It is constituted of player-centred videos recorded in natural conditions without markers or sensors, see Figure B.1.

It comprises 20 table tennis stroke classes, i.e. 8 services: `Serve Forehand Backspin`, `Serve Forehand Loop`, `Serve Forehand Sidespin`, `Serve Forehand Topspin`, `Serve Backhand Backspin`, `Serve Backhand Loop`, `Serve Backhand Sidespin`, `Serve Backhand Topspin`; 6 offensive strokes: `Offensive Forehand Hit`, `Offensive Forehand Loop`, `Offensive Forehand Flip`, `Offensive Backhand Hit`, `Offensive Backhand Loop`, `Offensive Backhand Flip`; and 6 defensive strokes: `Defensive Forehand Push`, `Defensive Forehand Block`, `Defensive Forehand Backspin`, `Defensive Backhand Push`, `Defensive Backhand Block`, `Defensive Backhand Backspin`. Also all the strokes can be divided in two super-classes: Forehand and Backhand. This taxonomy was designed with professional table tennis teachers.

[1] https://multimediaeval.github.io/2020-Sports-Video-Classification-Task/

a. Video acquisition **b.** Annotation platform

Figure B.1 – `TTStroke-21` acquisition process.

All videos are recorded in MPEG-4 format. Unlike the task at MediaEval 2019 Martin et al. (2019a), most of the faces have been blurred for MediaEval 2020 Martin et al. (2020a). To do so, faces are detected with OpenCV deep learning face detector, based on the Single Shot Detector (SSD) framework with a ResNet base network, for each frame of the original video. The detected face is blurred and frames are re-encoded in a video.

The organisation of the delivered data is as follows:

- The provided dataset is split into two subsets: i) training set and ii) test set;
- In each directory, there are several videos (in MPEG-4 format) and each video may contain several actions;
- Each video file is provided with a XML file describing the actions present in the video and if the player is right-handed or left-handed;
- Each action has 3 attributes: the starting frame, the ending frame, and the stroke class;
- In the train set XML files, all the attributes are specified. In the test set XML files, only the starting and ending frames are specified. The stroke class attribute is purposely set to value: "Unknown", and should be updated by the participants to one of the 20 valid classes.

.

4 Task Description

The Sport Video Annotation task consists, for each action of each test video, in assigning a label using a given taxonomy of 20 classes of Table Tennis strokes.

Participants may submit up to five runs. For each run, they must provide one XML file per video file containing, with the actions associated with the recognised stroke class. Runs may be submitted as an archive (zip or tar.gz file) with each run in a different directory. Participants should also indicate if any external data, such as other dataset or pretrained networks, was used to compute their runs. The task is considered fully automatic. Once the video are provided to the system, results should be produced without any human intervention.

5 Evaluation

For MediaEval 2019 and MediaEval 2020, we proposed a light-weight classification task. It consists in classification of table tennis strokes which temporal borders are supplied in the XML files accompanying each video file. Hence for each test video the participants are invited to produce an XML file in which each stroke is labelled accordingly to the given taxonomy. This means that the default label "unknown" has to be replaced by the label of the stroke class that the participant's system has assigned. All submissions will be evaluated in terms of *per-class accuracy* (A_i) and of global accuracy (GA). The A_i is computed for each i-th class as:

$$A_i = TP_i/(N_{gti}) \tag{B.1}$$

Here TP_i is the number of True Positives, i.e. correctly labelled, by the participant's system, strokes for the given i-th class,N_{gti} is the number of recorded strokes of the i-th class in the test dataset.

$$GA = TP/(N_{gt}) \tag{B.2}$$

Here $TP = \sum TP_i$ is the number of correctly labelled strokes for the whole dataset, and N_{gt} is the number of strokes in the ground truth - the whole test set.

The organizers will also provide to the participants different confusion matrices: one considering all the classes, and others considering the type of the stroke such as: offensive, defensive and defensive and/or using forehand and backhand superclasses of the strokes.

6 Discussion

In 2019 (Martin et al., 2019a) we had six fully registered participants to our task and three submitted their runs. They had reached a maximum accuracy of 22.9% (Martin et al., 2019b), 14.1%(Calandre et al., 2019) and 11.3% (Sriraman et al., 2019) leaving room for improvement. Results were presented the 27^{th} until the 30^{th} of October 2019 in Sophia Antipolis, France.

In 2020 we have ten fully registered participants. Results will be presented online due to the Covid19 pandemic the 11^{th} until 15^{th} of December 2020.

Appendix C

Source Code Available on GitHub

In order to give access to the implementation details of this thesis work, the code has been shared on the online platform GitHub[1]. The dedicated web page is represented in Figure C.1.

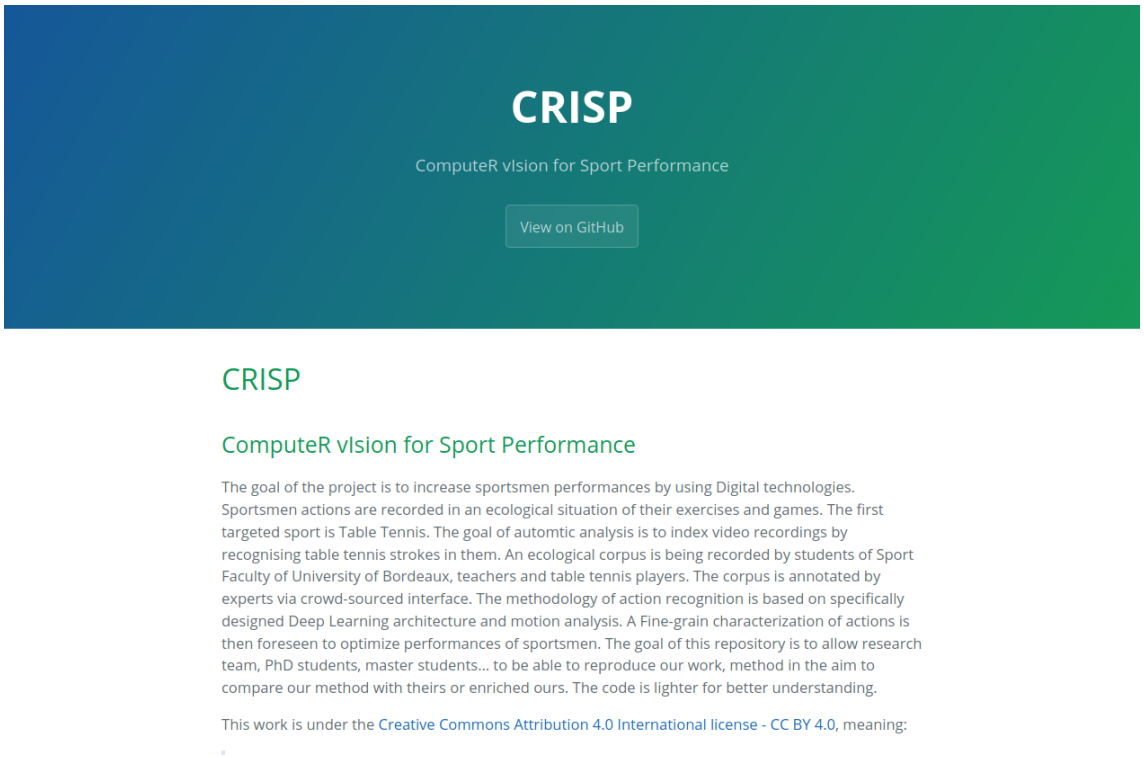


Figure C.1 – GitHub web page dedicated to CRISP project and its classification models implementation.

It has been similarly done for the `MediaEval 2020 Sports Video Classification` task[2].

[1] https://p-emartin.github.io/CRISP/
[2] https://multimediaeval.github.io/2020-Sports-Video-Classification-Task/

Appendix D

Scientific Popularisation

This thesis was also the object of various popularisation events. We report some of them in this appendix.

1 MT180s

MT180s is a popularisation challenge dedicated to PhD thesis in which the candidates must, with simple words, explain is PhD topic in 180 seconds. This contest is in french but its equivalent exists in English: 3MT (3 minutes My Thesis). MT180s has been performed at different occasions[1] as depicted in Figures D.1 and D.2. 3MT has also been performed at ICIP conference in 2019 (Martin et al., 2019c).

Figure D.1 – MT180s competition as representative of the University of Bordeaux.

[1] https://www.youtube.com/watch?v=5aBVSrVLrks

Figure D.2 – MT180s for animating the 80^{th} anniversary of The French National Centre for Scientific Research (CNRS): “Villages des 80ans du CNRS”. Credits to Gautier DUFAU for the photography.

2 Ma Thèse en 1024 Charactères

The *Société Informatique de France* (SIF) or the *Computer Science Society* in English, offers a newsletter dedicated to computer science. This newsletter appears three times a year. A section is dedicated to the ongoing thesis and have to be presented in exactly 1024 characters. This thesis led to one contribution in this section[2]:

"*Ma thèse porte sur la reconnaissance des gestes sportifs à partir de vidéos et j'applique mes travaux au tennis de table.*
Le but est de programmer un environnement informatique intelligent sur lequel étudiants et enseignants peuvent analyser la façon de jouer des sportifs. Le logiciel permet de segmenter et de classifier automatiquement les coups de tennis de table effectués par les joueurs à partir de vidéos. Ainsi le profil des joueurs peut être renseigné et l'enseignant peut adapter son cours pour améliorer au mieux leurs performances.
Pour ce faire, nous avons enregistré des jeux de tennis de table avec des étudiants. Ces enregistrements ont ensuite été annotés temporellement par des professionnels sur une plateforme participative d'annotation.
Cette nouvelle base de données, surnommée TTStroke21, nous permet d'entraîner et de tester notre modèle.
On introduit un réseau de neurones jumeau à convolutions spatio-temporelles prenant en entrée le flux vidéo et le flot optique. Traitées parallèlement, ces données permettent une classification efficace des segments de vidéos. A partir de ces classifications les frontières temporelles des coups effectués et leur classe sont renseignées."

[2] https://www.labri.fr/projet/AIV/pemartin/1024-numero-15_Article28.pdf

3 La Nuit des Chercheurs

The European Researchers' Night is an event that each year, for one evening, allows the public and researchers to meet in a festive and friendly atmosphere. This event (Figure D.3), which mobilizes more than 100 cities throughout Europe and 12 in France, is organized in Bordeaux by Cap Sciences, the University of Bordeaux and the University Bordeaux Montaigne. This thesis has been the object to several form of popularisation during this event in 2018 and 2019:

- MT180s as described above.
- speed searching: speed dating principle, meeting of the public that changes every 8 minutes.
- meeting in the dark: meeting of the audience in a dark space, interactions on the thesis topic for 15 minutes.
- The story desk: 3 objects are presented to the public who must guess your research.

Figure D.3 – Teaser frame of The European Researchers' Night 2018.

4 Jamming Assembly

The Jamming Assembly is a digital creation contest by team and on a common theme. Once the themes have been announced, the teams have 48 hours to complete their production. This thesis work was one the theme. Figure D.4 is an extracted frame of one of the projects which led to a short movie[3].

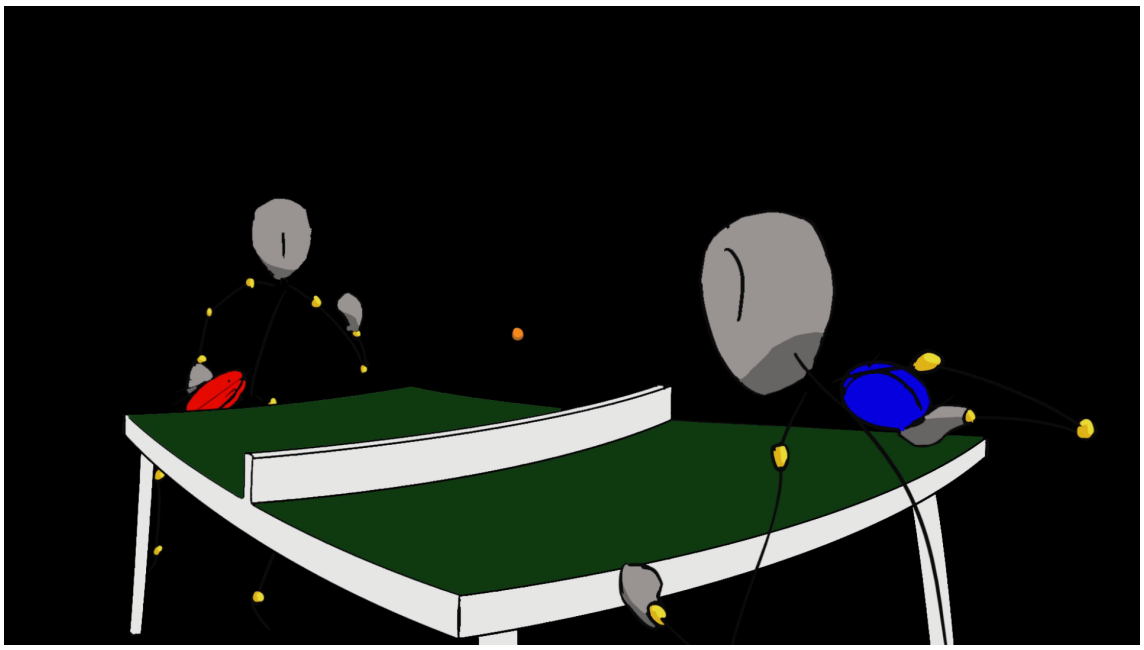


Figure D.4 – Snapshot from short film made during Jamming Assembly.

[3] https://vimeo.com/454459965 with password crisp

Bibliography

[Abu-El-Haija et al. 2016] Abu-El-Haija, Sami ; Kothari, Nisarg ; Lee, Joonseok ; Natsev, Paul ; Toderici, George ; Varadarajan, Balakrishnan ; Vijayanarasimhan, Sudheendra : YouTube-8M: A Large-Scale Video Classification Benchmark. In: *CoRR* abs/1609.08675 (2016). – URL http://arxiv.org/abs/1609.08675

[Adato et al. 2011] Adato, Yair ; Zickler, Todd E. ; Ben-Shahar, Ohad : A polar representation of motion and implications for optical flow. In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. (IEEE Computer Society, 2011), pp. 1145–1152. – URL https://doi.org/10.1109/CVPR.2011.5995419. – ISBN 978-1-4577-0394-2

[Adebayo et al. 2018] Adebayo, Julius ; Gilmer, Justin ; Muelly, Michael ; Goodfellow, Ian J. ; Hardt, Moritz ; Kim, Been : Sanity Checks for Saliency Maps. (2018), pp. 9525–9536. – URL http://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps

[Ahmadi et al. 2015] Ahmadi, Amin ; Mitchell, Edmond ; Richter, Chris ; Destelle, François ; Gowing, Marc ; O'Connor, Noel E. ; Moran, Kieran : Toward Automatic Activity Classification and Movement Assessment During a Sports Training Session. In: *IEEE Internet Things J.* 2 (2015), no. 1, pp. 23–32. – URL https://doi.org/10.1109/JIOT.2014.2377238

[Aizawa et al. 2018] Aizawa, Kiyoharu (Publ.) ; Lew, Michael S. (Publ.) ; Satoh, Shin'ichi (Publ.) : *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ICMR 2018, Yokohama, Japan, June 11-14, 2018*. ACM, 2018. – URL http://dl.acm.org/citation.cfm?id=3206025

[Akbarian et al. 2017] Akbarian, Mohammad Sadegh A. ; Saleh, Fatemehsadat ; Salzmann, Mathieu ; Fernando, Basura ; Petersson, Lars ; Andersson, Lars : Encouraging LSTMs to Anticipate Actions Very Early. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. (IEEE Computer Society, 2017b), pp. 280–289. – URL https://doi.org/10.1109/ICCV.2017.39. – ISBN 978-1-5386-1032-9

[Alayrac et al. 2018] Alayrac, Jean-Baptiste ; Bojanowski, Piotr ; Agrawal, Nishant ; Sivic, Josef ; Laptev, Ivan ; Lacoste-Julien, Simon : Learning from Narrated Instruction Videos. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018), no. 9, pp. 2194–2208. – URL https://doi.org/10.1109/TPAMI.2017.2749223

[Arik et al. 2017] Arik, Sercan O. ; Kliegl, Markus ; Child, Rewon ; Hestness, Joel ; Gibiansky, Andrew ; Fougner, Christopher ; Prenger, Ryan ; Coates, Adam : Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting. In: Lacerda, Francisco (Publ.): *Interspeech 2017, 18th Annual Conference of*

the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017, ISCA, 2017, pp. 1606–1610. – URL http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1737.html

[Asnaoui and Radeva 2020] Asnaoui, Khalid E. ; Radeva, Petia : Automatically Assess Day Similarity Using Visual Lifelogs. In: *J. Intell. Syst.* 29 (2020), no. 1, pp. 298–310. – URL https://doi.org/10.1515/jisys-2017-0364

[Association for Computer Linguistics 2014] : *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*. The Association for Computer Linguistics, 2014. – URL https://www.aclweb.org/anthology/volumes/W14-33/. – ISBN 978-1-941643-17-4

[Baccouche et al. 2011] Baccouche, Moez ; Mamalet, Franck ; Wolf, Christian ; Garcia, Christophe ; Baskurt, Atilla : Sequential Deep Learning for Human Action Recognition. In: Salah, Albert A. (Publ.) ; Lepri, Bruno (Publ.): *Human Behavior Unterstanding - Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings* 7065, Springer, 2011, pp. 29–39. – URL https://doi.org/10.1007/978-3-642-25446-8_4. – ISBN 978-3-642-25445-1

[Baker et al. 2011] Baker, Simon ; Scharstein, Daniel ; Lewis, J. P. ; Roth, Stefan ; Black, Michael J. ; Szeliski, Richard : A Database and Evaluation Methodology for Optical Flow. In: *Int. J. Comput. Vis.* 92 (2011), no. 1, pp. 1–31. – URL https://doi.org/10.1007/s11263-010-0390-2

[Barron et al. 1994] Barron, John L. ; Fleet, David J. ; Beauchemin, Steven S. : Performance of optical flow techniques. In: *Int. J. Comput. Vis.* 12 (1994), no. 1, pp. 43–77. – URL https://doi.org/10.1007/BF01420984

[Bay et al. 2008] Bay, Herbert ; Ess, Andreas ; Tuytelaars, Tinne ; Gool, Luc V. : Speeded-Up Robust Features (SURF). In: *Comput. Vis. Image Underst.* 110 (2008), no. 3, pp. 346–359. – URL https://doi.org/10.1016/j.cviu.2007.09.014

[Beaudry et al. 2014] Beaudry, Cyrille ; Péteri, Renaud ; Mascarilla, Laurent : Action recognition in videos using frequency analysis of critical point trajectories. In: *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, IEEE, 2014, pp. 1445–1449. – URL https://doi.org/10.1109/ICIP.2014.7025289. – ISBN 978-1-4799-5751-4

[Bellman and Kalaba 1959] Bellman, R. ; Kalaba, R. : On adaptive control processes. In: *IRE Transactions on Automatic Control* 4 (1959), no. 2, pp. 1–9

[Bilen et al. 2018] Bilen, Hakan ; Fernando, Basura ; Gavves, Efstratios ; Vedaldi, Andrea : Action Recognition with Dynamic Image Networks. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018), no. 12, pp. 2799–2813. – URL https://doi.org/10.1109/TPAMI.2017.2769085

[Billy et al. 2019] Billy, Antoine ; Pouteau, Sébastien ; Desbarats, Pascal ; Chaumette, Serge ; Domenger, Jean-Philippe : Adaptive SLAM with Synthetic Stereo Dataset Generation for Real-time Dense 3D Reconstruction. In: Trémeau, Alain (Publ.) ; Farinella, Giovanni M. (Publ.) ; Braz, José (Publ.): *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2019, Volume 5: VISAPP, Prague, Czech Republic, February 25-27, 2019*, SciTePress, 2019, pp. 840–848. – URL https://doi.org/10.5220/0007386508400848. – ISBN 978-989-758-354-4

[Bojarski et al. 2018] Bojarski, Mariusz ; Choromanska, Anna ; Choromanski, Krzysztof ; Firner, Bernhard ; Ackel, Larry J. ; Muller, Urs ; Yeres, Philip ; Zieba, Karol : VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving. In: *ICRA*, IEEE, 2018, pp. 1–8

[Bolaños et al. 2015] Bolaños, Marc ; Dimiccoli, Mariella ; Radeva, Petia : Towards Storytelling from Visual Lifelogging: An Overview. In: *CoRR* abs/1507.06120 (2015). – URL http://arxiv.org/abs/1507.06120

[Bolaños et al. 2018] Bolaños, Marc ; Peris, Álvaro ; Casacuberta, Francisco ; Soler, Sergi ; Radeva, Petia : Egocentric video description based on temporally-linked sequences. In: *J. Vis. Commun. Image Represent.* 50 (2018), pp. 205–216. – URL https://doi.org/10.1016/j.jvcir.2017.11.022

[Bollacker et al. 2008] Bollacker, Kurt D. ; Evans, Colin ; Paritosh, Praveen ; Sturge, Tim ; Taylor, Jamie : Freebase: a collaboratively created graph database for structuring human knowledge. In: Wang, Jason T. (Publ.): *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, ACM, 2008, pp. 1247–1250. – URL https://doi.org/10.1145/1376616.1376746. – ISBN 978-1-60558-102-6

[Boser et al. 1992] Boser, Bernhard E. ; Guyon, Isabelle ; Vapnik, Vladimir : A Training Algorithm for Optimal Margin Classifiers. In: Haussler, David (Publ.): *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, ACM, 1992, pp. 144–152. – URL https://doi.org/10.1145/130385.130401. – ISBN 0-89791-497-X

[Bossard et al. 2014] Bossard, Lukas ; Guillaumin, Matthieu ; Gool, Luc V. : Food-101 - Mining Discriminative Components with Random Forests. In: Fleet, David J. (Publ.) ; Pajdla, Tomás (Publ.) ; Schiele, Bernt (Publ.) ; Tuytelaars, Tinne (Publ.): *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI* 8694, Springer, 2014, pp. 446–461. – URL https://doi.org/10.1007/978-3-319-10599-4_29. – ISBN 978-3-319-10598-7

[Bost et al. 2019] Bost, Xavier ; Gueye, Serigne ; Labatut, Vincent ; Larson, Martha A. ; Linarès, Georges ; Malinas, Damien ; Roth, Raphaël : Remembering winter was coming - Character-oriented video summaries of TV series. In: *Multim. Tools Appl.* 78 (2019), no. 24, pp. 35373–35399. – URL https://doi.org/10.1007/s11042-019-07969-4

[Boutell 1997] Boutell, Thomas : PNG (Portable Network Graphics) Specification Version 1.0. In: *RFC* 2083 (1997), pp. 1–102. – URL https://doi.org/10.17487/RFC2083

[Buades et al. 2005] Buades, Antoni ; Coll, Bartomeu ; Morel, Jean-Michel : A Non-Local Algorithm for Image Denoising. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. (IEEE Computer Society, 2005), pp. 60–65. – URL https://doi.org/10.1109/CVPR.2005.38. – ISBN 0-7695-2372-2

[Budnik et al. 2017] Budnik, Mateusz ; Gutierrez-Gomez, Efrain-Leonardo ; Safadi, Bahjat ; Pellerin, Denis ; Quénot, Georges : Learned features versus engineered features for multimedia indexing. In: *Multim. Tools Appl.* 76 (2017), no. 9, pp. 11941–11958. – URL https://doi.org/10.1007/s11042-016-4240-2

[Butler et al. 2012] Butler, Daniel J. ; Wulff, Jonas ; Stanley, Garrett B. ; Black, Michael J. : A Naturalistic Open Source Movie for Optical Flow Evaluation. In: Fitzgibbon, Andrew W. (Publ.) ; Lazebnik, Svetlana (Publ.) ; Perona, Pietro (Publ.) ; Sato, Yoichi (Publ.) ; Schmid, Cordelia (Publ.): *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI* 7577, Springer, 2012, pp. 611–625. – URL https://doi.org/10.1007/978-3-642-33783-3_44. – ISBN 978-3-642-33782-6

[Bylinskii et al. 2019] Bylinskii, Zoya ; Judd, Tilke ; Oliva, Aude ; Torralba, Antonio ; Durand, Frédo : What Do Different Evaluation Metrics Tell Us About Saliency Models? In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (2019), no. 3, pp. 740–757. – URL https://doi.org/10.1109/TPAMI.2018.2815601

[Cai and Hu 2020] Cai, Jiahui ; Hu, Jianguo : 3D RANs: 3D Residual Attention Networks for action recognition. In: *Vis. Comput.* 36 (2020), no. 6, pp. 1261–1270. – URL https://doi.org/10.1007/s00371-019-01733-3

[Calandre et al. 2019] Calandre, Jordan ; Péteri, Renaud ; Mascarilla, Laurent : Optical Flow Singularities for Sports Video Annotation: Detection of Strokes in Table Tennis. In: (Larson et al., 2020). – URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_37.pdf

[Calandre et al. 2021] Calandre, Jordan ; Péteri, Renaud ; Mascarilla, Laurent ; Tremblais, Benoit : Extraction and analysis of 3D kinematic parameters of Table Tennis ball from a single camera. In: *ICPR 2020, 25th International Conference on Pattern Recognition (ICPR)*. (IEEE Computer Society, 2021). – URL https://hal.archives-ouvertes.fr/hal-02975085

[Calandre et al. 2020] Calandre, Jordan ; Péteri, Renaud ; Mascarilla, Laurent ; Tremblais, Benois : Extraction et analyse de trajectoires de balle de Tennis de Table à partir d'une seule caméra pour l'aide à la performance sportive. In: *RFIAP 2020, 23-26 Jun 2020, Vannes, France*, 2020, pp. 1–3

[de Campos et al. 2011] Campos, Teofilo de ; Barnard, Mark ; Mikolajczyk, Krystian ; Kittler, Josef ; Yan, Fei ; Christmas, William J. ; Windridge, David : An evaluation of bags-of-words and spatio-temporal shapes for action recognition. In: *IEEE Workshop on Applications of Computer Vision (WACV 2011), 5-7 January 2011, Kona, HI, USA*, IEEE Computer Society, 2011, pp. 344–351. – URL https://doi.org/10.1109/WACV.2011.5711524. – ISBN 978-1-4244-9496-5

[Cao et al. 2010] Cao, Liangliang ; Liu, Zicheng ; Huang, Thomas S. : Cross-dataset action detection. In: *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. (IEEE Computer Society, 2010), pp. 1998–2005. – URL https://doi.org/10.1109/CVPR.2010.5539875. – ISBN 978-1-4244-6984-0

[Carreira et al. 2018] Carreira, João ; Noland, Eric ; Banki-Horvath, Andras ; Hillier, Chloe ; Zisserman, Andrew : A Short Note about Kinetics-600. In: *CoRR* abs/1808.01340 (2018). – URL http://arxiv.org/abs/1808.01340

[Carreira et al. 2019] Carreira, João ; Noland, Eric ; Hillier, Chloe ; Zisserman, Andrew : A Short Note on the Kinetics-700 Human Action Dataset. In: *CoRR* abs/1907.06987 (2019). – URL http://arxiv.org/abs/1907.06987

[Carreira and Zisserman 2017] Carreira, João ; Zisserman, Andrew : Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 4724–4733. – URL https://doi.org/10.1109/CVPR.2017.502. – ISBN 978-1-5386-0457-1

[Cartas et al. 2017] Cartas, Alejandro ; Marín, Juan ; Radeva, Petia ; Dimiccoli, Mariella : Recognizing Activities of Daily Living from Egocentric Images. In: Alexandre, Luís A. (Publ.) ; Sánchez, José S. (Publ.) ; Rodrigues, Joao M. F. (Publ.): *Pattern Recognition and Image Analysis - 8th Iberian Conference, IbPRIA 2017, Faro, Portugal, June 20-23, 2017, Proceedings* 10255, Springer, 2017, pp. 87–95. – URL https://doi.org/10.1007/978-3-319-58838-4_10. – ISBN 978-3-319-58837-7

[Cartas et al. 2020] Cartas, Alejandro ; Radeva, Petia ; Dimiccoli, Mariella : Activities of Daily Living Monitoring via a Wearable Camera: Toward Real-World Applications. In: *IEEE Access* 8 (2020), pp. 77344–77363. – URL https://doi.org/10.1109/ACCESS.2020.2990333

[Ceroni et al. 2019] Ceroni, Andrea ; Ma, Chenyang ; Ewerth, Ralph : Mining exoticism from visual content with fusion-based deep neural networks. In: *Int. J. Multim. Inf. Retr.* 8 (2019), no. 1, pp. 19–33. – URL https://doi.org/10.1007/s13735-018-00165-4

[Chatterjee et al. 2020] Chatterjee, Soumick ; Saad, Fatima ; Sarasaen, Chompunuch ; Ghosh, Suhita ; Khatun, Rupali ; Radeva, Petia ; Rose, Georg ; Stober, Sebastian ; Speck, Oliver ; Nürnberger, Andreas : Exploration of Interpretability Techniques for Deep COVID-19 Classification using Chest X-ray Images. In: *CoRR* abs/2006.02570 (2020). – URL https://arxiv.org/abs/2006.02570

[Chaudhry et al. 2009] Chaudhry, Rizwan ; Ravichandran, Avinash ; Hager, Gregory D. ; Vidal, René : Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. (IEEE Computer Society, 2009a), pp. 1932–1939. – URL https://doi.org/10.1109/CVPR.2009.5206821. – ISBN 978-1-4244-3992-8

[Chen et al. 2018a] Chen, Chien-Wen ; Chen, Wen-Cheng ; Hu, Min-Chun : Doodle Master: A Doodle Beautification System Based on Auto-encoding Generative Adversarial Networks. In: Chu, Wei-Ta (Publ.) ; Tsumura, Norimichi (Publ.) ; Yamasaki, Toshihiko (Publ.) ; Shiratori, Takaaki (Publ.) ; Motomura, Hideto (Publ.): *Proceedings of the 2018 International Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia, MMArt&ACM@ICMR 2018, Yokohama, Japan, June 11, 2018*, ACM, 2018, pp. 2–7. – URL https://doi.org/10.1145/3209693.3209695

[Chen et al. 2018b] Chen, Di ; Zhang, Shanshan ; Ouyang, Wanli ; Yang, Jian ; Tai, Ying : Person Search via a Mask-Guided Two-Stream CNN Model. In: Ferrari, Vittorio (Publ.) ; Hebert, Martial (Publ.) ; Sminchisescu, Cristian (Publ.) ; Weiss, Yair (Publ.): *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII* 11211, Springer, 2018, pp. 764–781. – URL https://doi.org/10.1007/978-3-030-01234-2_45. – ISBN 978-3-030-01233-5

[Chen et al. 2018c] Chen, Xiaozhi ; Kundu, Kaustav ; Zhu, Yukun ; Ma, Huimin ; Fidler, Sanja ; Urtasun, Raquel : 3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018), no. 5, pp. 1259–1272. – URL https://doi.org/10.1109/TPAMI.2017.2706685

[Chen et al. 2018d] Chen, Yunpeng ; Kalantidis, Yannis ; Li, Jianshu ; Yan, Shuicheng ; Feng, Jiashi : Aˆ2-Nets: Double Attention Networks. In: *NeurIPS*, 2018, pp. 350–359

[Cheng et al. 2016] Cheng, Jianpeng ; Dong, Li ; Lapata, Mirella : Long Short-Term Memory-Networks for Machine Reading. In: Su, Jian (Publ.) ; Carreras, Xavier (Publ.) ; Duh, Kevin (Publ.): *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, The Association for Computational Linguistics, 2016, pp. 551–561. – URL https://doi.org/10.18653/v1/d16-1053. – ISBN 978-1-945626-25-8

[Chesneau 2018] Chesneau, Nicolas : *Learning to Recognize Actions with Weak Supervision. (Reconnaissance d'actions de manière faiblement supervisée)*, Grenoble Alpes University, France, Dissertation, 2018. – URL https://tel.archives-ouvertes.fr/tel-01893147

[Chesneau et al. 2018] Chesneau, Nicolas ; Alahari, Karteek ; Schmid, Cordelia : Learning From Web Videos for Event Classification. In: *IEEE Trans. Circuits Syst.*

Video Technol. 28 (2018), no. 10, pp. 3019–3029. – URL https://doi.org/10.1109/TCSVT.2017.2764624

[Choutas et al. 2018] Choutas, Vasileios ; Weinzaepfel, Philippe ; Revaud, Jérôme ; Schmid, Cordelia : PoTion: Pose MoTion Representation for Action Recognition. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. (IEEE Computer Society, 2018a), pp. 7024–7033. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Choutas_PoTion_Pose_MoTion_CVPR_2018_paper.html

[Çiçek et al. 2016] Çiçek, Özgün ; Abdulkadir, Ahmed ; Lienkamp, Soeren S. ; Brox, Thomas ; Ronneberger, Olaf : 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin, Sébastien (Publ.) ; Joskowicz, Leo (Publ.) ; Sabuncu, Mert R. (Publ.) ; Ünal, Gözde B. (Publ.) ; Wells, William (Publ.): *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016 - 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II* 9901, URL https://doi.org/10.1007/978-3-319-46723-8_49, 2016, pp. 424–432. – ISBN 978-3-319-46722-1

[Cohendet et al. 2018] Cohendet, Romain ; Yadati, Karthik ; Duong, Ngoc Q. K. ; Demarty, Claire-Hélène : Annotating, Understanding, and Predicting Long-term Video Memorability. In: (Aizawa et al., 2018), pp. 178–186. – URL https://doi.org/10.1145/3206025.3206056

[Collins et al. 2002] Collins, Michael ; Schapire, Robert E. ; Singer, Yoram : Logistic Regression, AdaBoost and Bregman Distances. In: *Mach. Learn.* 48 (2002), no. 1-3, pp. 253–285. – URL https://doi.org/10.1023/A:1013912006537

[Computer Vision Foundation / IEEE 2019] : *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019. – URL http://openaccess.thecvf.com/CVPR2019.py

[Crasto et al. 2019] Crasto, Nieves ; Weinzaepfel, Philippe ; Alahari, Karteek ; Schmid, Cordelia : MARS: Motion-Augmented RGB Stream for Action Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. (Computer Vision Foundation / IEEE, 2019), pp. 7882–7891. – URL http://openaccess.thecvf.com/content_CVPR_2019/html/Crasto_MARS_Motion-Augmented_RGB_Stream_for_Action_Recognition_CVPR_2019_paper.html

[Csurka and Perronnin 2010] Csurka, Gabriela ; Perronnin, Florent : Fisher Vectors: Beyond Bag-of-Visual-Words Image Representations. In: Richard, Paul (Publ.) ; Braz, José (Publ.): *Computer Vision, Imaging and Computer Graphics. Theory and Applications - International Joint Conference, VISIGRAPP 2010, Angers, France, May 17-21, 2010. Revised Selected Papers* 229, Springer, 2010, pp. 28–42. – URL https://doi.org/10.1007/978-3-642-25382-9_2. – ISBN 978-3-642-25381-2

[Dahl et al. 2013] Dahl, George E. ; Sainath, Tara N. ; Hinton, Geoffrey E. : Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, IEEE, 2013, pp. 8609–8613. – URL https://doi.org/10.1109/ICASSP.2013.6639346

[Dalal and Triggs 2005] Dalal, Navneet ; Triggs, Bill : Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. (IEEE Computer Society, 2005), pp. 886–893. – URL https://doi.org/10.1109/CVPR.2005.177. – ISBN 0-7695-2372-2

[Dalal et al. 2006] Dalal, Navneet ; Triggs, Bill ; Schmid, Cordelia : Human Detection Using Oriented Histograms of Flow and Appearance. In: Leonardis, Ales (Publ.) ; Bischof, Horst (Publ.) ; Pinz, Axel (Publ.): *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II* 3952, Springer, 2006, pp. 428–441. – URL https://doi.org/10.1007/11744047_33. – ISBN 3-540-33834-9

[Damen et al. 2018] Damen, Dima ; Doughty, Hazel ; Farinella, Giovanni M. ; Fidler, Sanja ; Furnari, Antonino ; Kazakos, Evangelos ; Moltisanti, Davide ; Munro, Jonathan ; Perrett, Toby ; Price, Will ; Wray, Michael : Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In: *CoRR* abs/1804.02748 (2018). – URL http://arxiv.org/abs/1804.02748

[Dang-Nguyen et al. 2018] Dang-Nguyen, Duc-Tien ; Riegler, Michael ; Zhou, Liting ; Gurrin, Cathal : Challenges and Opportunities within Personal Life Archives. In: (Aizawa et al., 2018), pp. 335–343. – URL https://doi.org/10.1145/3206025.3206040

[Das et al. 2019] Das, Srijan ; Dai, Rui ; Koperski, Michal ; Minciullo, Luca ; Garattoni, Lorenzo ; Brémond, François ; Francesca, Gianpiero : Toyota Smarthome: Real-World Activities of Daily Living. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. (IEEE, 2019b), pp. 833–842. – URL https://doi.org/10.1109/ICCV.2019.00092. – ISBN 978-1-7281-4803-8

[Daugman 1988] Daugman, John G. : Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. In: *IEEE Trans. Acoust. Speech Signal Process.* 36 (1988), no. 7, pp. 1169–1179. – URL https://doi.org/10.1109/29.1644

[Davies 2010] Davies, Mark : The Corpus of Contemporary American English as the first reliable monitor corpus of English. In: *Lit. Linguistic Comput.* 25 (2010), no. 4, pp. 447–464. – URL https://doi.org/10.1093/llc/fqq018

[Debard et al. 2018] Debard, Quentin ; Wolf, Christian ; Canu, Stéphane ; Arné, Julien : Learning to Recognize Touch Gestures: Recurrent vs. Convolutional Features and Dynamic Sampling. In: *13th IEEE International Conference on Automatic Face*

& Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018, IEEE Computer Society, 2018, pp. 114–121. – URL https://doi.org/10.1109/FG.2018.00026. – ISBN 978-1-5386-2335-0

[Deng et al. 2009] Deng, Jia ; Dong, Wei ; Socher, Richard ; Li, Li-Jia ; Li, Kai ; Li, Fei-Fei : ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. (IEEE Computer Society, 2009a), pp. 248–255. – URL https://doi.org/10.1109/CVPR.2009.5206848. – ISBN 978-1-4244-3992-8

[Derpanis et al. 2012] Derpanis, Konstantinos G. ; Lecce, Matthieu ; Daniilidis, Kostas ; Wildes, Richard P. : Dynamic scene understanding: The role of orientation features in space and time in scene classification. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, IEEE Computer Society, 2012, pp. 1306–1313. – URL https://doi.org/10.1109/CVPR.2012.6247815. – ISBN 978-1-4673-1226-4

[Devlin et al. 2019] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, Jill (Publ.) ; Doran, Christy (Publ.) ; Solorio, Thamar (Publ.): *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 4171–4186. – URL https://doi.org/10.18653/v1/n19-1423. – ISBN 978-1-950737-13-0

[Dhingra and Kunz 2019] Dhingra, Naina ; Kunz, Andreas M. : Res3ATN - Deep 3D Residual Attention Network for Hand Gesture Recognition in Videos. In: *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*, IEEE, 2019, pp. 491–501. – URL https://doi.org/10.1109/3DV.2019.00061. – ISBN 978-1-7281-3131-3

[Dollar et al. 2005] Dollar, P. ; Rabaud, V. ; Cottrell, G. ; Belongie, S. : Behavior recognition via sparse spatio-temporal features. In: *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72

[Donahue et al. 2017] Donahue, Jeff ; Hendricks, Lisa A. ; Rohrbach, Marcus ; Venugopalan, Subhashini ; Guadarrama, Sergio ; Saenko, Kate ; Darrell, Trevor : Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (2017), no. 4, pp. 677–691. – URL https://doi.org/10.1109/TPAMI.2016.2599174

[Donahue et al. 2014] Donahue, Jeff ; Jia, Yangqing ; Vinyals, Oriol ; Hoffman, Judy ; Zhang, Ning ; Tzeng, Eric ; Darrell, Trevor : DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June*

2014 32, JMLR.org, 2014, pp. 647–655. – URL http://proceedings.mlr.press/v32/donahue14.html

[Dosovitskiy et al. 2015] Dosovitskiy, Alexey ; Fischer, Philipp ; Ilg, Eddy ; Häusser, Philip ; Hazirbas, Caner ; Golkov, Vladimir ; Smagt, Patrick van der ; Cremers, Daniel ; Brox, Thomas : FlowNet: Learning Optical Flow with Convolutional Networks. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. (IEEE Computer Society, 2015a), pp. 2758–2766. – URL https://doi.org/10.1109/ICCV.2015.316. – ISBN 978-1-4673-8391-2

[Du et al. 2018] Du, Yang ; Yuan, Chunfeng ; Li, Bing ; Zhao, Lili ; Li, Yangxi ; Hu, Weiming : Interaction-Aware Spatio-Temporal Pyramid Attention Networks for Action Classification. In: *ECCV (16)* 11220, Springer, 2018, pp. 388–404

[Duchenne et al. 2009] Duchenne, Olivier ; Laptev, Ivan ; Sivic, Josef ; Bach, Francis R. ; Ponce, Jean : Automatic annotation of human actions in video. In: *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. (IEEE Computer Society, 2009b), pp. 1491–1498. – URL https://doi.org/10.1109/ICCV.2009.5459279. – ISBN 978-1-4244-4420-5

[Duchi et al. 2011] Duchi, John C. ; Hazan, Elad ; Singer, Yoram : Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In: *J. Mach. Learn. Res.* 12 (2011), pp. 2121–2159. – URL http://dl.acm.org/citation.cfm?id=2021068

[Durand et al. 2017] Durand, Thibaut ; Mordan, Taylor ; Thome, Nicolas ; Cord, Matthieu : WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 5957–5966. – URL https://doi.org/10.1109/CVPR.2017.631. – ISBN 978-1-5386-0457-1

[Ebner and Findling 2019] Ebner, Christopher J. ; Findling, Rainhard D. : Tennis Stroke Classification: Comparing Wrist and Racket as IMU Sensor Position. In: Haghighi, Pari D. (Publ.) ; Salvadori, Ivan L. (Publ.) ; Steinbauer, Matthias (Publ.) ; Khalil, Ismail (Publ.) ; Anderst-Kotsis, Gabriele (Publ.): *MoMM 2019: The 17th International Conference on Advances in Mobile Computing & Multimedia, Munich, Germany, December 2-4, 2019*, ACM, 2019, pp. 74–83. – URL https://doi.org/10.1145/3365921.3365929. – ISBN 978-1-4503-7178-0

[Efros et al. 2003] Efros, Alexei A. ; Berg, Alexander C. ; Mori, Greg ; Malik, Jitendra : Recognizing Action at a Distance. In: *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, IEEE Computer Society, 2003, pp. 726–733. – URL https://doi.org/10.1109/ICCV.2003.1238420. – ISBN 0-7695-1950-4

[Einfalt et al. 2018] Einfalt, Moritz ; Zecha, Dan ; Lienhart, Rainer : Activity-Conditioned Continuous Human Pose Estimation for Performance Analysis of Athletes

Using the Example of Swimming. In: *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. (IEEE Computer Society, 2018b), pp. 446–455. – URL https://doi.org/10.1109/WACV.2018.00055. – ISBN 978-1-5386-4886-5

[Emilien et al. 2013] Emilien, Aurelie ; Benois-Pineau, Jenny ; Elbes, Delphine ; Quesson, Bruno : Adaptive rejection of outliers for robust motion compensation in cardiac MR-thermometry. In: *IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*, IEEE, 2013, pp. 2514–2518. – URL https://doi.org/10.1109/ICIP.2013.6738518. – ISBN 978-1-4799-2341-0

[Ester et al. 1996] Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei : A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Simoudis, Evangelos (Publ.) ; Han, Jiawei (Publ.) ; Fayyad, Usama M. (Publ.): *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, AAAI Press, 1996, pp. 226–231. – URL http://www.aaai.org/Library/KDD/1996/kdd96-037.php. – ISBN 1-57735-004-9

[Everingham et al. 2010] Everingham, Mark ; Gool, Luc V. ; Williams, Christopher K. I. ; Winn, John M. ; Zisserman, Andrew : The Pascal Visual Object Classes (VOC) Challenge. In: *Int. J. Comput. Vis.* 88 (2010), no. 2, pp. 303–338. – URL https://doi.org/10.1007/s11263-009-0275-4

[Fan et al. 2019] Fan, Quanfu ; Chen, Chun-Fu (. ; Kuehne, Hilde ; Pistoia, Marco ; Cox, David D. : More Is Less: Learning Efficient Video Representations by Big-Little Network and Depthwise Temporal Aggregation. In: (Wallach et al., 2019), pp. 2261–2270. – URL http://papers.nips.cc/paper/8498-more-is-less-learning-efficient-video-representations-by-big-little-network-and-depthwise-temporal-aggregation

[Fani et al. 2019] Fani, Mehrnaz ; Vats, Kanav ; Dulhanty, Christopher ; Clausi, David A. ; Zelek, John S. : Pose-Projected Action Recognition Hourglass Network (PARHN) in Soccer. In: *16th Conference on Computer and Robot Vision, CRV 2019, Kingston, ON, Canada, May 29-31, 2019*, IEEE, 2019, pp. 201–208. – URL https://doi.org/10.1109/CRV.2019.00035. – ISBN 978-1-7281-1838-3

[Farnebäck 2003] Farnebäck, Gunnar : Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigün, Josef (Publ.) ; Gustavsson, Tomas (Publ.): *Image Analysis, 13th Scandinavian Conference, SCIA 2003, Halmstad, Sweden, June 29 - July 2, 2003, Proceedings* 2749, Springer, 2003, pp. 363–370. – URL https://doi.org/10.1007/3-540-45103-X_50. – ISBN 3-540-40601-8

[Feichtenhofer et al. 2019] Feichtenhofer, Christoph ; Fan, Haoqi ; Malik, Jitendra ; He, Kaiming : SlowFast Networks for Video Recognition. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019* (IEEE, 2019b), pp. 6201–6210. – URL https://doi.org/10.1109/ICCV.2019.00630. ISBN 978-1-7281-4803-8

[Feichtenhofer et al. 2017] Feichtenhofer, Christoph ; Pinz, Axel ; Wildes, Richard P. : Spatiotemporal Multiplier Networks for Video Action Recognition. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 7445–7454. – URL https://doi.org/10.1109/CVPR.2017.787. – ISBN 978-1-5386-0457-1

[Feichtenhofer et al. 2016] Feichtenhofer, Christoph ; Pinz, Axel ; Zisserman, Andrew : Convolutional Two-Stream Network Fusion for Video Action Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 1933–1941. – URL https://doi.org/10.1109/CVPR.2016.213. – ISBN 978-1-4673-8851-1

[Feng et al. 2019] Feng, Zunlei ; Yu, Zhenyun ; Jing, Yongcheng ; Wu, Sai ; Song, Mingli ; Yang, Yezhou ; Jiang, Junxiao : Interpretable Partitioned Embedding for Intelligent Multi-item Fashion Outfit Composition. In: *ACM Trans. Multim. Comput. Commun. Appl.* 15 (2019), no. 2s, pp. 61:1–61:20. – URL https://doi.org/10.1145/3326332

[Fischler and Bolles 1981] Fischler, Martin A. ; Bolles, Robert C. : Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Commun. ACM* 24 (1981), no. 6, pp. 381–395. – URL http://doi.acm.org/10.1145/358669.358692

[Fong and Vedaldi 2017] Fong, R. C. ; Vedaldi, A. : Interpretable Explanations of Black Boxes by Meaningful Perturbation. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3449–3457

[Fuad et al. 2020] Fuad, Kazi Ahmed A. ; Martin, Pierre-Etienne ; Giot, Romai ; Bourqui, Romain ; Benois-Pineau, Jenny ; Zemmari, Akka : Feature Understanding in 3D CNNs for Actions Recognition in Video. In: *Tenth International Conference on Image Processing Theory, Tools and Applications, IPTA 2020, Paris, France, November 9-12, 2020*, IEEE, 2020, pp. 1–6. – URL https://doi.org/1. – ISBN 978-1-7281-3975-3

[Förstner and Gülch 1987] Förstner, Wolfgang ; Gülch, Eberhard : A Fast Operator for Detection and Precise Location of Distict Point, Corners and Centres of Circular Features. In: *Proceedings of the ISPRS Conference on Fast Processing of Photogrammetric Data, Interlaken 1987*, 1987, pp. 281–305

[Gaidon et al. 2013] Gaidon, Adrien ; Harchaoui, Zaïd ; Schmid, Cordelia : Temporal Localization of Actions with Actoms. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013), no. 11, pp. 2782–2795. – URL https://doi.org/10.1109/TPAMI.2013.65

[Galasso et al. 2013] Galasso, Fabio ; Nagaraja, Naveen S. ; Cardenas, Tatiana J. ; Brox, Thomas ; Schiele, Bernt : A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. (IEEE Computer Society, 2013), pp. 3527–3534. – URL https://doi.org/10.1109/ICCV.2013.438. – ISBN 978-1-4799-2839-2

[Gao et al. 2010] Gao, Zan ; Chen, Ming-yu ; Hauptmann, Alexander G. ; Cai, Anni : Comparing Evaluation Protocols on the KTH Dataset. In: Salah, Albert A. (Publ.) ; Gevers, Theo (Publ.) ; Sebe, Nicu (Publ.) ; Vinciarelli, Alessandro (Publ.): *Human Behavior Understanding, First International Workshop, HBU 2010, Istanbul, Turkey, August 22, 2010. Proceedings* 6219, Springer, 2010, pp. 88–100. – URL https://doi.org/10.1007/978-3-642-14715-9_10. – ISBN 978-3-642-14714-2

[Ghadiyaram et al. 2019a] Ghadiyaram, Deepti ; Tran, Du ; Mahajan, Dhruv : Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. (Computer Vision Foundation / IEEE, 2019), pp. 12046–12055. – URL http://openaccess.thecvf.com/content_CVPR_2019/html/Ghadiyaram_Large-Scale_Weakly-Supervised_Pre-Training_for_Video_Action_Recognition_CVPR_2019_paper.html

[Ghadiyaram et al. 2019b] Ghadiyaram, Deepti ; Tran, Du ; Mahajan, Dhruv : Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. (Computer Vision Foundation / IEEE, 2019), pp. 12046–12055. – URL http://openaccess.thecvf.com/content_CVPR_2019/html/Ghadiyaram_Large-Scale_Weakly-Supervised_Pre-Training_for_Video_Action_Recognition_CVPR_2019_paper.html

[Ghahramani et al. 2014] Ghahramani, Zoubin (Publ.) ; Welling, Max (Publ.) ; Cortes, Corinna (Publ.) ; Lawrence, Neil D. (Publ.) ; Weinberger, Kilian Q. (Publ.) : *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. URL http://papers.nips.cc/book/advances-in-neural-information-processing-systems-27-2014, 2014

[Gillot et al. 2018] Gillot, Pierre ; Benois-Pineau, Jenny ; Zemmari, Akka ; Nesterov, Yurii E. : Increasing Training Stability for Deep CNNS. In: *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*. (IEEE, 2018), pp. 3423–3427. – URL https://doi.org/10.1109/ICIP.2018.8451570. – ISBN 978-1-4799-7061-2

[González-Díaz et al. 2019] González-Díaz, Iván ; Benois-Pineau, Jenny ; Domenger, Jean-Philippe ; Cattaert, Daniel ; Rugy, Aymar de : Perceptually-guided deep neural networks for ego-action prediction: Object grasping. In: *Pattern Recognit.* 88 (2019), pp. 223–235. – URL https://doi.org/10.1016/j.patcog.2018.11.013

[González-Díaz et al. 2018] González-Díaz, Iván ; Benois-Pineau, Jenny ; Domenger, Jean-Philippe ; Rugy, Aymar de : Perceptually-guided Understanding of Egocentric Video Content: Recognition of Objects to Grasp. In: (Aizawa et al., 2018), pp. 434–441. – URL https://doi.org/10.1145/3206025.3206073

[Gorelick et al. 2007] Gorelick, Lena ; Blank, Moshe ; Shechtman, Eli ; Irani, Michal ; Basri, Ronen : Actions as Space-Time Shapes. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007), no. 12, pp. 2247–2253. – URL https://doi.org/10.1109/TPAMI.2007.70711

[Gorelick et al. 2006] Gorelick, Lena ; Galun, Meirav ; Sharon, Eitan ; Basri, Ronen ; Brandt, Achi : Shape Representation and Classification Using the Poisson Equation. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006), no. 12, pp. 1991–2005. – URL https://doi.org/10.1109/TPAMI.2006.253

[Goyal et al. 2017] Goyal, Raghav ; Kahou, Samira E. ; Michalski, Vincent ; Materzynska, Joanna ; Westphal, Susanne ; Kim, Heuna ; Haenel, Valentin ; Fründ, Ingo ; Yianilos, Peter ; Mueller-Freitag, Moritz ; Hoppe, Florian ; Thurau, Christian ; Bax, Ingo ; Memisevic, Roland : The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. (IEEE Computer Society, 2017b), pp. 5843–5851. – URL https://doi.org/10.1109/ICCV.2017.622. – ISBN 978-1-5386-1032-9

[Graves et al. 2013] Graves, Alex ; Jaitly, Navdeep ; Mohamed, Abdel-rahman : Hybrid speech recognition with Deep Bidirectional LSTM. In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, IEEE, 2013, pp. 273–278. – URL https://doi.org/10.1109/ASRU.2013.6707742. – ISBN 978-1-4799-2756-2

[Gu et al. 2018] Gu, Chunhui ; Sun, Chen ; Ross, David A. ; Vondrick, Carl ; Pantofaru, Caroline ; Li, Yeqing ; Vijayanarasimhan, Sudheendra ; Toderici, George ; Ricco, Susanna ; Sukthankar, Rahul ; Schmid, Cordelia ; Malik, Jitendra : AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (IEEE Computer Society, 2018a), pp. 6047–6056. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Gu_AVA_A_Video_CVPR_2018_paper.html

[Guen and Thome 2020] Guen, Vincent L. ; Thome, Nicolas : Disentangling Physical Dynamics From Unknown Factors for Unsupervised Video Prediction. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. (IEEE, 2020), pp. 11471–11481. – URL https://doi.org/10.1109/CVPR42600.2020.01149. – ISBN 978-1-7281-7168-5

[Gurrin et al. 2016] Gurrin, Cathal ; Joho, Hideo ; Hopfgartner, Frank ; Zhou, Liting ; Albatal, Rami : Overview of NTCIR-12 Lifelog Task. In: Kando, Noriko (Publ.) ; Sakai, Tetsuya (Publ.) ; Sanderson, Mark (Publ.): *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*, National Institute of Informatics (NII), 2016. – URL http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings12/pdf/ntcir/OVERVIEW/01-NTCIR12-OV-LIFELOG-GurrinC.pdf

[Gurrin et al. 2019] Gurrin, Cathal (Publ.) ; Jónsson, Björn Pór (Publ.) ; Péteri, Renaud (Publ.) ; Rudinac, Stevan (Publ.) ; Marchand-Maillet, Stéphane (Publ.) ; Quénot, Georges (Publ.) ; McGuinness, Kevin (Publ.) ; Guðmundsson, Gylfi P. (Publ.) ; Little, Suzanne (Publ.) ; Katsurai, Marie (Publ.) ; Healy, Graham (Publ.) : *2019 International Conference on Content-Based Multimedia Indexing, CBMI 2019, Dublin, Ireland, September 4-6, 2019*. IEEE, 2019. – URL https://ieeexplore.ieee.org/xpl/conhome/8863324/proceeding. – ISBN 978-1-7281-4673-7

[Gurrin et al. 2018] Gurrin, Cathal (Publ.) ; Schoeffmann, Klaus (Publ.) ; Joho, Hideo (Publ.) ; Dang-Nguyen, Duc-Tien (Publ.) ; Riegler, Michael (Publ.) ; Piras, Luca (Publ.) : *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge, LSC@ICMR 2018, Yokohama, Japan, June 11, 2018*. ACM, 2018. – URL http://dl.acm.org/citation.cfm?id=3210539

[Hamel et al. 2016] Hamel, Shahrbanoo ; Guyader, Nathalie ; Pellerin, Denis ; Houzet, Dominique : Contribution of color in saliency model for videos. In: *Signal, Image and Video Processing* 10 (2016), no. 3, pp. 423–429. – URL https://doi.org/10.1007/s11760-015-0765-5

[Hanson and Pratt 1988] Hanson, Stephen J. ; Pratt, Lorien Y. : Comparing Biases for Minimal Network Construction with Back-Propagation. In: Touretzky, David S. (Publ.): *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, Morgan Kaufmann, 1988, pp. 177–185. – URL http://papers.nips.cc/paper/156-comparing-biases-for-minimal-network-construction-with-back-propagation. – ISBN 1-55860-015-9

[Harris and Stephens 1988] Harris, Christopher G. ; Stephens, Mike : A Combined Corner and Edge Detector. In: Taylor, Christopher J. (Publ.): *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*, Alvey Vision Club, 1988, pp. 1–6. – URL https://doi.org/10.5244/C.2.23

[Hassan et al. 2019] Hassan, Muneeb U. ; Mulhem, Philippe ; Pellerin, Denis ; Quénot, Georges : Explaining Visual Classification using Attributes. In: (Gurrin et al., 2019), pp. 1–6. – URL https://doi.org/10.1109/CBMI.2019.8877393. – ISBN 978-1-7281-4673-7

[He et al. 2020a] He, Kaiming ; Gkioxari, Georgia ; Dollár, Piotr ; Girshick, Ross B. : Mask R-CNN. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (2020), no. 2, pp. 386–397. – URL https://doi.org/10.1109/TPAMI.2018.2844175

[He et al. 2016] He, Kaiming ; Zhang, Xiangyu ; Ren, Shaoqing ; Sun, Jian : Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (IEEE Computer Society, 2016), pp. 770–778. – URL https://doi.org/10.1109/CVPR.2016.90. ISBN 978-1-4673-8851-1

[He et al. 2020b] He, Pengcheng ; Liu, Xiaodong ; Gao, Jianfeng ; Chen, Weizhu : DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In: *CoRR* abs/2006.03654 (2020). – URL https://arxiv.org/abs/2006.03654

[Heilbron et al. 2015] Heilbron, Fabian C. ; Escorcia, Victor ; Ghanem, Bernard ; Niebles, Juan C. : ActivityNet: A large-scale video benchmark for human activity understanding. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. (IEEE Computer Society, 2015b), pp. 961–970. – URL https://doi.org/10.1109/CVPR.2015.7298698. – ISBN 978-1-4673-6964-0

[Heilbron and Niebles 2014] Heilbron, Fabian C. ; Niebles, Juan C. : Collecting and Annotating Human Activities in Web Videos. In: Kankanhalli, Mohan S. (Publ.) ; Rueger, Stefan (Publ.) ; Manmatha, R. (Publ.) ; Jose, Joemon M. (Publ.) ; Rijsbergen, Keith van (Publ.): *International Conference on Multimedia Retrieval, ICMR '14, Glasgow, United Kingdom - April 01 - 04, 2014*, ACM, 2014, pp. 377. – URL https://doi.org/10.1145/2578726.2578775. – ISBN 978-1-4503-2782-4

[Hinton et al. 2012] Hinton, Geoffrey E. ; Srivastava, Nitish ; Krizhevsky, Alex ; Sutskever, Ilya ; Salakhutdinov, Ruslan : Improving neural networks by preventing co-adaptation of feature detectors. In: *CoRR* abs/1207.0580 (2012). – URL http://arxiv.org/abs/1207.0580

[Hohman et al. 2019] Hohman, Fred ; Kahng, Minsuk ; Pienta, Robert ; Chau, Duen H. : Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. In: *IEEE Trans. Vis. Comput. Graph.* 25 (2019), no. 8, pp. 2674–2693

[Hooker et al. 2019] Hooker, Sara ; Erhan, Dumitru ; Kindermans, Pieter-Jan ; Kim, Been : A Benchmark for Interpretability Methods in Deep Neural Networks. In: (Wallach et al., 2019), pp. 9734–9745. – URL http://papers.nips.cc/paper/9167-a-benchmark-for-interpretability-methods-in-deep-neural-networks

[Horn and Schunck 1981] Horn, Berthold K. P. ; Schunck, Brian G. : Determining Optical Flow. In: *Artif. Intell.* 17 (1981), no. 1-3, pp. 185–203. – URL https://doi.org/10.1016/0004-3702(81)90024-2

[Hou et al. 2017] Hou, Rui ; Chen, Chen ; Shah, Mubarak : Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. (IEEE Computer Society, 2017b), pp. 5823–5832. – URL https://doi.org/10.1109/ICCV.2017.620. – ISBN 978-1-5386-1032-9

[Hou et al. 2019] Hou, Rui ; Chen, Chen ; Sukthankar, Rahul ; Shah, Mubarak : An Efficient 3D CNN for Action/Object Segmentation in Video. In: *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, BMVA Press, 2019, pp. 170. – URL https://bmvc2019.org/wp-content/uploads/papers/0162-paper.pdf

[Hu et al. 2020] Hu, Jie ; Shen, Li ; Albanie, Samuel ; Sun, Gang ; Wu, Enhua : Squeeze-and-Excitation Networks. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (2020), no. 8, pp. 2011–2023. – URL https://doi.org/10.1109/TPAMI.2019.2913372

[Huang et al. 2018] Huang, Feiran ; Zhang, Xiaoming ; Li, Chaozhuo ; Li, Zhoujun ; He, Yueying ; Zhao, Zhonghua : Multimodal Network Embedding via Attention based Multi-view Variational Autoencoder. In: *ICMR*, ACM, 2018, pp. 108–116

[Huang et al. 2017] Huang, Gao ; Liu, Zhuang ; Maaten, Laurens van der ; Weinberger, Kilian Q. : Densely Connected Convolutional Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 2261–2269. – URL https://doi.org/10.1109/CVPR.2017.243. – ISBN 978-1-5386-0457-1

[IEEE 2018] : *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*. IEEE, 2018. – URL https://ieeexplore.ieee.org/xpl/conhome/8436606/proceeding. – ISBN 978-1-4799-7061-2

[IEEE 2019a] : *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. IEEE, 2019. – URL https://ieeexplore.ieee.org/xpl/conhome/8791230/proceeding. – ISBN 978-1-5386-6249-6

[IEEE 2019b] : *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019. – URL https://ieeexplore.ieee.org/xpl/conhome/8972782/proceeding. – ISBN 978-1-7281-4803-8

[IEEE 2019c] : *Ninth International Conference on Image Processing Theory, Tools and Applications, IPTA 2019, Istanbul, Turkey, November 6-9, 2019*. IEEE, 2019. – URL https://ieeexplore.ieee.org/xpl/conhome/8932637/proceeding. – ISBN 978-1-7281-3975-3

[IEEE 2020] : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020. – URL https://ieeexplore.ieee.org/xpl/conhome/9142308/proceeding. – ISBN 978-1-7281-7168-5

[IEEE Computer Society 2005] : *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 2005. – URL https://ieeexplore.ieee.org/xpl/conhome/9901/proceeding. – ISBN 0-7695-2372-2

[IEEE Computer Society 2008] : *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008. – URL https://ieeexplore.ieee.org/xpl/conhome/4558014/proceeding. – ISBN 978-1-4244-2242-5

[IEEE Computer Society 2009a] : *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009. – URL https://ieeexplore.ieee.org/xpl/conhome/5191365/proceeding. – ISBN 978-1-4244-3992-8

[IEEE Computer Society 2009b] : *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. IEEE Computer Society, 2009. – URL https://ieeexplore.ieee.org/xpl/conhome/5453389/proceeding. – ISBN 978-1-4244-4420-5

[IEEE Computer Society 2010] : *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE Computer Society, 2010. – URL https://ieeexplore.ieee.org/xpl/conhome/5521876/proceeding. – ISBN 978-1-4244-6984-0

[IEEE Computer Society 2011] : *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 2011. – URL https://ieeexplore.ieee.org/xpl/conhome/5968010/proceeding. – ISBN 978-1-4577-0394-2

[IEEE Computer Society 2013] : *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. IEEE Computer Society, 2013. – URL https://ieeexplore.ieee.org/xpl/conhome/6750807/proceeding. – ISBN 978-1-4799-2839-2

[IEEE Computer Society 2014] : *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014. – URL https://ieeexplore.ieee.org/xpl/conhome/6909096/proceeding. – ISBN 978-1-4799-5118-5

[IEEE Computer Society 2015a] : *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015. – URL https://ieeexplore.ieee.org/xpl/conhome/7407725/proceeding. – ISBN 978-1-4673-8391-2

[IEEE Computer Society 2015b] : *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015. – URL https://ieeexplore.ieee.org/xpl/conhome/7293313/proceeding. – ISBN 978-1-4673-6964-0

[IEEE Computer Society 2016] : *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016. – URL https://ieeexplore.ieee.org/xpl/conhome/7776647/proceeding. – ISBN 978-1-4673-8851-1

[IEEE Computer Society 2017a] : *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017. – URL https://ieeexplore.ieee.org/xpl/conhome/8097368/proceeding. – ISBN 978-1-5386-0457-1

[IEEE Computer Society 2017b] : *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017. – URL https://ieeexplore.ieee.org/xpl/conhome/8234942/proceeding. – ISBN 978-1-5386-1032-9

[IEEE Computer Society 2018a] : *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018. – URL https://ieeexplore.ieee.org/xpl/conhome/8576498/proceeding

[IEEE Computer Society 2018b] : *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 2018. – URL https://ieeexplore.ieee.org/xpl/conhome/8345804/proceeding. – ISBN 978-1-5386-4886-5

[IEEE Computer Society 2021] : *25th International Conference on Pattern Recognition (ICPR2020) - MiCo Milano Congress Center, Italy, 10-15 January 2021*. IEEE Computer Society, 2021

[Ilg et al. 2017] Ilg, Eddy ; Mayer, Nikolaus ; Saikia, Tonmoy ; Keuper, Margret ; Dosovitskiy, Alexey ; Brox, Thomas : FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 1647–1655. – URL https://doi.org/10.1109/CVPR.2017.179. – ISBN 978-1-5386-0457-1

[Ioffe and Szegedy 2015] Ioffe, Sergey ; Szegedy, Christian : Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *ICML* 37, JMLR.org, 2015, pp. 448–456

[Ionescu et al. 2014] Ionescu, Bogdan (Publ.) ; Benois-Pineau, Jenny (Publ.) ; Piatrik, Tomas (Publ.) ; Quénot, Georges (Publ.) : *Fusion in Computer Vision - Understanding Complex Visual Content*. Springer, 2014 (Advances in Computer Vision and Pattern Recognition). – URL https://doi.org/10.1007/978-3-319-05696-8. – ISBN 978-3-319-05695-1

[Iriguchi et al. 2018] Iriguchi, Mayuko ; Koda, Hiroki ; Masataka, Nobuo : Colour Perception Characteristics of Women in Menopause. In: (Aizawa et al., 2018), pp. 20–25. – URL https://doi.org/10.1145/3209693.3209694

[Jaakkola and Haussler 1998] Jaakkola, Tommi S. ; Haussler, David : Exploiting Generative Models in Discriminative Classifiers. In: Kearns, Michael J. (Publ.) ; Solla, Sara A. (Publ.) ; Cohn, David A. (Publ.): *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998]*, The MIT Press, 1998, pp. 487–493. – URL http://papers.nips.cc/paper/1520-exploiting-generative-models-in-discriminative-classifiers. – ISBN 0-262-11245-0

[Jain et al. 2014] Jain, Mihir ; Gemert, Jan C. van ; Jégou, Hervé ; Bouthemy, Patrick ; Snoek, Cees G. M. : Action Localization with Tubelets from Motion. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. (IEEE Computer Society, 2014), pp. 740–747. – URL https://doi.org/10.1109/CVPR.2014.100. – ISBN 978-1-4799-5118-5

[Jain et al. 2017] Jain, Mihir ; Gemert, Jan C. van ; Jégou, Hervé ; Bouthemy, Patrick ; Snoek, Cees G. M. : Tubelets: Unsupervised Action Proposals from Spatiotemporal Super-Voxels. In: *Int. J. Comput. Vis.* 124 (2017), no. 3, pp. 287–311. – URL https://doi.org/10.1007/s11263-017-1023-9

[Jhuang et al. 2013] Jhuang, Hueihan ; Gall, Juergen ; Zuffi, Silvia ; Schmid, Cordelia ; Black, Michael J. : Towards Understanding Action Recognition. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. (IEEE Computer Society, 2013), pp. 3192–3199. – URL https://doi.org/10.1109/ICCV.2013.396. – ISBN 978-1-4799-2839-2

[Ji et al. 2010] Ji, Shuiwang ; Xu, Wei ; Yang, Ming ; Yu, Kai : 3D Convolutional Neural Networks for Human Action Recognition. In: Fürnkranz, Johannes (Publ.) ; Joachims, Thorsten (Publ.): *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, Omnipress, 2010, pp. 495–502. – URL https://icml.cc/Conferences/2010/papers/100.pdf

[Ji et al. 2013] Ji, Shuiwang ; Xu, Wei ; Yang, Ming ; Yu, Kai : 3D Convolutional Neural Networks for Human Action Recognition. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013), no. 1, pp. 221–231. – URL https://doi.org/10.1109/TPAMI.2012.59

[Jiang et al. 2018] Jiang, Yangbangyan ; Xu, Qianqian ; Cao, Xiaochun ; Huang, Qingming : Who to Ask: An Intelligent Fashion Consultant. In: (Aizawa et al., 2018), pp. 525–528. – URL https://doi.org/10.1145/3206025.3206092

[Jiang et al. 2017] Jiang, Zhuolin ; Rozgic, Viktor ; Adali, Sancar : Learning Spatiotemporal Features for Infrared Action Recognition with 3D Convolutional Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 309–317. – URL https://doi.org/10.1109/CVPRW.2017.44. – ISBN 978-1-5386-0733-6

[Jing et al. 2019] Jing, Yuan ; Hao, Jinshan ; Li, Peng : Learning Spatiotemporal Features of CSI for Indoor Localization With Dual-Stream 3D Convolutional Neural Networks. In: *IEEE Access* 7 (2019), pp. 147571–147585. – URL https://doi.org/10.1109/ACCESS.2019.2946870

[Jolliffe 1986] Jolliffe, Ian T. : *Principal Component Analysis*. Springer, 1986 (Springer Series in Statistics). – URL https://doi.org/10.1007/978-1-4757-1904-8. – ISBN 978-1-4757-1906-2

[Judd et al. 2009] Judd, Tilke ; Ehinger, Krista A. ; Durand, Frédo ; Torralba, Antonio : Learning to predict where humans look. In: *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. (IEEE Computer Society, 2009b), pp. 2106–2113. – URL https://doi.org/10.1109/ICCV.2009.5459462. – ISBN 978-1-4244-4420-5

[Kalfaoglu et al. 2020] Kalfaoglu, M. E. ; Kalkan, Sinan ; Alatan, A. A. : Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. In: *CoRR* abs/2008.01232 (2020). – URL https://arxiv.org/abs/2008.01232

[Kanade et al. 2000] Kanade, Takeo ; Tian, Ying-li ; Cohn, Jeffrey F. : Comprehensive Database for Facial Expression Analysis. In: *4th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000), 26-30 March 2000, Grenoble, France*, IEEE Computer Society, 2000, pp. 46–53. – URL https://doi.org/10.1109/AFGR.2000.840611. – ISBN 0-7695-0580-5

[Karpathy et al. 2014] Karpathy, Andrej ; Toderici, George ; Shetty, Sanketh ; Leung, Thomas ; Sukthankar, Rahul ; Li, Fei-Fei : Large-Scale Video Classification with Convolutional Neural Networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. (IEEE Computer Society, 2014), pp. 1725–1732. – URL https://doi.org/10.1109/CVPR.2014.223. – ISBN 978-1-4799-5118-5

[Kay et al. 2017] Kay, Will ; Carreira, João ; Simonyan, Karen ; Zhang, Brian ; Hillier, Chloe ; Vijayanarasimhan, Sudheendra ; Viola, Fabio ; Green, Tim ; Back, Trevor ; Natsev, Paul ; Suleyman, Mustafa ; Zisserman, Andrew : The Kinetics Human Action Video Dataset. In: *CoRR* abs/1705.06950 (2017). – URL http://arxiv.org/abs/1705.06950

[Kenwright 2015] Kenwright, Ben : Free-Form Tetrahedron Deformation. In: *ISVC (2)* 9475, Springer, 2015, pp. 787–796

[Kern 2007] Kern, Robert : *NEP 1 — A Simple File Format for NumPy Arrays*, 2007. – URL https://numpy.org/neps/nep-0001-npy-format.html. – Accessed: 2020-08-01

[Khosla et al. 2012] Khosla, Aditya ; Zhou, Tinghui ; Malisiewicz, Tomasz ; Efros, Alexei A. ; Torralba, Antonio : Undoing the Damage of Dataset Bias. In: Fitzgibbon, Andrew W. (Publ.) ; Lazebnik, Svetlana (Publ.) ; Perona, Pietro (Publ.) ; Sato, Yoichi (Publ.) ; Schmid, Cordelia (Publ.): *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I* 7572, Springer, 2012, pp. 158–171. – URL https://doi.org/10.1007/978-3-642-33718-5_12. – ISBN 978-3-642-33717-8

[Kijak et al. 2006] Kijak, Ewa ; Gravier, Guillaume ; Oisel, Lionel ; Gros, Patrick : Audiovisual integration for tennis broadcast structuring. In: *Multim. Tools Appl.* 30 (2006), no. 3, pp. 289–311. – URL https://doi.org/10.1007/s11042-006-0031-5

[Kim et al. 2007] Kim, Ho J. ; Lee, Joseph S. ; Yang, Hyun S. : Human Action Recognition Using a Modified Convolutional Neural Network. In: Liu, Derong (Publ.) ; Fei, Shumin (Publ.) ; Hou, Zeng-Guang (Publ.) ; Zhang, Huaguang (Publ.) ; Sun, Changyin (Publ.): *Advances in Neural Networks - ISNN 2007, 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007, Proceedings, Part II* 4492, Springer, 2007, pp. 715–723. – URL https://doi.org/10.1007/978-3-540-72393-6_85. – ISBN 978-3-540-72392-9

[Kläser et al. 2008] Kläser, Alexander ; Marszalek, Marcin ; Schmid, Cordelia : A Spatio-Temporal Descriptor Based on 3D-Gradients. In: Everingham, Mark (Publ.) ; Needham, Chris J. (Publ.) ; Fraile, Roberto (Publ.): *Proceedings of the British Machine Vision Conference 2008, Leeds, UK, September 2008*, British Machine Vision Association, 2008, pp. 1–10. – URL https://doi.org/10.5244/C.22.99. – ISBN 978-1-901725-36-0

[Koch et al. 2018] Koch, David ; Despotovic, Miroslav ; Sakeena, Muntaha ; Döller, Mario ; Zeppelzauer, Matthias : Visual Estimation of Building Condition with Patch-level ConvNets. In: Kiyota, Yoji (Publ.) ; Yamasaki, Toshihiko (Publ.) ; Shimizu, Chihiro (Publ.) ; Suwa, Hirohiko (Publ.) ; Arakawa, Yutaka (Publ.) ; Kitagaki, Ryoma (Publ.) ; Nomura, Shimpei (Publ.): *Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech, RETech@ICMR 2018, Yokohama, Japan, June 11, 2018*, ACM, 2018, pp. 12–17. – URL https://doi.org/10.1145/3210499.3210526

[Krapac et al. 2011] Krapac, Josip ; Verbeek, Jakob J. ; Jurie, Frédéric : Modeling spatial layout with fisher vectors for image categorization. In: (Metaxas et al., 2011), pp. 1487–1494. – URL https://doi.org/10.1109/ICCV.2011.6126406. – ISBN 978-1-4577-1101-5

[Krause et al. 2018] Krause, Jonas ; Sugita, Gavin ; Baek, Kyungim ; Lim, Lipyeow : *WTPlant* (What's That Plant?): A Deep Learning System for Identifying Plants in Natural Images. In: (Aizawa et al., 2018), pp. 517–520. – URL https://doi.org/10.1145/3206025.3206089

[Krizhevsky 2009] Krizhevsky, Alex : *Learning Multiple Layers of Features from Tiny Images*. April 2009

[Krizhevsky et al. 2012] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E. : ImageNet Classification with Deep Convolutional Neural Networks. In: Bartlett, Peter L. (Publ.) ; Pereira, Fernando C. N. (Publ.) ; Burges, Christopher J. C. (Publ.) ; Bottou, Léon (Publ.) ; Weinberger, Kilian Q. (Publ.): *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks, 2012, pp. 1106–1114

[Krizhevsky et al. 2017] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E. : ImageNet classification with deep convolutional neural networks. In: *Commun. ACM* 60 (2017), no. 6, pp. 84–90. – URL http://doi.acm.org/10.1145/3065386

[Kroeger et al. 2016] Kroeger, Till ; Timofte, Radu ; Dai, Dengxin ; Gool, Luc V. : Fast Optical Flow Using Dense Inverse Search. In: Leibe, Bastian (Publ.) ; Matas, Jiri (Publ.) ; Sebe, Nicu (Publ.) ; Welling, Max (Publ.): *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV* 9908, Springer, 2016, pp. 471–488. – URL https://doi.org/10.1007/978-3-319-46493-0_29. – ISBN 978-3-319-46492-3

[Kuehne et al. 2011] Kuehne, Hildegard ; Jhuang, Hueihan ; Garrote, Estíbaliz ; Poggio, Tomaso A. ; Serre, Thomas : HMDB: A large video database for human motion recognition. In: (Metaxas et al., 2011), pp. 2556–2563. – URL https://doi.org/10.1109/ICCV.2011.6126543. – ISBN 978-1-4577-1101-5

[Kuzovkin et al. 2018] Kuzovkin, Dmitry ; Pouli, Tania ; Cozot, Rémi ; Meur, Olivier L. ; Kervec, Jonathan ; Bouatouch, Kadi : Image Selection in Photo Albums. In: (Aizawa et al., 2018), pp. 397–404. – URL https://doi.org/10.1145/3206025.3206077

[Kwolek and Kepski 2014] Kwolek, Bogdan ; Kepski, Michal : Human fall detection on embedded platform using depth maps and wireless accelerometer. In: *Comput. Methods Programs Biomed.* 117 (2014), no. 3, pp. 489–501. – URL https://doi.org/10.1016/j.cmpb.2014.09.005

[Laptev 2005] Laptev, Ivan : On Space-Time Interest Points. In: *Int. J. Comput. Vis.* 64 (2005), no. 2-3, pp. 107–123. – URL https://doi.org/10.1007/s11263-005-1838-7

[Laptev and Pérez 2007] Laptev, Ivan ; Pérez, Patrick : Retrieving actions in movies. In: *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, IEEE Computer Society, 2007, pp. 1–8. – URL https://doi.org/10.1109/ICCV.2007.4409105. – ISBN 978-1-4244-1630-1

[Larson et al. 2020] Larson, Martha A. (Publ.) ; Hicks, Steven A. (Publ.) ; Constantin, Mihai G. (Publ.) ; Bischke, Benjamin (Publ.) ; Porter, Alastair (Publ.) ; Zhao, Peijian (Publ.) ; Lux, Mathias (Publ.) ; Quiros, Laura C. (Publ.) ; Calandre, Jordan (Publ.) ; Jones, Gareth (Publ.) : *Working Notes Proceedings of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-30 October 2019. 2670*. CEUR-WS.org, 2020. (CEUR Workshop Proceedings). – URL http://ceur-ws.org/Vol-2670

[LeCun et al. 1989] LeCun, Yann ; Boser, Bernhard E. ; Denker, John S. ; Henderson, Donnie ; Howard, Richard E. ; Hubbard, Wayne E. ; Jackel, Lawrence D. : Backpropagation Applied to Handwritten Zip Code Recognition. In: *Neural Comput.* 1 (1989), no. 4, pp. 541–551. – URL https://doi.org/10.1162/neco.1989.1.4.541

[Lee and Jung 2020] Lee, Jinkue ; Jung, Hoeryong : TUHAD: Taekwondo Unit Technique Human Action Dataset with Key Frame-Based CNN Action Recognition. In: *Sensors* 20 (2020), no. 17, pp. 4871

[Lei et al. 2019] Lei, Jianjun ; Jia, Yalong ; Peng, Bo ; Huang, Qingming : Channel-wise Temporal Attention Network for Video Action Recognition. In: *IEEE International*

Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019, IEEE, 2019, pp. 562–567. – URL https://doi.org/10.1109/ICME.2019.00103. – ISBN 978-1-5386-9552-4

[Leibe et al. 2016] Leibe, Bastian (Publ.) ; Matas, Jiri (Publ.) ; Sebe, Nicu (Publ.) ; Welling, Max (Publ.) : *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII. 9912*. Springer, 2016. (Lecture Notes in Computer Science). – URL https://doi.org/10.1007/978-3-319-46484-8. – ISBN 978-3-319-46483-1

[Leibetseder and Schoeffmann 2018] Leibetseder, Andreas ; Schoeffmann, Klaus : Extracting and Using Medical Expert Knowledge to Advance in Video Processing for Gynecologic Endoscopy. In: (Aizawa et al., 2018), pp. 485–488. – URL https://doi.org/10.1145/3206025.3206082

[Li et al. 2020] Li, Ang ; Thotakuri, Meghana ; Ross, David A. ; Carreira, João ; Vostrikov, Alexander ; Zisserman, Andrew : The AVA-Kinetics Localized Human Actions Video Dataset. In: *CoRR* abs/2005.00214 (2020). – URL https://arxiv.org/abs/2005.00214

[Li et al. 2019a] Li, Cheng ; Zhao, Yuming ; Peng, Shihao ; Chen, Jinting : Bidirectional Single-Stream Temporal Sentence Query Localization in Untrimmed Videos. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 270–274. – URL https://doi.org/10.1109/ICIP.2019.8802929. – ISBN 978-1-5386-6249-6

[Li et al. 2019b] Li, Heyi ; Tian, Yunke ; Mueller, Klaus ; Chen, Xin : Beyond saliency: Understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. In: *Image Vis. Comput.* 83-84 (2019), pp. 70–86

[Li et al. 2018] Li, Yingwei ; Li, Yi ; Vasconcelos, Nuno : RESOUND: Towards Action Recognition Without Representation Bias. In: Ferrari, Vittorio (Publ.) ; Hebert, Martial (Publ.) ; Sminchisescu, Cristian (Publ.) ; Weiss, Yair (Publ.): *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI* 11210, Springer, 2018, pp. 520–535. – URL https://doi.org/10.1007/978-3-030-01231-1_32. – ISBN 978-3-030-01230-4

[Li et al. 2016] Li, Zhihao ; Wang, Wenmin ; Li, Nannan ; Wang, Jinzhuo : Tube ConvNets: Better exploiting motion for action recognition. In: *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, IEEE, 2016, pp. 3056–3060. – URL https://doi.org/10.1109/ICIP.2016.7532921. – ISBN 978-1-4673-9961-6

[Li and Yu 2018] Li, Zhiwei ; Yu, Lei : Compare Stereo Patches Using Atrous Convolutional Neural Networks. In: (Aizawa et al., 2018), pp. 473–480. – URL https://doi.org/10.1145/3206025.3206075

[Lima et al. 2017] Lima, Tiago ; Fernandes, Bruno J. T. ; Barros, Pablo V. A. : Human action recognition with 3D convolutional neural network. In: *IEEE Latin American Conference on Computational Intelligence, LA-CCI 2017, Arequipa, Peru, November 8-10, 2017*, IEEE, 2017, pp. 1–6. – URL https://doi.org/10.1109/LA-CCI.2017.8285700. – ISBN 978-1-5386-3734-0

[Lin et al. 2020] Lin, Hsien-I ; Yu, Zhangguo ; Huang, Yi-Chen : Ball Tracking and Trajectory Prediction for Table-Tennis Robots. In: *Sensors* 20 (2020), no. 2, pp. 333. – URL https://doi.org/10.3390/s20020333

[Lin et al. 2019] Lin, Ji ; Gan, Chuang ; Han, Song : TSM: Temporal Shift Module for Efficient Video Understanding. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. (IEEE, 2019b), pp. 7082–7092. – URL https://doi.org/10.1109/ICCV.2019.00718. – ISBN 978-1-7281-4803-8

[Liu 2009] Liu, Ce : *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*, Massachusetts Institute of Technology, Dissertation, 5 2009

[Liu et al. 2009] Liu, Jingen ; Luo, Jiebo ; Shah, Mubarak : Recognizing realistic actions from videos. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. (IEEE Computer Society, 2009a), pp. 1996–2003. – URL https://doi.org/10.1109/CVPR.2009.5206744. – ISBN 978-1-4244-3992-8

[Liu and Shah 2008] Liu, Jingen ; Shah, Mubarak : Learning human actions via information maximization. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. (IEEE Computer Society, 2008). – URL https://doi.org/10.1109/CVPR.2008.4587723. – ISBN 978-1-4244-2242-5

[Liu et al. 2017] Liu, Jun ; Wang, Gang ; Hu, Ping ; Duan, Ling-Yu ; Kot, Alex C. : Global Context-Aware Attention LSTM Networks for 3D Action Recognition. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 3671–3680. – URL https://doi.org/10.1109/CVPR.2017.391. – ISBN 978-1-5386-0457-1

[Liu et al. 2019a] Liu, Min ; Shi, Yifei ; Zheng, Lintao ; Xu, Kai ; Huang, Hui ; Manocha, Dinesh : Recurrent 3D attentional networks for end-to-end active object recognition. In: *Comput. Vis. Media* 5 (2019), no. 1, pp. 91–104. – URL https://doi.org/10.1007/s41095-019-0135-2

[Liu et al. 2019b] Liu, Ruichen ; Wang, Zhelong ; Shi, Xin ; Zhao, Hongyu ; Qiu, Sen ; Li, Jie ; Yang, Ning : Table Tennis Stroke Recognition Based on Body Sensor Network. In: Montella, Raffaele (Publ.) ; Ciaramella, Angelo (Publ.) ; Fortino, Giancarlo (Publ.) ; Guerrieri, Antonio (Publ.) ; Liotta, Antonio (Publ.): *Internet and Distributed Computing Systems - 12th International Conference, IDCS 2019, Naples, Italy, October 10-12, 2019, Proceedings* 11874, Springer, 2019, pp. 1–10. – URL https://doi.org/10.1007/978-3-030-34914-1_1. – ISBN 978-3-030-34913-4

[Liu and Hu 2019] Liu, Zheng ; Hu, Haifeng : Spatiotemporal Relation Networks for Video Action Recognition. In: *IEEE Access* 7 (2019), pp. 14969–14976. – URL https://doi.org/10.1109/ACCESS.2019.2894025

[Long et al. 2018] Long, Xiang ; Gan, Chuang ; Melo, Gerard de ; Wu, Jiajun ; Liu, Xiao ; Wen, Shilei : Attention Clusters: Purely Attention Based Local Feature Integration for Video Classification. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. (IEEE Computer Society, 2018a), pp. 7834–7843. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Long_Attention_Clusters_Purely_CVPR_2018_paper.html

[Loshchilov and Hutter 2017] Loshchilov, Ilya ; Hutter, Frank : SGDR: Stochastic Gradient Descent with Warm Restarts. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. (OpenReview.net, 2017). – URL https://openreview.net/forum?id=Skq89Scxx

[Lowe 2004] Lowe, David G. : Distinctive Image Features from Scale-Invariant Keypoints. In: *Int. J. Comput. Vis.* 60 (2004), no. 2, pp. 91–110. – URL https://doi.org/10.1023/B:VISI.0000029664.99615.94

[Lu et al. 2019] Lu, Na ; Wu, Yidan ; Feng, Li ; Song, Jinbo : Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data. In: *IEEE J. Biomed. Health Informatics* 23 (2019), no. 1, pp. 314–323. – URL https://doi.org/10.1109/JBHI.2018.2808281

[Lucas and Kanade 1981] Lucas, Bruce D. ; Kanade, Takeo : An Iterative Image Registration Technique with an Application to Stereo Vision. In: Hayes, Patrick J. (Publ.): *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981*, William Kaufmann, 1981, pp. 674–679. – URL http://ijcai.org/proceedings/1981-1

[Luo and Yuille 2019] Luo, Chenxu ; Yuille, Alan L. : Grouped Spatial-Temporal Aggregation for Efficient Action Recognition. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. (IEEE, 2019b), pp. 5511–5520. – URL https://doi.org/10.1109/ICCV.2019.00561. – ISBN 978-1-7281-4803-8

[Luo et al. 2016] Luo, Wenjie ; Li, Yujia ; Urtasun, Raquel ; Zemel, Richard S. : Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In: *NIPS*, 2016, pp. 4898–4906

[Luvizon et al. 2018] Luvizon, Diogo C. ; Picard, David ; Tabia, Hedi : 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. (IEEE Computer Society, 2018a), pp. 5137–5146. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Luvizon_2D3D_Pose_Estimation_CVPR_2018_paper.html

[Ma et al. 2014] Ma, Xin ; Wang, Haibo ; Xue, Bingxia ; Zhou, Mingang ; Ji, Bing ; Li, Yibin : Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine. In: *IEEE J. Biomed. Health Informatics* 18 (2014), no. 6, pp. 1915–1922. – URL https://doi.org/10.1109/JBHI.2014.2304357

[Mahdisoltani et al. 2018] Mahdisoltani, Farzaneh ; Berger, Guillaume ; Gharbieh, Waseem ; Fleet, David J. ; Memisevic, Roland : Fine-grained Video Classification and Captioning. In: *CoRR* abs/1804.09235 (2018). – URL http://arxiv.org/abs/1804.09235

[Manerba et al. 2008] Manerba, Francesca ; Benois-Pineau, Jenny ; Leonardi, Riccardo ; Mansencal, Boris : Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain. In: *EURASIP J. Adv. Sig. Proc.* 2008 (2008)

[Marcelino et al. 2018] Marcelino, Gonçalo ; Pinto, Ricardo ; Magalhães, João : Ranking News-Quality Multimedia. In: (Aizawa et al., 2018), pp. 10–18. – URL https://doi.org/10.1145/3206025.3206053

[Marszalek et al. 2009] Marszalek, Marcin ; Laptev, Ivan ; Schmid, Cordelia : Actions in context. In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. (IEEE Computer Society, 2009a), pp. 2929–2936. – URL https://doi.org/10.1109/CVPR.2009.5206557. – ISBN 978-1-4244-3992-8

[Martin et al. 2019a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Mascarilla, Laurent ; Calandre, Jordan ; Morlier, Julien : Sports Video Annotation: Detection of Strokes in Table Tennis Task for MediaEval 2019. In: (Larson et al., 2020). – URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_6.pdf

[Martin et al. 2020a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Mascarilla, Laurent ; Calandre, Jordan ; Morlier, Julien : Sports Video Classification: Classification of Strokes in Table Tennis for MediaEval 2020. In: *Proc. of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, 2020

[Martin et al. 2019b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Morlier, Julien : Siamese Spatio-Temporal Convolutional Neural Network for Stroke Classification in Table Tennis Games. In: (Larson et al., 2020). – URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_58.pdf

[Martin et al. 2020b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Mansencal, Boris ; Péteri, Renaud ; Morlier, Julien : Classification of Strokes in Table Tennis with a Three Stream Spatio-Temporal CNN for MediaEval 2020. In: *Proc. of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, 2020

[Martin et al. 2019c] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud : Fine-Grained Action Detection and Classification in Table Tennis with Siamese Spatio-Temporal Convolutional Neural Network. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 3027–3028. – URL https://doi.org/10.1109/ICIP.2019.8803382. – ISBN 978-1-5386-6249-6

[Martin et al. 2018] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis. In: *2018 International Conference on Content-Based Multimedia Indexing, CBMI 2018, La Rochelle, France, September 4-6, 2018*, IEEE, 2018, pp. 1–6. – URL https://doi.org/10.1109/CBMI.2018.8516488. – ISBN 978-1-5386-7021-7

[Martin et al. 2019d] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Optimal Choice of Motion Estimation Methods for Fine-Grained Action Classification with 3D Convolutional Networks. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 554–558. – URL https://doi.org/10.1109/ICIP.2019.8803780. – ISBN 978-1-5386-6249-6

[Martin et al. 2020c] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks. In: *Multim. Tools Appl.* 79 (2020), no. 27-28, pp. 20429–20447. – URL https://doi.org/10.1007/s11042-020-08917-3

[Martin et al. 2021a] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Morlier, Julien : 3D attention mechanisms in Twin Spatio-Temporal Convolutional Neural Networks. Application to action classification in videos of table tennis games. In: *25th International Conference on Pattern Recognition (ICPR2020) - MiCo Milano Congress Center, Italy, 10-15 January 2021*. (IEEE Computer Society, 2021)

[Martin et al. 2021b] Martin, Pierre-Etienne ; Benois-Pineau, Jenny ; Péteri, Renaud ; Zemmari, Akka : Multi-faceted Deep Learning:Models and Data. In: *Springer* (2021)

[MathWorks] MathWorks (Organizer) : *LSTM*. – URL https://fr.mathworks.com/discovery/lstm.html. – Accessed: 2020-11-03

[Mayer et al. 2016] Mayer, Nikolaus ; Ilg, Eddy ; Häusser, Philip ; Fischer, Philipp ; Cremers, Daniel ; Dosovitskiy, Alexey ; Brox, Thomas : A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 4040–4048. – URL https://doi.org/10.1109/CVPR.2016.438. – ISBN 978-1-4673-8851-1

[Metaxas et al. 2011] Metaxas, Dimitris N. (Publ.) ; Quan, Long (Publ.) ; Sanfeliu, Alberto (Publ.) ; Gool, Luc V. (Publ.) : *IEEE International Conference on Computer*

Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011. IEEE Computer Society, 2011. – URL https://ieeexplore.ieee.org/xpl/conhome/6118259/proceeding. – ISBN 978-1-4577-1101-5

[Mettes et al. 2015] Mettes, Pascal ; Gemert, Jan C. van ; Cappallo, Spencer ; Mensink, Thomas ; Snoek, Cees G. M. : Bag-of-Fragments: Selecting and Encoding Video Fragments for Event Detection and Recounting. In: Hauptmann, Alexander G. (Publ.) ; Ngo, Chong-Wah (Publ.) ; Xue, Xiangyang (Publ.) ; Jiang, Yu-Gang (Publ.) ; Snoek, Cees (Publ.) ; Vasconcelos, Nuno (Publ.): *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015*, ACM, 2015, pp. 427–434. – URL https://doi.org/10.1145/2671188.2749404. – ISBN 978-1-4503-3274-3

[Mi et al. 2018] Mi, Yang ; Zheng, Kang ; Wang, Song : Recognizing Actions in Wearable-Camera Videos by Training Classifiers on Fixed-Camera Videos. In: (Aizawa et al., 2018), pp. 169–177. – URL https://doi.org/10.1145/3206025.3206041

[Minsky and Papert 1987] Minsky, Marvin ; Papert, Seymour : *Perceptrons - an introduction to computational geometry*. MIT Press, 1987. – ISBN 978-0-262-63111-2

[Monfort et al. 2019] Monfort, Mathew ; Ramakrishnan, Kandan ; Andonian, Alex ; McNamara, Barry A. ; Lascelles, Alex ; Pan, Bowen ; Fan, Quanfu ; Gutfreund, Dan ; Feris, Rogério S. ; Oliva, Aude : Multi-Moments in Time: Learning and Interpreting Models for Multi-Action Video Understanding. In: *CoRR* abs/1911.00232 (2019). – URL http://arxiv.org/abs/1911.00232

[Monfort et al. 2020] Monfort, Mathew ; Vondrick, Carl ; Oliva, Aude ; Andonian, Alex ; Zhou, Bolei ; Ramakrishnan, Kandan ; Bargal, Sarah A. ; Yan, Tom ; Brown, Lisa M. ; Fan, Quanfu ; Gutfreund, Dan : Moments in Time Dataset: One Million Videos for Event Understanding. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (2020), no. 2, pp. 502–508. – URL https://doi.org/10.1109/TPAMI.2019.2901464

[Montavon et al. 2019] Montavon, Grégoire ; Binder, Alexander ; Lapuschkin, Sebastian ; Samek, Wojciech ; Müller, Klaus-Robert : Layer-Wise Relevance Propagation: An Overview. In: *Explainable AI* 11700. Springer, 2019, pp. 193–209

[Morel et al. 2017] Morel, Marion ; Achard, Catherine ; Kulpa, Richard ; Dubuisson, Séverine : Automatic evaluation of sports motion: A generic computation of spatial and temporal errors. In: *Image Vis. Comput.* 64 (2017), pp. 67–78. – URL https://doi.org/10.1016/j.imavis.2017.05.008

[Morency et al. 2010] Morency, Louis-Philippe ; Kok, Iwan de ; Gratch, Jonathan : A probabilistic multimodal approach for predicting listener backchannels. In: *Auton. Agents Multi Agent Syst.* 20 (2010), no. 1, pp. 70–84. – URL https://doi.org/10.1007/s10458-009-9092-y

[Münzer et al. 2018] Münzer, Bernd ; Leibetseder, Andreas ; Kletz, Sabrina ; Primus, Manfred J. ; Schoeffmann, Klaus : lifeXplore at the Lifelog Search Challenge 2018. In: (Gurrin et al., 2018), pp. 3–8. – URL https://doi.org/10.1145/3210539.3210541

[Nesterov 2004] Nesterov, Yurii E. : *Applied Optimization*. 87 : *Introductory Lectures on Convex Optimization - A Basic Course*. Springer, 2004. – URL https://doi.org/10.1007/978-1-4419-8853-9. – ISBN 978-1-4613-4691-3

[Neverova et al. 2016] Neverova, Natalia ; Wolf, Christian ; Taylor, Graham W. ; Nebout, Florian : ModDrop: Adaptive Multi-Modal Gesture Recognition. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2016), no. 8, pp. 1692–1706. – URL https://doi.org/10.1109/TPAMI.2015.2461544

[Newell et al. 2016] Newell, Alejandro ; Yang, Kaiyu ; Deng, Jia : Stacked Hourglass Networks for Human Pose Estimation. In: (Leibe et al., 2016), pp. 483–499. – URL https://doi.org/10.1007/978-3-319-46484-8_29. – ISBN 978-3-319-46483-1

[Ng et al. 2018] Ng, Joe Y. ; Choi, Jonghyun ; Neumann, Jan ; Davis, Larry S. : ActionFlowNet: Learning Motion Representation for Action Recognition. In: *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. (IEEE Computer Society, 2018b), pp. 1616–1624. – URL https://doi.org/10.1109/WACV.2018.00179. – ISBN 978-1-5386-4886-5

[Ng et al. 2015] Ng, Joe Y. ; Hausknecht, Matthew J. ; Vijayanarasimhan, Sudheendra ; Vinyals, Oriol ; Monga, Rajat ; Toderici, George : Beyond short snippets: Deep networks for video classification. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. (IEEE Computer Society, 2015b), pp. 4694–4702. – URL https://doi.org/10.1109/CVPR.2015.7299101. – ISBN 978-1-4673-6964-0

[Nguyen et al. 2015] Nguyen, Vinh-Tiep ; Ngo, Thanh D. ; Tran, Minh-Triet ; Le, Duy-Dinh ; Duong, Duc A. : A Combination of Spatial Pyramid and Inverted Index for Large-Scale Image Retrieval. In: *Int. J. Multim. Data Eng. Manag.* 6 (2015), no. 2, pp. 37–51. – URL https://doi.org/10.4018/IJMDEM.2015040103

[Niebles et al. 2010] Niebles, Juan C. ; Chen, Chih-Wei ; Li, Fei-Fei : Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In: Daniilidis, Kostas (Publ.) ; Maragos, Petros (Publ.) ; Paragios, Nikos (Publ.): *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II* 6312, Springer, 2010, pp. 392–405. – URL https://doi.org/10.1007/978-3-642-15552-9_29. – ISBN 978-3-642-15551-2

[Nogas et al. 2020] Nogas, Jacob ; Khan, Shehroz S. ; Mihailidis, Alex : DeepFall: Non-Invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders. In: *Journal of Healthcare Informatics Research* 4 (2020), Mar, no. 1, pp. 50–70. – URL https://doi.org/10.1007/s41666-019-00061-4. – ISSN 2509-498X

[Noiumkar and Tirakoat 2013] Noiumkar, S. ; Tirakoat, S. : Use of Optical Motion Capture in Sports Science: A Case Study of Golf Swing. In: *ICICM*, 2013, pp. 310–313

[Obeso et al. 2019] Obeso, Abraham M. ; Benois-Pineau, Jenny ; García-Vázquez, Mireya S. ; Ramírez-Acosta, Alejandro A. : Forward-backward visual saliency propagation in Deep NNs vs internal attentional mechanisms. In: *Ninth International Conference on Image Processing Theory, Tools and Applications, IPTA 2019, Istanbul, Turkey, November 6-9, 2019*. (IEEE, 2019c), pp. 1–6. – URL https://doi.org/10.1109/IPTA.2019.8936125. – ISBN 978-1-7281-3975-3

[Obeso et al. 2016] Obeso, Abraham M. ; Reyes, Laura Mariel A. ; Rodriguez, Mario L. ; Cruz, Mario Humberto M. ; Vázquez, Mireya Saraí G. ; Benois-Pineau, Jenny ; Fuentes, Luis Miguel Z. ; Martinez, Elizabeth C. ; Secundino, Jesús Abimelek F. ; Martinez, José Luis R. ; Ramírez Acosta, Alejandro Álvaro : Image annotation for Mexican buildings database. In: Iftekharuddin, Khan M. (Publ.) ; Awwal, Abdul A. S. (Publ.) ; Vázquez, Mireya G. (Publ.) ; Márquez, Andrés (Publ.) ; Matin, Mohammad A. (Publ.): *Optics and Photonics for Information Processing X* 9970 International Society for Optics and Photonics (Organizer), SPIE, 2016, pp. 201 – 208. – URL https://doi.org/10.1117/12.2238352

[Oliva and Torralba 2001] Oliva, Aude ; Torralba, Antonio : Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. In: *Int. J. Comput. Vis.* 42 (2001), no. 3, pp. 145–175. – URL https://doi.org/10.1023/A:1011139631724

[Oostdijk et al. 2020] Oostdijk, Nelleke ; Halteren, Hans van ; Basar, Erkan ; Larson, Martha A. : The Connection between the Text and Images of News Articles: New Insights for Multimedia Analysis. In: Calzolari, Nicoletta (Publ.) ; Béchet, Frédéric (Publ.) ; Blache, Philippe (Publ.) ; Choukri, Khalid (Publ.) ; Cieri, Christopher (Publ.) ; Declerck, Thierry (Publ.) ; Goggi, Sara (Publ.) ; Isahara, Hitoshi (Publ.) ; Maegaard, Bente (Publ.) ; Mariani, Joseph (Publ.) ; Mazo, Hélène (Publ.) ; Moreno, Asunción (Publ.) ; Odijk, Jan (Publ.) ; Piperidis, Stelios (Publ.): *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, European Language Resources Association, 2020, pp. 4343–4351. – URL https://www.aclweb.org/anthology/2020.lrec-1.535/. – ISBN 979-10-95546-34-4

[OpenReview.net 2017] : *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. – URL https://openreview.net/group?id=ICLR.cc/2017/conference

[OpenReview.net 2021] : *9th International Conference on Learning Representations, ICLR 2021, Virtual Only, May 4-8, 2021*. OpenReview.net, 2021. – URL https://iclr.cc/Conferences/2021

[Otte and Nagel 1994] Otte, Michael ; Nagel, Hans-Hellmut : Optical Flow Estimation: Advances and Comparisons. In: Eklundh, Jan-Olof (Publ.): *Computer Vision - ECCV'94, Third European Conference on Computer Vision, Stockholm, Sweden, May 2-6, 1994, Proceedings, Volume I* 800, Springer, 1994, pp. 51–60. – URL https://doi.org/10.1007/3-540-57956-7_5. – ISBN 3-540-57956-7

[Over et al. 2011a] Over, Paul (Publ.) ; Awad, George (Publ.) ; Fiscus, Jonathan G. (Publ.) ; Antonishek, Brian (Publ.) ; Michel, Martial (Publ.) ; Smeaton, Alan F. (Publ.) ; Kraaij, Wessel (Publ.) ; Quénot, Georges (Publ.) : *2011 TREC Video Retrieval Evaluation, TRECVID 2011, Gaithersburg, MD, USA, December 5-7, 2011*. National Institute of Standards and Technology (NIST), 2011. – URL https://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.11.org.html

[Over et al. 2011b] Over, Paul ; Awad, George ; Fiscus, Jonathan G. ; Antonishek, Brian ; Michel, Martial ; Smeaton, Alan F. ; Kraaij, Wessel ; Quénot, Georges : TRECVID 2011-Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In: *2011 TREC Video Retrieval Evaluation, TRECVID 2011, Gaithersburg, MD, USA, December 5-7, 2011*. (Over et al., 2011a). – URL https://www-nlpir.nist.gov/projects/tvpubs/tv11.papers/tv11overview.pdf

[Over et al. 2008] Over, Paul (Publ.) ; Awad, George (Publ.) ; Rose, R. T. (Publ.) ; Fiscus, Jonathan G. (Publ.) ; Kraaij, Wessel (Publ.) ; Smeaton, Alan F. (Publ.) : *TRECVID 2008 workshop participants notebook papers, Gaithersburg, MD, USA, November 2008*. National Institute of Standards and Technology (NIST), 2008. – URL https://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.8.org.html

[Ozbulak 2019] Ozbulak, Utku : *PyTorch CNN Visualizations*. https://github.com/utkuozbulak/pytorch-cnn-visualizations. 2019

[PAN et al. 2019] PAN, Xingyu ; Desbarats, Pascal ; Chaumette, Serge : A Deep Learning based Framework for UAV Trajectory Pattern Recognition. In: *Ninth International Conference on Image Processing Theory, Tools and Applications, IPTA 2019, Istanbul, Turkey, November 6-9, 2019*. (IEEE, 2019c), pp. 1–6. – URL https://doi.org/10.1109/IPTA.2019.8936099. – ISBN 978-1-7281-3975-3

[Peng et al. 2019] Peng, Wei ; Hong, Xiaopeng ; Zhao, Guoying : Video Action Recognition Via Neural Architecture Searching. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 11–15. – URL https://doi.org/10.1109/ICIP.2019.8802919. – ISBN 978-1-5386-6249-6

[Pentland et al. 2005] Pentland, Alex ; Choudhury, Tanzeem ; Eagle, Nathan ; Singh, Push : Human dynamics: computation for organizations: Human dynamics: computation for organizations. In: *Pattern Recognit. Lett.* 26 (2005), no. 4, pp. 503–511. – URL https://doi.org/10.1016/j.patrec.2004.08.012

[Petscharnig and Schöffmann 2017] Petscharnig, Stefan ; Schöffmann, Klaus : Deep Learning for Shot Classification in Gynecologic Surgery Videos. In: Amsaleg, Laurent (Publ.) ; Guðmundsson, Gylfi P. (Publ.) ; Gurrin, Cathal (Publ.) ; Jónsson, Björn Þór (Publ.) ; Satoh, Shin'ichi (Publ.): *MultiMedia Modeling - 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I* 10132, Springer, 2017, pp. 702–713. – URL https://doi.org/10.1007/978-3-319-51811-4_57. – ISBN 978-3-319-51810-7

[Qiu et al. 2018] Qiu, Haonan ; Zheng, Yingbin ; Ye, Hao ; Lu, Yao ; Wang, Feng ; He, Liang : Precise Temporal Action Localization by Evolving Temporal Proposals. In: (Aizawa et al., 2018), pp. 388–396. – URL https://doi.org/10.1145/3206025.3206029

[Rabiner and Schafer 2007] Rabiner, Lawrence R. ; Schafer, Ronald W. : Introduction to Digital Speech Processing. In: *Foundations and Trends in Signal Processing* 1 (2007), no. 1/2, pp. 1–194. – URL https://doi.org/10.1561/2000000001

[Rahmani et al. 2018] Rahmani, Hossein ; Mian, Ajmal S. ; Shah, Mubarak : Learning a Deep Model for Human Action Recognition from Novel Viewpoints. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018), no. 3, pp. 667–681. – URL https://doi.org/10.1109/TPAMI.2017.2691768

[Ramamonjisoa and Lepetit 2019] Ramamonjisoa, Michaël ; Lepetit, Vincent : SharpNet: Fast and Accurate Recovery of Occluding Contours in Monocular Depth Estimation. In: *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, IEEE, 2019, pp. 2109–2118. – URL https://doi.org/10.1109/ICCVW.2019.00266. – ISBN 978-1-7281-5023-9

[Ratner et al. 2019] Ratner, Alexander J. ; Hancock, Braden ; Ré, Christopher : The Role of Massively Multi-Task and Weak Supervision in Software 2.0. In: *CIDR 2019, 9th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 13-16, 2019, Online Proceedings*, www.cidrdb.org, 2019. – URL http://cidrdb.org/cidr2019/papers/p58-ratner-cidr19.pdf

[Reddy and Shah 2013] Reddy, Kishore K. ; Shah, Mubarak : Recognizing 50 human action categories of web videos. In: *Mach. Vis. Appl.* 24 (2013), no. 5, pp. 971–981. – URL https://doi.org/10.1007/s00138-012-0450-4

[Ren et al. 2017] Ren, Shaoqing ; He, Kaiming ; Girshick, Ross B. ; Sun, Jian : Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (2017), no. 6, pp. 1137–1149. – URL https://doi.org/10.1109/TPAMI.2016.2577031

[Rodriguez et al. 2008] Rodriguez, Mikel D. ; Ahmed, Javed ; Shah, Mubarak : Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. (IEEE Computer Society, 2008). – URL https://doi.org/10.1109/CVPR.2008.4587727. – ISBN 978-1-4244-2242-5

[Roß et al. 2020] Ross, Tobias ; Reinke, Annika ; Full, Peter M. ; Wagner, Martin ; Kenngott, Hannes ; Apitz, Martin ; Hempe, Hellena ; Filimon, Diana M. ; Scholz, Patrick ; Tran, Thuy N. ; Bruno, Pierangela ; Arbeláez, Pablo ; Bian, Gui-Bin ; Bodenstedt, Sebastian ; Bolmgren, Jon L. ; Sánchez, Laura B. ; Chen, Hua-Bin ; González, Cristina ; Guo, Dong ; Halvorsen, Pål ; Heng, Pheng-Ann ; Hosgor, Enes ; Hou, Zeng-Guang ; Isensee, Fabian ; Jha, Debesh ; Jiang, Tingting ; Jin,

Yueming ; Kirtaç, Kadir ; Kletz, Sabrina ; Leger, Stefan ; Li, Zhixuan ; Maier-Hein, Klaus H. ; Ni, Zhen-Liang ; Riegler, Michael A. ; Schoeffmann, Klaus ; Shi, Ruohua ; Speidel, Stefanie ; Stenzel, Michael ; Twick, Isabell ; Wang, Gutai ; Wang, Jiacheng ; Wang, Liansheng ; Wang, Lu ; Zhang, Yujie ; Zhou, Yan-Jie ; Zhu, Lei ; Wiesenfarth, Manuel ; Kopp-Schneider, Annette ; Müller-Stich, Beat P. ; Maier-Hein, Lena : Robust Medical Instrument Segmentation Challenge 2019. In: *CoRR* abs/2003.10299 (2020). – URL https://arxiv.org/abs/2003.10299

[Rossetto et al. 2019] Rossetto, Luca ; Parian, Mahnaz A. ; Gasser, Ralph ; Giangreco, Ivan ; Heller, Silvan ; Schuldt, Heiko : Deep Learning-Based Concept Detection in vitrivr. In: Kompatsiaris, Ioannis (Publ.) ; Huet, Benoit (Publ.) ; Mezaris, Vasileios (Publ.) ; Gurrin, Cathal (Publ.) ; Cheng, Wen-Huang (Publ.) ; Vrochidis, Stefanos (Publ.): *MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part II* 11296, Springer, 2019, pp. 616–621. – URL https://doi.org/10.1007/978-3-030-05716-9_55. – ISBN 978-3-030-05715-2

[Russakovsky et al. 2015] Russakovsky, Olga ; Deng, Jia ; Su, Hao ; Krause, Jonathan ; Satheesh, Sanjeev ; Ma, Sean ; Huang, Zhiheng ; Karpathy, Andrej ; Khosla, Aditya ; Bernstein, Michael S. ; Berg, Alexander C. ; Li, Fei-Fei : ImageNet Large Scale Visual Recognition Challenge. In: *Int. J. Comput. Vis.* 115 (2015), no. 3, pp. 211–252. – URL https://doi.org/10.1007/s11263-015-0816-y

[Safaei et al. 2018] Safaei, Marjaneh ; Balouchian, Pooyan ; Foroosh, Hassan : TICNN: A Hierarchical Deep Learning Framework for Still Image Action Recognition Using Temporal Image Prediction. In: *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*. (IEEE, 2018), pp. 3463–3467. – URL https://doi.org/10.1109/ICIP.2018.8451193. – ISBN 978-1-4799-7061-2

[Saffar et al. 2018] Saffar, Mohammad H. ; Fayyaz, Mohsen ; Sabokrou, Mohammad ; Fathy, Mahmood : Semantic Video Segmentation: A Review on Recent Approaches. In: *CoRR* abs/1806.06172 (2018). – URL http://arxiv.org/abs/1806.06172

[Sainath and Parada 2015] Sainath, Tara N. ; Parada, Carolina : Convolutional neural networks for small-footprint keyword spotting. In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, ISCA, 2015, pp. 1478–1482. – URL http://www.isca-speech.org/archive/interspeech_2015/i15_1478.html

[Schindler and Gool 2008] Schindler, Konrad ; Gool, Luc V. : Action snippets: How many frames does human action recognition require? In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. (IEEE Computer Society, 2008). – URL https://doi.org/10.1109/CVPR.2008.4587730. – ISBN 978-1-4244-2242-5

[Schoeffmann 2019] Schoeffmann, Klaus : Video Browser Showdown 2012-2019: A Review. In: (Gurrin et al., 2019), pp. 1–4. – URL https://doi.org/10.1109/CBMI.2019.8877397. – ISBN 978-1-7281-4673-7

[Schubert et al. 2017] Schubert, Erich ; Sander, Jörg ; Ester, Martin ; Kriegel, Hans-Peter ; Xu, Xiaowei : DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. In: *ACM Trans. Database Syst.* 42 (2017), no. 3, pp. 19:1–19:21. – URL https://doi.org/10.1145/3068335

[Schüldt et al. 2004] Schüldt, Christian ; Laptev, Ivan ; Caputo, Barbara : Recognizing Human Actions: A Local SVM Approach. In: *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004*, IEEE Computer Society, 2004, pp. 32–36. – URL https://doi.org/10.1109/ICPR.2004.1334462. – ISBN 0-7695-2128-2

[Schuler 2006] Schuler, Karin K. : *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*, University of Pennsylvania, Dissertation, 2006. – URL http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf

[SciPy community 2008] The SciPy community (Organizer) : *NPY format*. 2008. – URL https://numpy.org/devdocs/reference/generated/numpy.lib.format.html. – Accessed: 2020-08-01

[Selvaraju et al. 2020] Selvaraju, Ramprasaath R. ; Cogswell, Michael ; Das, Abhishek ; Vedantam, Ramakrishna ; Parikh, Devi ; Batra, Dhruv : Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: *Int. J. Comput. Vis.* 128 (2020), no. 2, pp. 336–359

[Serre et al. 2007] Serre, Thomas ; Wolf, Lior ; Bileschi, Stanley M. ; Riesenhuber, Maximilian ; Poggio, Tomaso A. : Robust Object Recognition with Cortex-Like Mechanisms. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007), no. 3, pp. 411–426. – URL https://doi.org/10.1109/TPAMI.2007.56

[Seymour and Zhang 2018] Seymour, Zachary ; Zhang, Zhongfei (. : Image Annotation Retrieval with Text-Domain Label Denoising. In: (Aizawa et al., 2018), pp. 240–248. – URL https://doi.org/10.1145/3206025.3206063

[Shao et al. 2020] Shao, Dian ; Zhao, Yue ; Dai, Bo ; Lin, Dahua : FineGym: A Hierarchical Video Dataset for Fine-Grained Action Understanding. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020* (IEEE, 2020), pp. 2613–2622. – URL https://doi.org/10.1109/CVPR42600.2020.00269. ISBN 978-1-7281-7168-5

[Shroff et al. 2010] Shroff, Nitesh ; Turaga, Pavan K. ; Chellappa, Rama : Moving vistas: Exploiting motion for describing scenes. In: *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010.* (IEEE Computer Society, 2010), pp. 1911–1918. – URL https://doi.org/10.1109/CVPR.2010.5539864. – ISBN 978-1-4244-6984-0

[Sigurdsson et al. 2018] Sigurdsson, Gunnar A. ; Gupta, Abhinav ; Schmid, Cordelia ; Farhadi, Ali ; Alahari, Karteek : Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos. In: *CoRR* abs/1804.09626 (2018). – URL http://arxiv.org/abs/1804.09626

[Sigurdsson et al. 2016] Sigurdsson, Gunnar A. ; Varol, Gül ; Wang, Xiaolong ; Farhadi, Ali ; Laptev, Ivan ; Gupta, Abhinav : Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In: Leibe, Bastian (Publ.) ; Matas, Jiri (Publ.) ; Sebe, Nicu (Publ.) ; Welling, Max (Publ.): *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I* 9905, Springer, 2016, pp. 510–526. – URL https://doi.org/10.1007/978-3-319-46448-0_31. – ISBN 978-3-319-46447-3

[Simonyan et al. 2014] Simonyan, Karen ; Vedaldi, Andrea ; Zisserman, Andrew : Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In: Bengio, Yoshua (Publ.) ; LeCun, Yann (Publ.): *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, URL http://arxiv.org/abs/1312.6034, 2014

[Simonyan and Zisserman 2014] Simonyan, Karen ; Zisserman, Andrew : Two-Stream Convolutional Networks for Action Recognition in Videos. In: (Ghahramani et al., 2014), pp. 568–576. – URL http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos

[Simonyan and Zisserman 2015] Simonyan, Karen ; Zisserman, Andrew : Very Deep Convolutional Networks for Large-Scale Image Recognition. (2015). – URL http://arxiv.org/abs/1409.1556

[Singh et al. 2016] Singh, Bharat ; Marks, Tim K. ; Jones, Michael J. ; Tuzel, Oncel ; Shao, Ming : A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 1961–1970. – URL https://doi.org/10.1109/CVPR.2016.216. – ISBN 978-1-4673-8851-1

[Singh et al. 2020] Singh, Harpreet ; Hand, Emily M. ; Alexis, Kostas : Anomalous Motion Detection on Highway Using Deep Learning. In: *CoRR* abs/2006.08143 (2020). – URL https://arxiv.org/abs/2006.08143

[Sivaprasad et al. 2018] Sivaprasad, Sarath ; Joshi, Tanmayee ; Agrawal, Rishabh ; Pedanekar, Niranjan : Multimodal Continuous Prediction of Emotions in Movies using Long Short-Term Memory Networks. In: (Aizawa et al., 2018), pp. 413–419. – URL https://doi.org/10.1145/3206025.3206076

[Smaira et al. 2020] Smaira, Lucas ; Carreira, João ; Noland, Eric ; Clancy, Ellen ; Wu, Amy ; Zisserman, Andrew : A Short Note on the Kinetics-700-2020 Human Action Dataset. In: *CoRR* abs/2010.10864 (2020). – URL https://arxiv.org/abs/2010.10864

[Smith et al. 2019] Smith, Abraham G. ; Petersen, Jens ; Selvan, Raghavendra ; Rasmussen, Camilla R. : Segmentation of Roots in Soil with U-Net. In: *CoRR* abs/1902.11050 (2019). – URL http://arxiv.org/abs/1902.11050

[Sokolova et al. 2020] Sokolova, Natalia ; Taschwer, Mario ; Sarny, Stephanie ; Putzgruber-Adamitsch, Doris ; Schoeffmann, Klaus : Pixel-Based Iris and Pupil Segmentation in Cataract Surgery Videos Using Mask R-CNN. In: *2020 IEEE 17th International Symposium on Biomedical Imaging Workshops (ISBI Workshops), Iowa City, IA, USA, April 4, 2020*, IEEE, 2020, pp. 1–4. – URL https://doi.org/10.1109/ISBIWorkshops50223.2020.9153367. – ISBN 978-1-7281-7401-3

[Soomro and Zamir 2014] Soomro, Khurram ; Zamir, Amir R. : *Action Recognition in Realistic Sports Videos*. pp. 181–208. In: Moeslund, Thomas B. (Publ.) ; Thomas, Graham (Publ.) ; Hilton, Adrian (Publ.): *Computer Vision in Sports*. Cham : Springer International Publishing, 2014. – URL https://doi.org/10.1007/978-3-319-09396-3_9. – ISBN 978-3-319-09396-3

[Soomro et al. 2012] Soomro, Khurram ; Zamir, Amir R. ; Shah, Mubarak : UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. In: *CoRR* abs/1212.0402 (2012). – URL http://arxiv.org/abs/1212.0402

[Springenberg et al. 2015] Springenberg, Jost T. ; Dosovitskiy, Alexey ; Brox, Thomas ; Riedmiller, Martin A. : Striving for Simplicity: The All Convolutional Net. In: Bengio, Yoshua (Publ.) ; LeCun, Yann (Publ.): *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, URL http://arxiv.org/abs/1412.6806, 2015

[Sriraman et al. 2019] Sriraman, Siddharth ; Srinivasan, Srinath ; Krishnan, Vishnu K. ; J, Bhuvana ; Mirnalinee, T. T. : MediaEval 2019: LRCNs for Stroke Detection in Table Tennis. In: (Larson et al., 2020). – URL http://ceur-ws.org/Vol-2670/MediaEval_19_paper_36.pdf

[Stöckli et al. 2018] Stöckli, Sabrina ; Schulte-Mecklenbeck, Michael ; Borer, Stefan ; Samson, Andrea C. : Facial expression analysis with AFFDEX and FACET: A validation study. In: *Behavior Research Methods* 50 (2018), Aug, no. 4, pp. 1446–1460. – URL https://doi.org/10.3758/s13428-017-0996-1. – ISSN 1554-3528

[Stoian et al. 2016] Stoian, Andrei ; Ferecatu, Marin ; Benois-Pineau, Jenny ; Crucianu, Michel : Fast Action Localization in Large-Scale Video Archives. In: *IEEE Trans. Circuits Syst. Video Techn.* 26 (2016), no. 10, pp. 1917–1930. – URL https://doi.org/10.1109/TCSVT.2015.2475835

[Sudhakaran et al. 2020] Sudhakaran, Swathikiran ; Escalera, Sergio ; Lanz, Oswald : Gate-Shift Networks for Video Action Recognition. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. (IEEE, 2020), pp. 1099–1108. – URL https://doi.org/10.1109/CVPR42600.2020.00118. – ISBN 978-1-7281-7168-5

[Sun et al. 2010] Sun, Ju ; Mu, Yadong ; Yan, Shuicheng ; Cheong, Loong F. : Activity recognition using dense long-duration trajectories. In: *Proceedings of the 2010 IEEE International Conference on Multimedia and Expo, ICME 2010, 19-23 July 2010, Singapore*, IEEE Computer Society, 2010, pp. 322–327. – URL https://doi.org/10.1109/ICME.2010.5583046. – ISBN 978-1-4244-7491-2

[Sun et al. 2015] Sun, Lin ; Jia, Kui ; Yeung, Dit-Yan ; Shi, Bertram E. : Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. (IEEE Computer Society, 2015a), pp. 4597–4605. – URL https://doi.org/10.1109/ICCV.2015.522. – ISBN 978-1-4673-8391-2

[Sundberg et al. 2011] Sundberg, Patrik ; Brox, Thomas ; Maire, Michael ; Arbelaez, Pablo ; Malik, Jitendra : Occlusion boundary detection and figure/ground assignment from optical flow. In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. (IEEE Computer Society, 2011), pp. 2233–2240. – URL https://doi.org/10.1109/CVPR.2011.5995364. – ISBN 978-1-4577-0394-2

[Sutskever et al. 2013] Sutskever, Ilya ; Martens, James ; Dahl, George E. ; Hinton, Geoffrey E. : On the importance of initialization and momentum in deep learning. In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013* 28, JMLR.org, 2013, pp. 1139–1147. – URL http://proceedings.mlr.press/v28/sutskever13.html

[Szegedy et al. 2015] Szegedy, Christian ; Liu, Wei ; Jia, Yangqing ; Sermanet, Pierre ; Reed, Scott E. ; Anguelov, Dragomir ; Erhan, Dumitru ; Vanhoucke, Vincent ; Rabinovich, Andrew : Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. (IEEE Computer Society, 2015b), pp. 1–9. – URL https://doi.org/10.1109/CVPR.2015.7298594. – ISBN 978-1-4673-6964-0

[Szegedy et al. 2016] Szegedy, Christian ; Vanhoucke, Vincent ; Ioffe, Sergey ; Shlens, Jonathon ; Wojna, Zbigniew : Rethinking the Inception Architecture for Computer Vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 2818–2826. – URL https://doi.org/10.1109/CVPR.2016.308. – ISBN 978-1-4673-8851-1

[Tabrizi et al. 2020] Tabrizi, S. S. ; Pashazadeh, S. ; Javani, V. : Comparative Study of Table Tennis Forehand Strokes Classification Using Deep Learning and SVM. In: *IEEE Sensors Journal* (2020), pp. 1–1

[Tang et al. 2013] Tang, Kevin D. ; Yao, Bangpeng ; Li, Fei-Fei ; Koller, Daphne : Combining the Right Features for Complex Event Recognition. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. (IEEE Computer Society, 2013), pp. 2696–2703. – URL https://doi.org/10.1109/ICCV.2013.335. – ISBN 978-1-4799-2839-2

[Tang et al. 2018] Tang, Yi ; Zou, Wenbin ; Jin, Zhi ; Li, Xia : Multi-Scale Spatiotemporal Conv-LSTM Network for Video Saliency Detection. In: (Aizawa et al., 2018), pp. 362–369. – URL https://doi.org/10.1145/3206025.3206052

[Theriault et al. 2013a] Theriault, Christian ; Thome, Nicolas ; Cord, Matthieu : Dynamic Scene Classification: Learning Motion Descriptors with Slow Features Analysis. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, IEEE Computer Society, 2013, pp. 2603–2610. – URL https://doi.org/10.1109/CVPR.2013.336. – ISBN 978-0-7695-4989-7

[Theriault et al. 2013b] Theriault, Christian ; Thome, Nicolas ; Cord, Matthieu : Extended Coding and Pooling in the HMAX Model. In: *IEEE Trans. Image Process.* 22 (2013), no. 2, pp. 764–777. – URL https://doi.org/10.1109/TIP.2012.2222900

[Theriault et al. 2014] Theriault, Christian ; Thome, Nicolas ; Cord, Matthieu ; Pérez, Patrick : Perceptual Principles for Video Classification With Slow Feature Analysis. In: *IEEE J. Sel. Top. Signal Process.* 8 (2014), no. 3, pp. 428–437. – URL https://doi.org/10.1109/JSTSP.2014.2315742

[Tomsett et al. 2020] Tomsett, Richard ; Harborne, Dan ; Chakraborty, Supriyo ; Gurram, Prudhvi ; Preece, Alun D. : Sanity Checks for Saliency Metrics. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 6021–6029. – URL https://aaai.org/ojs/index.php/AAAI/article/view/6064. – ISBN 978-1-57735-823-7

[Tran et al. 2015] Tran, Du ; Bourdev, Lubomir D. ; Fergus, Rob ; Torresani, Lorenzo ; Paluri, Manohar : Learning Spatiotemporal Features with 3D Convolutional Networks. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. (IEEE Computer Society, 2015a), pp. 4489–4497. – URL https://doi.org/10.1109/ICCV.2015.510. – ISBN 978-1-4673-8391-2

[Tran et al. 2018] Tran, Du ; Wang, Heng ; Torresani, Lorenzo ; Ray, Jamie ; LeCun, Yann ; Paluri, Manohar : A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. (IEEE Computer Society, 2018a), pp. 6450–6459. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Tran_A_Closer_Look_CVPR_2018_paper.html

[Truong et al. 2018] Truong, Thanh-Dat ; Duy, Tung D. ; Nguyen, Vinh-Tiep ; Tran, Minh-Triet : Lifelogging Retrieval based on Semantic Concepts Fusion. In: (Gurrin et al., 2018), pp. 24–29. – URL https://doi.org/10.1145/3210539.3210545

[Tsai 2018] Tsai, Wan-Lun : Personal Basketball Coach: Tactic Training through Wireless Virtual Reality. In: (Aizawa et al., 2018), pp. 481–484. – URL https://doi.org/10.1145/3206025.3206084

[Ullah et al. 2018] Ullah, Amin ; Ahmad, Jamil ; Muhammad, Khan ; Sajjad, Muhammad ; Baik, Sung W. : Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. In: *IEEE Access* 6 (2018), pp. 1155–1166. – URL https://doi.org/10.1109/ACCESS.2017.2778011

[Vadivelu et al. 2016] Vadivelu, Somasundaram ; Ganesan, Sudakshin ; Murthy, O. V. R. ; Dhall, Abhinav : Thermal Imaging Based Elderly Fall Detection. In: Chen, Chu-Song (Publ.) ; Lu, Jiwen (Publ.) ; Ma, Kai-Kuang (Publ.): *Computer Vision - ACCV 2016 Workshops - ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III* 10118, Springer, 2016, pp. 541–553. – URL https://doi.org/10.1007/978-3-319-54526-4_40

[Varol et al. 2018] Varol, Gül ; Laptev, Ivan ; Schmid, Cordelia : Long-Term Temporal Convolutions for Action Recognition. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (2018), no. 6, pp. 1510–1517. – URL https://doi.org/10.1109/TPAMI.2017.2712608

[Vaswani et al. 2017] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, Lukasz ; Polosukhin, Illia : Attention is All you Need. In: Guyon, Isabelle (Publ.) ; Luxburg, Ulrike von (Publ.) ; Bengio, Samy (Publ.) ; Wallach, Hanna M. (Publ.) ; Fergus, Rob (Publ.) ; Vishwanathan, S. V. N. (Publ.) ; Garnett, Roman (Publ.): *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, URL http://papers.nips.cc/paper/7181-attention-is-all-you-need, 2017, pp. 5998–6008

[Vo et al. 2019] Vo, Huy V. ; Bach, Francis ; Cho, Minsu ; Han, Kai ; LeCun, Yann ; Pérez, Patrick ; Ponce, Jean : Unsupervised Image Matching and Object Discovery as Optimization. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. (Computer Vision Foundation / IEEE, 2019), pp. 8287–8296. – URL http://openaccess.thecvf.com/content_CVPR_2019/html/Vo_Unsupervised_Image_Matching_and_Object_Discovery_as_Optimization_CVPR_2019_paper.html

[Voeikov et al. 2020] Voeikov, Roman ; Falaleev, Nikolay ; Baikulov, Ruslan : TTNet: Real-time temporal and spatial video analysis of table tennis. In: *CoRR* abs/2004.09927 (2020). – URL https://arxiv.org/abs/2004.09927

[Vondrick et al. 2010] Vondrick, Carl ; Ramanan, Deva ; Patterson, Donald J. : Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces. In: Daniilidis, Kostas (Publ.) ; Maragos, Petros (Publ.) ; Paragios, Nikos (Publ.): *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV* 6314, Springer, 2010, pp. 610–623. – URL https://doi.org/10.1007/978-3-642-15561-1_44. – ISBN 978-3-642-15560-4

[Wallach et al. 2019] Wallach, Hanna M. (Publ.) ; Larochelle, Hugo (Publ.) ; Beygelzimer, Alina (Publ.) ; d'Alché-Buc, Florence (Publ.) ; Fox, Emily B. (Publ.) ; Garnett, Roman (Publ.) : *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. URL http://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019, 2019

[Wang et al. 2017a] Wang, Fei ; Jiang, Mengqing ; Qian, Chen ; Yang, Shuo ; Li, Cheng ; Zhang, Honggang ; Wang, Xiaogang ; Tang, Xiaoou : Residual Attention Network for Image Classification. In: *CVPR*, IEEE Computer Society, 2017, pp. 6450–6458

[Wang et al. 2011] Wang, Heng ; Kläser, Alexander ; Schmid, Cordelia ; Liu, Cheng-Lin : Action recognition by dense trajectories. In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. (IEEE Computer Society, 2011), pp. 3169–3176. – URL https://doi.org/10.1109/CVPR.2011.5995407. – ISBN 978-1-4577-0394-2

[Wang et al. 2013] Wang, Heng ; Kläser, Alexander ; Schmid, Cordelia ; Liu, Cheng-Lin : Dense Trajectories and Motion Boundary Descriptors for Action Recognition. In: *Int. J. Comput. Vis.* 103 (2013), no. 1, pp. 60–79. – URL https://doi.org/10.1007/s11263-012-0594-8

[Wang and Schmid 2013] Wang, Heng ; Schmid, Cordelia : Action Recognition with Improved Trajectories. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. (IEEE Computer Society, 2013), pp. 3551–3558. – URL https://doi.org/10.1109/ICCV.2013.441. – ISBN 978-1-4799-2839-2

[Wang et al. 2020] Wang, Jiachen ; Zhao, Kejian ; Deng, Dazhen ; Cao, Anqi ; Xie, Xiao ; Zhou, Zheng ; Zhang, Hui ; Wu, Yingcai : Tac-Simur: Tactic-based Simulative Visual Analytics of Table Tennis. In: *IEEE Trans. Vis. Comput. Graph.* 26 (2020), no. 1, pp. 407–417. – URL https://doi.org/10.1109/TVCG.2019.2934630

[Wang et al. 2019a] Wang, Jiahao ; Du, Zhengyin ; Li, Annan ; Wang, Yunhong : Atrous Temporal Convolutional Network for Video Action Segmentation. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 1585–1589. – URL https://doi.org/10.1109/ICIP.2019.8803088. – ISBN 978-1-5386-6249-6

[Wang et al. 2017b] Wang, Limin ; Xiong, Yuanjun ; Lin, Dahua ; Gool, Luc V. : UntrimmedNets for Weakly Supervised Action Recognition and Detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 6402–6411. – URL https://doi.org/10.1109/CVPR.2017.678. – ISBN 978-1-5386-0457-1

[Wang et al. 2016] Wang, Limin ; Xiong, Yuanjun ; Wang, Zhe ; Qiao, Yu ; Lin, Dahua ; Tang, Xiaoou ; Gool, Luc V. : Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In: (Leibe et al., 2016), pp. 20–36. – URL https://doi.org/10.1007/978-3-319-46484-8_2. – ISBN 978-3-319-46483-1

[Wang et al. 2019b] Wang, Limin ; Xiong, Yuanjun ; Wang, Zhe ; Qiao, Yu ; Lin, Dahua ; Tang, Xiaoou ; Gool, Luc V. : Temporal Segment Networks for Action Recognition in Videos. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (2019), no. 11, pp. 2740–2755. – URL https://doi.org/10.1109/TPAMI.2018.2868668

[Wang et al. 2019c] Wang, Tao ; Rostamza, Mina ; Song, Zhihang ; Wang, Liangju ; McNickle, G. ; Iyer-Pascuzzi, Anjali S. ; Qiu, Zhengjun ; Jin, Jian : SegRoot: A high throughput segmentation method for root image analysis. In: *Comput. Electron. Agric.* 162 (2019), pp. 845–854. – URL https://doi.org/10.1016/j.compag.2019.05.017

[Wang et al. 2018a] Wang, Xiaolong ; Girshick, Ross B. ; Gupta, Abhinav ; He, Kaiming : Non-Local Neural Networks. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. (IEEE Computer Society, 2018a), pp. 7794–7803. – URL http://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Non-Local_Neural_Networks_CVPR_2018_paper.html

[Wang et al. 2015] Wang, Xin ; Kumar, Devinder ; Thome, Nicolas ; Cord, Matthieu ; Precioso, Frédéric : Recipe recognition with large multimodal food dataset. In: *2015 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops 2015, Turin, Italy, June 29 - July 3, 2015*, IEEE Computer Society, 2015, pp. 1–6. – URL https://doi.org/10.1109/ICMEW.2015.7169757. – ISBN 978-1-4799-7079-7

[Wang et al. 2017c] Wang, Xin ; Thome, Nicolas ; Cord, Matthieu : Gaze latent support vector machine for image classification improved by weakly supervised region selection. In: *Pattern Recognit.* 72 (2017), pp. 59–71. – URL https://doi.org/10.1016/j.patcog.2017.07.001

[Wang et al. 2018b] Wang, Xuanhan ; Gao, Lianli ; Wang, Peng ; Sun, Xiaoshuai ; Liu, Xianglong : Two-Stream 3-D convNet Fusion for Action Recognition in Videos With Arbitrary Size and Length. In: *IEEE Trans. Multimedia* 20 (2018), no. 3, pp. 634–644. – URL https://doi.org/10.1109/TMM.2017.2749159

[Weinzaepfel et al. 2015] Weinzaepfel, Philippe ; Harchaoui, Zaïd ; Schmid, Cordelia : Learning to Track for Spatio-Temporal Action Localization. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. (IEEE Computer Society, 2015a), pp. 3164–3172. – URL https://doi.org/10.1109/ICCV.2015.362. – ISBN 978-1-4673-8391-2

[Weinzaepfel et al. 2013] Weinzaepfel, Philippe ; Revaud, Jérôme ; Harchaoui, Zaïd ; Schmid, Cordelia : DeepFlow: Large Displacement Optical Flow with Deep Matching. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. (IEEE Computer Society, 2013), pp. 1385–1392. – URL https://doi.org/10.1109/ICCV.2013.175. – ISBN 978-1-4799-2839-2

[Wiskott and Sejnowski 2002] Wiskott, Laurenz ; Sejnowski, Terrence J. : Slow Feature Analysis: Unsupervised Learning of Invariances. In: *Neural Computation* 14 (2002), no. 4, pp. 715–770. – URL https://doi.org/10.1162/089976602317318938

[Wright et al. 2009] Wright, John ; Yang, Allen Y. ; Ganesh, Arvind ; Sastry, Shankar S. ; Ma, Yi : Robust Face Recognition via Sparse Representation. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2009), no. 2, pp. 210–227. – URL https://doi.org/10.1109/TPAMI.2008.79

[Wu et al. 2019] Wu, Chengjie ; Han, Jiayue ; Li, Xiaoqiang : Time-Asymmetric 3d Convolutional Neural Networks for Action Recognition. In: *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. (IEEE, 2019a), pp. 21–25. – URL https://doi.org/10.1109/ICIP.2019.8802910. – ISBN 978-1-5386-6249-6

[Wu and Koike 2020] Wu, Erwin ; Koike, Hideki : FuturePong: Real-time Table Tennis Trajectory Forecasting using Pose Prediction Network. In: Bernhaupt, Regina (Publ.) ; Mueller, Florian '. (Publ.) ; Verweij, David (Publ.) ; Andres, Josh (Publ.) ; McGrenere, Joanna (Publ.) ; Cockburn, Andy (Publ.) ; Avellino, Ignacio (Publ.) ; Goguey, Alix (Publ.) ; Bjøn, Pernille (Publ.) ; Zhao, Shengdong (Publ.) ; Samson, Briane P. (Publ.) ; Kocielnik, Rafal (Publ.): *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, Honolulu, HI, USA, April 25-30, 2020*, ACM, 2020, pp. 1–8. – URL https://doi.org/10.1145/3334480.3382853. – ISBN 978-1-4503-6819-3

[Xia et al. 2020] Xia, Kun ; Wang, Hanyu ; Xu, Menghan ; Li, Zheng ; He, Sheng ; Tang, Yusong : Racquet Sports Recognition Using a Hybrid Clustering Model Learned from Integrated Wearable Sensor. In: *Sensors* 20 (2020), no. 6, pp. 1638. – URL https://doi.org/10.3390/s20061638

[Xie et al. 2017] Xie, Saining ; Girshick, Ross B. ; Dollár, Piotr ; Tu, Zhuowen ; He, Kaiming : Aggregated Residual Transformations for Deep Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. (IEEE Computer Society, 2017a), pp. 5987–5995. – URL https://doi.org/10.1109/CVPR.2017.634. – ISBN 978-1-5386-0457-1

[Xu et al. 2018] Xu, Baohan ; Ye, Hao ; Zheng, Yingbin ; Wang, Heng ; Luwang, Tianyu ; Jiang, Yu-Gang : Dense Dilated Network for Few Shot Action Recognition. In: (Aizawa et al., 2018), pp. 379–387. – URL https://doi.org/10.1145/3206025.3206028

[Yasrab et al. 2019] Yasrab, Robail ; Atkinson, Jonathan A. ; Wells, Darren M. ; French, Andrew P. ; Pridmore, Tony P. ; Pound, Michael P. : RootNav 2.0: Deep learning for automatic navigation of complex plant root architectures. In: *GigaScience* 8 (2019), Nov, no. 11, pp. giz123. – URL https://pubmed.ncbi.nlm.nih.gov/31702012. – ISSN 2047-217X

[Yasrab et al. 2020] Yasrab, Robail ; Pound, Michael P. ; French, Andrew P. ; Pridmore, Tony P. : PhenomNet: Bridging Phenotype-Genotype Gap: A CNN-LSTM Based Automatic Plant Root Anatomization System. In: *bioRxiv* (2020). – URL https://www.biorxiv.org/content/early/2020/05/07/2020.05.03.075184

[Yu et al. 2020] Yu, Fisher ; Chen, Haofeng ; Wang, Xin ; Xian, Wenqi ; Chen, Yingying ; Liu, Fangchen ; Madhavan, Vashisht ; Darrell, Trevor : BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. (IEEE, 2020), pp. 2633–2642. – URL https://doi.org/10.1109/CVPR42600.2020.00271. – ISBN 978-1-7281-7168-5

[Yu 2010] Yu, Shun-Zheng : Hidden semi-Markov models. In: *Artif. Intell.* 174 (2010), no. 2, pp. 215–243. – URL https://doi.org/10.1016/j.artint.2009.11.011

[Yuan et al. 2016] Yuan, Jun ; Ni, Bingbing ; Yang, Xiaokang ; Kassim, Ashraf A. : Temporal Action Localization with Pyramid of Score Distribution Features. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 3093–3102. – URL https://doi.org/10.1109/CVPR.2016.337. – ISBN 978-1-4673-8851-1

[Zach et al. 2007] Zach, Christopher ; Pock, Thomas ; Bischof, Horst : A Duality Based Approach for Realtime TV-L^1 Optical Flow. In: Hamprecht, Fred A. (Publ.) ; Schnörr, Christoph (Publ.) ; Jähne, Bernd (Publ.): *Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings* 4713, Springer, 2007, pp. 214–223. – URL https://doi.org/10.1007/978-3-540-74936-3_22. – ISBN 978-3-540-74933-2

[Zagoruyko and Komodakis 2016] Zagoruyko, Sergey ; Komodakis, Nikos : Wide Residual Networks. In: Wilson, Richard C. (Publ.) ; Hancock, Edwin R. (Publ.) ; Smith, William A. P. (Publ.): *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, BMVA Press, 2016. – URL http://www.bmva.org/bmvc/2016/papers/paper087/index.html

[Zagoruyko and Komodakis 2017] Zagoruyko, Sergey ; Komodakis, Nikos : Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. (OpenReview.net, 2017). – URL https://openreview.net/forum?id=Sks9_ajex

[Zemmari and Benois-Pineau 2020] Zemmari, Akka ; Benois-Pineau, Jenny : *Deep Learning in Mining of Visual Content*. Springer, 2020 (Springer Briefs in Computer Science). – URL https://doi.org/10.1007/978-3-030-34376-7. – ISBN 978-3-030-34375-0

[Zhang et al. 2016] Zhang, Bowen ; Wang, Limin ; Wang, Zhe ; Qiao, Yu ; Wang, Hanli : Real-Time Action Recognition with Enhanced Motion Vector CNNs. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. (IEEE Computer Society, 2016), pp. 2718–2726. – URL https://doi.org/10.1109/CVPR.2016.297. – ISBN 978-1-4673-8851-1

[Zhang and Tao 2012] Zhang, Zhang ; Tao, Dacheng : Slow Feature Analysis for Human Action Recognition. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2012), no. 3, pp. 436–450. – URL https://doi.org/10.1109/TPAMI.2011.157

[Zhao et al. 2019] Zhao, Hang ; Torralba, Antonio ; Torresani, Lorenzo ; Yan, Zhicheng : HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. (IEEE, 2019b), pp. 8667–8677. – URL https://doi.org/10.1109/ICCV.2019.00876. – ISBN 978-1-7281-4803-8

[Zhao and Snoek 2019] Zhao, Jiaojiao ; Snoek, Cees G. M. : Dance With Flow: Two-In-One Stream Action Detection. In: *CVPR*, Computer Vision Foundation / IEEE, 2019, pp. 9935–9944

[Zhao et al. 2020] Zhao, Yue ; Xiong, Yuanjun ; Wang, Limin ; Wu, Zhirong ; Tang, Xiaoou ; Lin, Dahua : Temporal Action Detection with Structured Segment Networks. In: *Int. J. Comput. Vis.* 128 (2020), no. 1, pp. 74–95. – URL https://doi.org/10.1007/s11263-019-01211-2

[Zheng et al. 2020] Zheng, Tianhang ; Liu, Sheng ; Chen, Changyou ; Yuan, Junsong ; Li, Baochun ; Ren, Kui : Towards Understanding the Adversarial Vulnerability of Skeleton-based Action Recognition. In: *CoRR* abs/2005.07151 (2020). – URL https://arxiv.org/abs/2005.07151

[Zhou et al. 2014] Zhou, Bolei ; Lapedriza, Àgata ; Xiao, Jianxiong ; Torralba, Antonio ; Oliva, Aude : Learning Deep Features for Scene Recognition using Places Database. In: (Ghahramani et al., 2014), pp. 487–495. – URL http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database

[Zintgraf et al. 2017] Zintgraf, Luisa M. ; Cohen, Taco S. ; Adel, Tameem ; Welling, Max : Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. (OpenReview.net, 2017). – URL https://openreview.net/forum?id=BJ5UeU9xx

[Zivkovic and van der Heijden 2006] Zivkovic, Zoran ; Heijden, Ferdinand van der : Efficient adaptive density estimation per image pixel for the task of background subtraction. In: *Pattern Recognit. Lett.* 27 (2006), no. 7, pp. 773–780. – URL https://doi.org/10.1016/j.patrec.2005.11.005

List of Acronyms

aaAE average aAE 106, 107

aAE average AE 105, 106, 121, 255

aaEPE average aEPE 106, 107

ADL Activities of Daily Living 43, 59

AE angular error 105, 255

aEPE average EPE 105, 106, 121, 255

AI Artificial Intelligence 3–5, 138

aMSE average MSE 106–108, 121

AMT Amazon Mechanical Turk 51, 52, 61, 67, 69, 260

AVA Atomic Visual Actions 68, 71, 260

BERT Bidirectional Encoder Representations from Transformers 43, 46, 50, 162

BoVW Bag-of-Visual Words 16

BoW Bag of Words 16, 21, 26

BP Beyond Pixel 105–108, 110, 112, 113, 115–117, 120, 125, 164

C3D Convolutional 3D 38, 39, 46

CNN Convolutional Neural Network 3, 4, 15, 31–34, 36–39, 42, 43, 45, 91, 114, 121, 137, 139–141, 162, 260

CNNs Convolutional Neural Networks 31, 35, 43, 45, 137, 138, 140, 141, 150, 159, 162, 178

CPU Central Processing Unit 106

CPUs Central Processing Units 78

CRISP ComputeR vIsion for Sport Performance 4–6, 70, 78

CRNN Convolutional Recurrent Neural Network 33

DCNN Deep Convolutional Neural Network 15

DIS Dense Inverse Search 105–108, 110

DNN Deep Neural Network 15, 16, 23, 28, 34, 35, 78, 139

DNNs Deep Neural Networks 15, 35, 115, 139

EF-STCNN Early Fusion Spatio-Temporal Convolutional Neural Network 127, 128, 131, 263

EPE end-point error 105, 255

FC Fully Connected 31, 36, 45, 141

Flow-STCNN Flow Spatio Temporal Convolutional Neural Network 103, 112–115, 118–121, 127, 129, 133, 134, 164, 168–173, 177, 261

fps frames per second 54, 62, 64, 77, 78, 94, 115

FSTCN Factorized Spatio-Temporal Convolutional Networks 36, 46

GPU Graphics Processing Unit 94, 176

GPUs Graphics Processing Units 34, 47, 78

GSM Gate-Shift Module 38, 260

HMDB Human Motion DataBase 64, 260

HMM Hidden Markov Model 27

HMMs Hidden Markov Models 25

HOF Histogram of Oriented Optical Flow 17, 19–21, 259

HOG Histogram of Oriented Gradients 17, 19, 21, 26

HSL Hue, Saturation, Lightness 22, 108, 145, 261

IDT Improved Dense Trajectories 21, 22, 39, 46

IPCV Image Processing and Computer Vision 138

LF-STCNN Late Fusion Spatio-Temporal Convolutional Neural Network 127, 128, 171, 176, 177, 263

LRCN Long-term Recurrent Convolutional Network 37, 46

LRP Fully Layer-Wise Relevance Propagation 140

LSTM Long Short-Term Memory 34, 36, 37, 42, 43, 159, 161, 260

LTC Long-term Temporal Convolutions 42, 87, 260

MARS Motion-Augmented RGB Stream 42, 46

MBH Motion Boundary Histogram 17–19, 21, 259

MFCC Mel-Frequency Cepstral Coefficient 19

MLP Multi-Layer Perceptron 32, 33, 260

MSE Mean Squared Error 105, 106, 108, 255

MSN Multi-Stream Network 36

NN Neural Network 32, 45

NNC Nearest Neighbor Classification 17

NTSC National Television System Committee 54

OF Optical Flow 15, 18–20, 25, 26, 35–37, 39, 42, 43, 45, 46, 103–106, 108, 110–112, 114–117, 119–121, 123, 124, 126–129, 134, 138, 145, 148–150, 161, 162, 164, 170, 173, 176, 183, 259, 261

PCA Principal Component Analysis 23, 27

PCC Pearson Correlation Coefficient 148, 149

R-CNN Residual Convolutional Neural Network 36, 39

RANSAC Random Sample Consensus 21

ResNet Residual Network 32, 37, 42, 43

RGB Red, Green, Blue 35–37, 39, 42, 43, 45, 46, 57, 87–89, 91, 94–97, 100, 103, 111, 120, 121, 123–127, 131, 134, 138, 141, 144–150, 161–164, 167, 168, 171, 173, 178, 183, 196, 261–263

RGB-STCNN RGB Spatio Temporal Convolutional Neural Network 88, 89, 94, 96–98, 100, 112, 114, 115, 120, 127, 129, 131, 133, 134, 164, 168–174, 176, 177, 183, 261, 262

RNN Recurrent Neural Network 34, 38, 260

ROI Region-of-Interest 36, 89, 90, 110, 115, 125, 126

ROIs Regions-of-Interest 110

SCI Sparse Concentration Index 36

SFA Slow Feature Analysis 23

SGD Stochastic Gradient Descent 32, 164

SIFT Scale-Invariant Feature Transform 16, 19

ST-ResNet Spatio-Temporal Residual Network 39, 46

STIP Spatio-Temporal Interest Points 16, 17, 259

STPP Spatial Temporal Pyramid Pooling Layer 43

STRN Spatio-Temporal Relation Networks 43, 46

STS Space-Time Shapes 17, 18, 26, 259

SURF Speeded Up Robust Features 21

SVM Support Vector Machine 15, 17, 19, 26, 27, 36, 39, 46

T-CNN Tube Convolutional Neural Network 36, 39, 40, 46, 260

T-STCNN Twin Spatio-Temporal Convolutional Neural Network 124–131, 133, 134, 138, 143–147, 150, 151, 159, 160, 162–164, 167–173, 175–178, 183, 184, 196, 261–263

TSM Temporal Shift Module 37, 46

TSN Temporal Segment Network 37, 46

TUHAD Human Motion Taekwondo Unit Technique Human Action Dataset 58, 260

List of Figures

List of Tables

Table of Contents

Title: Fine-Grained Action Detection and Classification from Videos with Spatio-Temporal Convolutional Neural Networks. Application to Table Tennis.

Abstract: This thesis deals with fine-grained classification of sport gestures from videos, with an application to table tennis. Our aim is to design a smart system for students and teachers for analyzing their performance. By profiling his players, a teacher can therefore tailor his training sessions more efficiently for improving their skills. Players can also have an instant feedback on their performance.
For developing such a system with fine-grained classification, a very specific dataset is needed to supervise the learning process. To that aim, we built the "`TTStroke-21`" dataset, that allows us to train and test our model. We introduce a convolutional spatio-temporal neural network with a Twin architecture, as inputs an RGB image sequence and its computed Optical Flow. These two types of data are processed in parallel and allow an efficient classification of our video clips. The proposed method is also applied to joint detection and classification task, and reaches a good accuracy.

Keywords: Deep Learning, Action classification, Spatio-temporal convolution, Table tennis, Optical Flow, Computer Vision, Video indexing

---

Titre : Détection et classification fines d'actions à partir de vidéos par réseaux de neurones à convolutions spatio-temporelles. Application au tennis de table.

Résumé : Cette thèse porte sur la classification de gestes sportifs à partir de vidéos, avec comme cadre applicatif le tennis de table. Notre objectif est de concevoir un système intelligent permettant d'analyser les performances des élèves pongistes, et de donner la possibilité à l'entraîneur d'adapter ses séances d'entraînement pour améliorer leurs performances.
Nous introduisons dans ce travail de thèse un réseau de neurones spatio-temporel convolutif avec une architecture Jumelle, prenant en entrée une séquence d'images RGB et son flot optique estimé. Afin de superviser le processus d'apprentissage, il est nécessaire d'avoir une base de données spécifique. Dans ce but, nous avons élaboré la base de données "`TTStroke-21`" afin d'entraîner et de tester notre modèle. La méthode proposée permet une classification efficace des vidéos issues de `TTStroke-21`, et atteint aussi une bonne précision lorsqu'elle est appliquée pour conjointement détecter et classer une action.

Mots clés : Apprentissage profond, Classification d'actions, Tennis de table, Convolutions Spatio-temporelles, Indexation vidéo, Flot optique, Vision par ordinateur

---

Unités de recherche

Université de Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
Université de La Rochelle, MIA, La Rochelle, France