Egocentric Hand Activity Recognition : The principal components of an egocentric hand activity recognition framework, exploitable for augmented reality user assistance

Mohamed Yasser Boutaleb

DOCTORAT BRETAGNE LOIRE MATHSTIC

CentraleSupélec

THESE DE DOCTORAT DE

CENTRALESUPELEC

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *AST – Signal, Image, Vision*

Par

Mohamed Yasser BOUTALEB

Egocentric Hand Activity Recognition

The principal components of an egocentric hand activity recognition framework, exploitable for augmented reality user assistance

Thèse présentée et soutenue à l'IRT B<>COM le 06/12/2022
Unité de recherche : Équipe AIMAC - IETR/CentraleSupélec
Thèse N° : 2022CSUP0007

Rapporteurs avant soutenance :

Catherine ACHARD Professeure, Sorbone Université, Paris, France
Mohamed DAOUDI Professeur, IMT Nord Europe, UMR9189 CRStAL, Villeneuve d'Ascq, France

Composition du Jury :

Président : Nicolas COURTY Professeur, Université Bretagne Sud, IRISA, éq. Oblelix, Vannes, France

Examinateurs :
Monique THONNAT Directrice de Recherche INRIA, INRIA, Sophia Antipolis, France
Nicolas COURTY Professeur, Université Bretagne Sud, IRISA, éq. Oblelix, Vannes, France

Dir. de thèse : Renaud SEGUIER Professeur, CentraleSupélec, IETR, éq. AIMAC, Cesson-Sévigné, France

Co-encadrants de thèse :
Catherine SOLADIE Professeure Assistante, CentraleSupélec, IETR, éq. AIMAC, Cesson-Sévigné, France
Jérôme ROYAN Principal Scientist, IRT B<>COM, Cesson-Sévigné, France
Nam-Duong DUONG Research Engineer, IRT B<>COM, Cesson-Sévigné, France

I would like to dedicate this thesis to my mother.

Acknowledgements

I am humbled to express my heartfelt gratitude to everyone who has played an instrumental role in my journey toward obtaining a doctorate in artificial intelligence and computer vision. Three years ago, I embarked on this journey to broaden my knowledge and gain valuable experiences. Along the way, I was fortunate to make new friends, companions, and mentors who have enriched my life in numerous ways.

Foremost, I would like to extend my sincerest appreciation to my supervisors, Assoc. Prof. Catherine Soladié, Dr. Nam-duong Duong, Dr. Jérôme Royan, and Prof. Renaud Séguier, for their unwavering guidance and support during the past three years. They gave me an incredible opportunity to work on this interesting project and wholeheartedly instructed me to walk the path of scientific research efficiently. Without their support and encouragement, I would not have been able to complete this journey successfully.

Moreover, I am grateful to my colleagues at IRT b<>com, especially the ARCloud team at Immersive Iteration laboratory, and my friends at the AIMAC team in CentraleSupélec. Working alongside such talented individuals in various research projects was a privilege, and their contributions were crucial to my academic success. I am also thankful for the knowledge I gained from them about French culture and language. Special thanks to Amine Kacete, who has been dedicated to supporting me.

I am also indebted to my family in Algeria for their unwavering love and care. Despite being thousands of kilometers away, their constant support and encouragement gave me the strength to overcome challenges and continue my studies. My most profound appreciation goes to my mother, who not only gave birth to me and raised me but also inspired me to pursue this thesis.

Abstract

Humans use their hands for various tasks in daily life and industry, making research in this area a recent focus of significant interest. Moreover, analyzing and interpreting human behavior using visual signals is one of the most animated and explored areas of computer vision. With the advent of new augmented reality technologies, researchers are increasingly interested in hand activity understanding from a first-person perspective exploring its suitability for human guidance and assistance. Our work is based on machine learning technology to contribute to this research area. Recently, deep neural networks have proven their outstanding effectiveness in many research fields, allowing researchers to jump significantly in efficiency and robustness.

This thesis's main objective is to propose a user's activity recognition framework including four key components, which can be used to assist users during their activities oriented towards specific objectives: industry 4.0 (e.g., assisted assembly, maintenance) and teaching. Thus the system observes the user's hands and the manipulated objects from the user's viewpoint to recognize his performed hand activity. The desired framework must robustly recognize the user's usual activities. Nevertheless, it must detect unusual ones to feedback and prevent him from performing wrong maneuvers, a fundamental requirement for user assistance. This thesis, therefore, combines techniques from the research fields of computer vision and machine learning to propose comprehensive hand activity recognition components essential for a complete assistance tool. This work explores recent advances in neural network algorithms, which have proven to be effective for classic closed-set supervised learning problems when a large amount of data is available. However, to detect unusual (unknown) activities, the system must be learned in an open-set setting, a more realistic and challenging scenario where incomplete knowledge of the world exists during training time, and unknown activities can be seen during testing. These detected unknown activities are collected, automatically annotated, and then incorporated into the system. In this way, the desired framework learns and gradually expands by making each new activity known to the system. Much of the work was also devoted to minimizing computation and data acquisition costs, resulting in cost-effective and easily adaptable components.

Résumé

Les êtres humains utilisent leurs mains pour diverses tâches dans la vie quotidienne et professionnelle, ce qui fait que la recherche dans ce domaine a récemment suscitée un grand intérêt. De plus, l'analyse et l'interprétation du comportement humain à l'aide de signaux visuels est l'un des domaines les plus actifs et les plus explorés de la vision par ordinateur. Avec l'arrivée des nouvelles technologies de réalité augmentée, les chercheurs s'intéressent de plus en plus à la compréhension de l'activité de la main d'un point de vue de la première personne, en explorant la pertinence de son utilisation pour le guidage et l'assistance humaine. Notre travail est basé sur la technologie de l'apprentissage automatique pour contribuer à ce domaine de recherche. Récemment, les réseaux neuronaux profonds ont prouvé leur efficacité exceptionnelle dans de nombreux domaines de recherche, permettant aux chercheurs de faire un bond en avant en matière de robustesse et d'efficacité de manière significative.

L'objectif principal de cette thèse est de proposer un système de reconnaissance de l'activité de l'utilisateur incluant quatre composants essentiels, qui peut être utilisé pour assister les utilisateurs lors d'activités orientées vers des objectifs spécifiques : industrie 4.0 (par exemple, assemblage assisté, maintenance) et enseignement. Ainsi, le système observe les mains de l'utilisateur et les objets manipulés depuis le point de vue de l'utilisateur afin de reconnaître et comprendre ses activités manuelles réalisées. Le système de réalité augmenté souhaité doit reconnaître de manière robuste les activités habituelles de l'utilisateur. Néanmoins, il doit détecter les activités inhabituelles afin d'informer l'utilisateur et l'empêcher d'effectuer de mauvaises manœuvres, une exigence fondamentale pour l'assistance à l'utilisateur. Cette thèse combine donc des techniques issues des domaines de recherche de la vision par ordinateur et de l'apprentissage automatique afin de proposer une brique de reconnaissance de l'activité de l'utilisateur nécessaire à un outil d'assistance complet. Ce travail explore les avancées récentes des algorithmes de réseaux de neurones, qui se sont avérés efficaces pour les problèmes classiques d'apprentissage supervisé à ensemble fermé lorsqu'une grande quantité de données est disponible. Cependant, pour détecter des activités inhabituelles (inconnues), le système doit être appris dans un ensemble ouvert, un scénario plus réaliste et plus difficile où la connaissance du monde est incomplète au moment de l'apprentissage et où des activités inconnues peuvent être observées pendant le test. Ces

activités inconnues détectées sont collectées, annotées automatiquement, puis intégrées aux modèles. De cette façon, le système de reconnaissance de l'activité de l'utilisateur proposé apprend et s'étend progressivement en faisant connaître chaque nouvelle activité au système. Une grande partie du travail a également été consacrée à la minimisation des coûts de calcul et d'acquisition de données, ce qui a permis de proposer un système rentable et facilement adaptable et industrialisable.

Table of contents

Chapter 1

Introduction

Contents

This thesis takes place at the Institute of Research and Technology irt b-com in collaboration with CentraleSupélec Rennes, the Artificial Intelligence for Multimodal Affective Computing (AIMAC) team part of the Institute of Electronics and Telecommunications (IETR) of Rennes. The thesis is a part of the irt b-com ARCloud project. This chapter first introduces the general context in section 1.1. Section 1.2 then presents the scientific research hypotheses, the objectives, and challenges. Section 1.3 presents the contributions of this thesis, and section 1.4 provides the organization of the following chapters of this manuscript.

1.1 Context and background

In recent years, the development of the Internet of Things has allowed digital technologies to be used in both everyday contexts as well as in the world of "industry 4.0." With the release of more compact, more powerful, and lighter Augmented Reality (AR) visualization devices, manufacturing companies are reinvesting in AR and exploring its suitability for the industrial context. To reduce costs and production times, manufacturers are searching for solutions to inconveniences associated with the interruption of production machines, e.g., due to maintenance reasons, specifications of certain products.

The training of operators is one part of the production process where AR can be integrated to save time and money. Appropriate and adequate training of industrial operators can be less costly than a reconfiguration of the whole production process. Because of their flexibility, operators can respond quite easily and quickly to the particular needs of the manufacturer. Their training often requires another sufficiently experienced operator and teacher to transmit the required skills. In the case that such a person is not available, instruction manuals can be used to train operators. However, they can sometimes be more cumbersome, outdated, insufficiently informative, and not very comprehensive. These manuals may not meet the specificities of an apprenticeship nor the technical level of the apprentices. In order to be effective, complete, and lasting, the training procedure in an industrial context requires our different senses, notably sight, hearing, and touch. The best way to learn appropriately and effectively acquire new knowledge is to visualize the task to be performed and to experience it. The AR can meet these requirements by displaying relevant information at the right time in the right place. In figure 1.1, the right picture shows a demonstration of operator teaching using AR device.

On the other hand, assembly is crucial for the entire manufacturing process. The total cost of a product, the time required to manufacture it, and its quality depend on the efficiency and precision of the various assembly stages. Assembly operations can often be complex and require fine adjustments to achieve an acceptable result. The assembly sequence can be long, with many parts to be assembled in a precise order to ensure the proper functioning of the product. For these reasons, workers must be skilled and trained to do so within the cycle time imposed by the production rate. The final product may depend on the variant to be assembled and may require the consultation of paper manuals. These reference tables can lead to time loss, distraction, and safety problems. In manual assembly, tasks are performed by human operators assisted by semi-automatic tools or machines. However, human error is also a fundamental problem in assembly lines. They can result in increased production waste or processing time and costs and degraded product quality due to manufacturing defects. Various methods, such as extensive training sessions or detection devices, are used to overcome this

Fig. 1.1 The left picture gives an illustration of an AR operator being assisted by the AR assistance system while performing an assembly task. The right picture shows an illustration of operator teaching using an AR system. AR projects the learner into a context closer to reality than traditional learning methods. It makes the task easier for the assembler by more pleasantly displaying the instructions to be followed.

problem. These approaches are often costly and, in many cases, do not provide complete assurance of avoiding these annoyances. The likelihood of human error during assembly must be reduced to avoid potential damage to the entire production system. To this end, the need to combine traditional manual assembly with a tool capable of improving the effectiveness and efficiency of the process, such as an AR system, becomes obvious. The operator must be supported and guided in his activities, e.g., by performing the assembly operations while being equipped with an intelligent AR assistant system. Such an AR system may prevent the operator from diverting his attention and not being distracted from the process, which helps perform tasks efficiently. In figure 1.1, the left picture shows an illustration of an operator being assisted with an AR system while preforming his assembly activity.

An intelligent AR system may be particularly well suited to address these issues. In this regard, understanding the operator's activities from a first-person viewpoint is fundamental in such an AR system. Understanding performed activities in an AR context allows the worker to interact with the natural environment and virtual information while guided and receiving feedback. This may give the ability to replace traditional teaching with AR-based interactive instructions and paper manuals with AR-based assistance. Motivated by all these observations, this thesis focuses on the first-person hand activity understanding and its related challenges. The following section will give more details about the thesis objectives.

1.2 Objective and challenges

Attempting to address issues discussed in the previous section, this thesis work was devoted to the design of the main components of a comprehensive framework that recognizes the activities of AR users to assist them in their complex activities. The desired user activity recognition framework must robustly recognize usual activities based on the first-person viewpoint. Moreover, it must detect unusual ones to allow preventing the user from performing wrong maneuvers, a fundamental requirement in teaching and user assistance use cases.

To this end, we based our research on Machine Learning (ML), and Computer Vision (CV) approaches as recommended state-of-the-art choices. Much of the work was devoted to minimizing computational and data acquisition costs, leading to a low-cost and easily adaptable components. The targeted user activity recognition framework is expected to allow being learned on a limited amount of data, e.g., it can be first trained on a publicly available dataset, then quickly adapted to another private use case, such as an industrial application. This significantly reduces the cost of annotated data acquisition and extends the range of applications to cover different industrial domains.

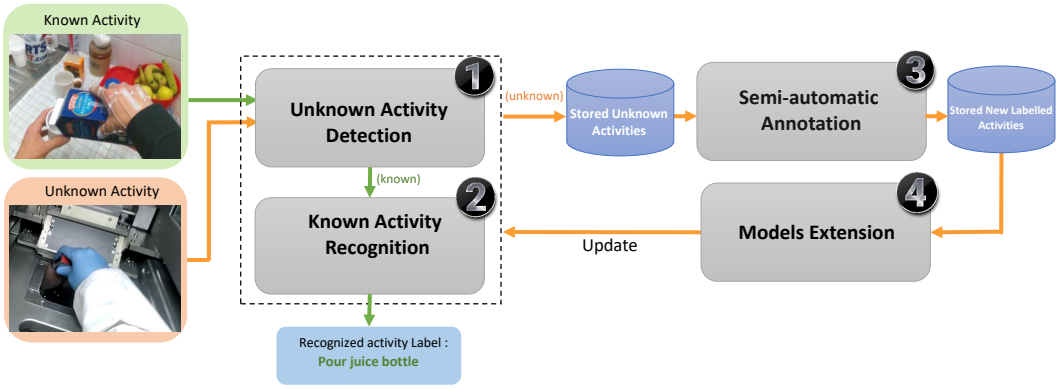

Fig. 1.2 The four components of the targeted user activity recognition framework. We give two examples of hand activities: one framed in green, known to the system, and placed in the class *"pour juice bottle"*; a second activity framed in orange, unknown to the system, which is identified as unknown. Detected unknown activities are stored to be annotated in the third component and reintegrated into the system by the final component. Thus the system will be able to recognize it in the future.

The targeted framework consists of four main components. Figure 1.2 illustrates the execution of these four components on given two hand activities known and unknown (i.e., previously seen or unseen in the recognition learning procedure, respectively). In the first

component, the activity is checked if it is known or unknown. Thus, if the activity is identified as known, the associated class label will be given by the second component that classifies the known activities. Else, if the activity is unknown, it will be stored. Once a certain number of detected unknown activities is reached, they will be semi-automatically annotated in the third component. Finally, the annotated unknown activities will be integrated into the recognition model in the final component. Thanks to this final component, the annotated unknown activities will be handled as known and classified in the future. The following subsections describe each component of the targeted framework and its related challenges.

1.2.1 First-person hand activity recognition

As humans, we are able to recognize the activity of *"cutting potatoes"* presented in figure 1.3, in which we see one frame of an activity video. The activity is filmed from the first-person viewpoint, as though we are the ones performing it. Our understanding of the activity is perfect without difficulty by simply looking at the hands and the objects they are manipulating. Furthermore, we would likely be able to repeat the same activity, even if we had never cut potatoes before, by looking at the hands, the grasp on the objects, and how the person handles them. This process of understanding human activities is a task that the desired framework should be able to accomplish.

Fig. 1.3 Someone is cutting potatoes from an egocentric viewpoint. We can recognize the activity by looking at the hands and the object they are manipulating.

We refer to activity recognition as classifying an activity assuming that the activity video is temporally segmented and defined by a temporal interval, where the start and the end instants are known. Otherwise, the problem is usually referred to as online activity detection, whereby one must determine where and when the action is occurring and identify the activity class. This thesis focuses on the offline activity recognition of temporally localized and

defined activities. More precisely, we focus on hand activity recognition as a sub-problem of egocentric activity recognition (EAR), where activities are supposed to be performed with the hands.

In this regard, this component aims at recognizing hand activities from the first-person viewpoint. This particular perspective, also called the egocentric viewpoint, provides a good field of vision that covers the user's hand activities and allows observing hand-object interactions. Moreover, it perfectly meets the AR context requirements and the actual AR devices. In this thesis, we mainly orientated our research focus on supervised learning classification approaches to solve the recognition task. More precisely, we used artificial neural network algorithms to learn classifying activities. Thus, a recognition neural network model is trained and evaluated using annotated hand activity samples. The main challenges we faced during this component are the following:

- **Recognition Accuracy.** Human activity recognition presents challenges such as intra-class variation and across-class similarities. Intra-class variations include viewpoints, actor styles, aspects, and execution speeds. Across-class similarities occur when different action categories share similar characteristics, such as motion or objects involved, e.g., *"Open juice bottle"* and *"Close juice bottle"* where the involved object *"juice bottle"* remain the same while the actions *"open"* and "close" differs. On the other hand, as another essential characteristic of a first-person viewpoint, the wearable sensor is not fixed, and abrupt motion might appear, which could complicate the application of standard state-of-the-art pertinent feature learning methods.

  With all these challenges, providing an accurate and generalizable recognition model applicable to real-world scenarios becomes difficult, which requires a large amount of data and carefully designed neural network architectures.

- **The data scarcity problem.** It is one of the most critical problems we faced in this work. Since annotated samples in real-world activity recognition applications are usually difficult or expensive to obtain, they require human expert annotators' efforts. This is a prevalent problem in most supervised machine learning-based applications.

- **Computational cost.** The purpose of the recognition model is to be used in real-time applications. This requires a fast inference and light model that can be embedded in AR devices with limited computational resources.

1.2.2 Unknown hand activity detection

As discussed in section 1.2, the desired framework is expected to recognize usual activities. Nerveless, it also should be capable of detecting unusual ones, which are unknown or, in other terms, previously unseen by the recognition model during the training procedure. This allows guiding and providing feedback to the user, e.g., to warn of bad maneuvers in the case of assembling assistance usage. Moreover, the detected unknown activities are exploited in the next component of the targeted framework to overcome the data scarcity problem introduced in the previous section 1.2.1.

To this end, an unknown activity detection model can be learned based on the trained recognition model and its initial training dataset. We list the following main challenges encountered in this unknown hand activity detection component:

- **The detection of unknown activities.** Traditional supervised learning aims to train a recognition in the closed-set world, where training and test samples are annotated and share the same label space. In our case, the recognition is performed in the open-set world, where there exists test samples from the classes that are unseen during training procedure. These unseen samples are expected to be identified, which is a more realistic and a very challenging state-of-the-art problem.

- **Balance between activities recognition and unknown activities detection.** This is the most important problem that we have faced in this step. It is also a well-known problem in open-set learning setting. Effectively, the learned model must perfectly detect unknown activities without deteriorating the performance of the known activities recognition.

1.2.3 Semi-automatic unknown activity annotation

This component aims to handle and exploit the detected unknown activities. To this end, these detected activities are semi-automatically annotated based on unsupervised learning approaches. First, the unknown activities are regrouped automatically by exploiting the knowledge from the recognition model. Next, a meaningful label is assigned by a human for each activity group. This results in annotated activities. The main challenges that we have faced in this component are:

- **Unsupervised hand activities clustering.** Clustering hand activities without prior knowledge (supervision) is still a challenging problem, especially when insufficient data is available. This is due to hand activity's high intra-class variation and inter-class similarity.

- **Providing a clustering-friendly manifold.** To facilitate the unknown hand activities clustering, which is a very challenging task, these activities must be mapped into a highly discriminative features space that allows regrouping them based on similarity measures. Providing such a learnable mapping function is one of the challenging problems we have addressed in this step.

1.2.4 Incremental hand activity recognition (models extension)

In real-world applications, the learning is continuous. Thus, the desired user activity recognition framework must be flexible and adaptable for possible future applications. To this end, the resulting annotated activities from the previous component are integrated into the initial recognition model and the unknown activities detection model to be recognized in the future. Incremental learning poses particular challenges for artificial neural networks that we briefly introduce as follows:

- **Catastrophic forgetting.** It is the tendency for knowledge of the previously learned task to be abruptly forgotten as information relevant to the current task is integrated. This phenomenon, termed *catastrophic forgetting*, occurs when the network is trained sequentially multiple times as a new set of annotated activities arrives. This decreases the recognition performance and, consequently, the unknown activities detection performance.

- **Computational cost.** To fight the *catastrophic forgetting* problem, a natural direction is to extend the neural networks progressively by storing training data from previous old classes. However, this technique can be very demanding in terms of memory and computing capacity.

1.3 Contributions

In this work we have tried to overcome all the challenges mentioned in the previous section. This led us to a proposal with the following main contributions summarized:

- **Hand activity recognition solutions.** To address the challenges of hand activity recognition, one of the most important component of the targeted framework, we proposed two solutions for first-person hand activity recognition. In addition to the robustness of the recognition, we also considered the data scarcity and computational cost. The proposed solutions are based on 3D skeletal data and RGB images, respectively:

    1. A new hybrid learning pipeline for skeleton-based hand activity recognition consists of three blocks. First, the spatial features for a given sequence of hand joint positions are rapidly extracted using a specific combination of our local and global hand-crafted spatial features. Then, the temporal dependencies are learned using a multi-stream learning strategy. Finally, a hand activity sequence classifier is learned using our post-fusion strategy and applied to the previously learned temporal dependencies. This multi-stage learning pipeline allows training with a limited number of training samples while ensuring good accuracy, which addresses the problem of data scarcity. Experiments evaluated on two real-world datasets show that our approach performs better than state-of-the-art. For another ablation study, we compared our post-fusion strategy with three traditional fusion baselines and showed an improvement in accuracy.
    2. A novel, low-cost, multi-stage transfer learning pipeline for RGB-based first-person hand activity recognition data addresses the data scarcity problem. The first stage extracts regions of interest for a given RGB image activity sequence using a pre-trained neural network. Unlike existing methods that use visual attention through Deep Learning and require a large amount of data, we propose to directly use the right and left hands as relevant regions of interest that provide information about manipulated objects and performed actions. These regions of interest are extracted using a transfer learning technique. Our experiments have shown that this information is key to first-person hand activity recognition. We propose a data augmentation procedure tailored to these regions of interest to strengthen the recognition model. Then, high-level spatial features are extracted in the second stage using a pre-trained deep neural network. In the third stage, temporal dependencies are learned. Finally, a hand activity sequence classifier is learned in the last stage by applying a post-fusion strategy to the previously learned temporal dependencies.

Adapting transfer learning allows learning with limited training examples while providing good accuracy. It also reduces the training cost since the transferred neural network is already pre-trained.

Experiments evaluated on two real-world datasets show that our pipeline achieves state-of-the-art performance. Moreover, the proposed pipeline achieves good results even on limited data.

We also experimented the combination of both, 3D skeleton hand joints and RGB images based pipelines, which significantly improves the accuracy of hand activity recognition.

- **Unknown hand activity detection.** The above-presented solutions for hand activity recognition are based on the classical closed-set recognition paradigm, where train and test activity samples are supposed to be known. However, one of the components of our targeted user activity recognition framework aims to perform recognition in an open-set setting. Thus, it recognizes activities from known classes while, at the same time, detecting and rejecting unknown activities from unknown classes previously unseen during the training procedure. In this regard, we presented in this thesis an adopted consensus-based open-set hand activity recognition that groups three approaches to deciding whether a test activity sample is from a known or unknown class. To increase overall open-set recognition performance, we employ a consensus of three outlier detection approaches and aggregate their decisions via voting.

- **Unlabeled hand activity clustering.** The third component of our targeted framework aims to cluster (categorize) detected unknown activities in order to be annotated. In this regard, we proposed a new and original approach that approximates the unsupervised domain adaptation to cluster unlabeled hand activities. It uses the knowledge obtained from labeled samples of the source domain (the known activities) to categorize the unlabeled samples of the target domain (the detected unknown activities). Thus, we introduced a novel and original metric learning-based loss function to learn a highly discriminative representation while maintaining good recognition accuracy of the activities in the source domain. The learned representation is used as a low-level manifold to cluster unlabeled activity samples. To achieve the best clustering results, we also proposed a statistical and consensus-based strategy for clustering.

- **Incremental hand activity recognition.** The final component of the desired activity recognition framework aims to make the learned models extendable and adaptable for future applications. To this end, the resulting clustered and annotated activities from

the previous component are integrated into the initial recognition and the unknown activity detection models. Thus, with this component, the framework incrementally learns and extends its multi-class classifier, making each new class "known" to the models. To this end, we present our adopted method for incremental hand activity recognition in this thesis. The proposed method is based on fine-tuning with an efficient replay memory method to avoid the problem of *catastrophic forgeting*.

1.4 Thesis outline

The following chapters of this thesis are organized as follows:

Chapter 2 first reviews the state-of-the-art of existing methods for first-person hand activity recognition, which we divide into three distinct categories. We give a description, advantages, and limitations for each category. Then, we present the available state-of-the-art datasets for hand activity recognition evaluation.

The rest of the chapter is devoted to an overview of the state-of-the-art methods related to our contributions. Specifically, we address unknown hand activity detection, unknown hand activity clustering, and incremental learning for hand activity recognition. We analyze how each method works and show its advantages and limitations.

Chapter 3 details our proposed hand activity recognition methods. First, we present our temporal multi-stream learning and post-fusion strategy for 3D skeleton-based hand activity recognition. It is an efficient and cost-effective learning pipeline that falls into the category of hybrid approaches that combine hand-based and deep learning methods. Then, we present our second method for hand activity recognition, which is based on RGB images. It is a new learning pipeline that aims to overcome the problem of data sparsity while providing low-cost, excellent, and accurate recognition. It consists of four sequential steps that mainly use transfer learning techniques.

We explain the functioning and methodology of each proposed method. Then, we present a detailed experiment that includes implementation details, ablation studies, and results compared to the state-of-art methods. Finally, we discuss the results and point out possible future improvements.

Chapter 4 presents our proposed methods that correspond to the rest of the desired activity recognition framework components. First present our adopted method for unknown hand activity detection. Then, we present our proposed method for unlabeled hand activity clustering, which concerns the primary contributions of this thesis. Finally, we present our adopted method for incremental hand activity recognition.

We explain the functioning and methodology of each proposed method. Then, we present a detailed experiment that includes implementation details, ablation studies, and results compared to the state-of-art methods. In the end, we discuss the results and point out possible future improvements.

Chapter 6 concludes with a summary of this thesis and suggests directions for future research.

Chapter 2

State-of-the-art

Contents

2.1 Introduction

As we explained in the previous chapter, section 1.2, the main objective of this thesis is to propose the essential components for a hand activity recognition framework, which can be used to assist AR users. We illustrate these components in the figure 2.1. The first component aims at detecting unknown activities. The second component aims to recognize hand activities from the first-person point of view. The third component clusters and annotates detected unknown hand activities. Finally, the last component integrates these annotated activities into the recognition and detection models.

By exploring existing related work, we concede that the general idea of our objectives falls in the fields of lifelong learning [30] or open-world recognition [12]. Following *Bendale et al.* [12], the open-world recognition system must recognize objects and associate them with known classes while also detecting unseen classes as unknown. These "novel unknowns" must then be collected and labeled (e.g., by humans). When there are sufficient labeled unknowns for new class learning, the system must incrementally learn and extend the multi-class classifier, thereby making each new class "known" to the system. Open-world recognition moves beyond being robust to unknown classes and towards a scalable system adapting itself and learning in an open world.

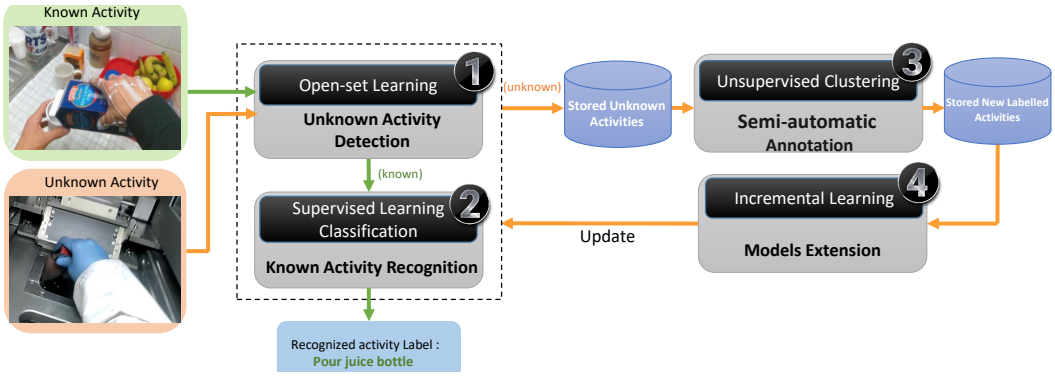

Fig. 2.1 The four principal components of the desired hand activity recognition framework introduced in section 1.2. The black box on each component represents the corresponding related research domain.

Each of the four components can be considered an open-world recognition step, representing a complex and extensive research area: open-set learning, supervised learning classification, unsupervised clustering, and incremental learning. Therefore, in this chapter, we describe each component's general idea and present its related existing state-of-the-art approaches. This chapter is organized as follows:

- **Section 2.2** begins with an introduction to first-person hand activity recognition. Then, we present related state-of-the-art methods, such as dynamic hand gesture recognition and human activity recognition. The introduced methods are categorized according to their input data type and the adapted techniques, deep learning, hand-craft, or hybrid-based techniques. Finally, we present the widely used datasets and conclude the section.

- **Section 2.3** presents state-of-the-art methods related to the detection of unknown hand activities. Therefore, we first introduce and formally define the open-set recognition concept, a fundamental requirement for unknown sample identification. Then we present two categories of existing methods, traditional ML-based and DNN-based open-set recognition methods. Finally, we overview the state-of-the-art open-set action recognition methods and conclude the section.

- **Section 2.4** gives related work to our targeted framework step of automatically annotating detected unknown activities. This step requires using prior knowledge and adapting the learned models to facilitate clustering these unknown samples. Thus we present the Unsupervised Domain Adaptation (UDA) paradigm and deep Metric Learning state-of-the-art related approaches. We then, introduce a selected set of most used clustering algorithms and their evaluation metrics. Finally we conclude the section.

- **Section 2.5** presents the state-of-the-art related approaches to the last step of our targeted framework. Thus, we give an overview of incremental learning existing methods and discuss their advantages and disadvantages.

- **Section 2.6** concludes the state-of-the-art chapter. We highlight relevant research paths that can lead us to solve issues related to our objectives.

2.2 Hand activity recognition

Understanding first-person hand activity is a challenging problem in computer vision that has attracted much attention due to its extensive research and practical applications, such as Human-Computer Interaction [190], Humanoid Robotics [152], Virtual/Augmented Reality [198], and Multi-media for automated video analysis [8]. It aims to recognize the hand activity performed by the user (the wearer of the sensor) from the first-person viewpoint to approximate its field of view. The hand activity is supposed to be performed by hand while manipulating objects (figure 2.2). These objects can be static or dynamic and can be deformable or rigid.

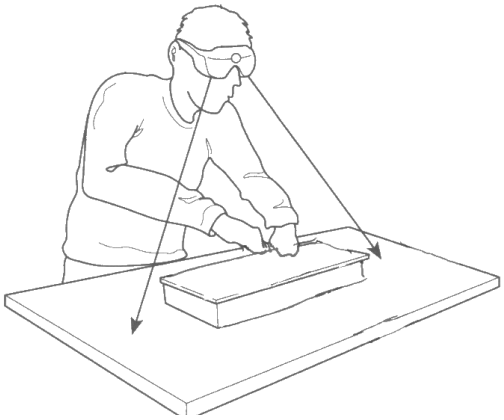

Fig. 2.2 Illustration of an operator using an AR device while performing a hand activity. The picture shows the first-person viewpoint that covers the hands and the manipulated object.

This thesis focuses on analyzing and recognizing AR user hand activities performed with one or two hands. This recognition aims to study the spatial components of the hands, the possible manipulated objects, and their temporal evolution to determine the activity class. It is, therefore, a classification problem. We found two types of data commonly used to analyze activity sequences for a classification task, videos streams and skeleton data:

Video streams. Whether color, depth, or optical [15], are the most easily accessible data or particularly require very little pre-processing in the case of the RGB color stream. Primarily, they are accessible to an extensive range of potential users. The RGB cameras that make up the vast majority of smartphones and depth cameras are, nowadays, very inexpensive and accessible types of equipment.

Skeleton data. Sequences of the skeleton of the hand constitute much more refined and precise information, allowing to obtain the temporal evolution of each hand joint. However, these sequences require either heavy or expensive pre-processing methods. This also requires

an additional computation time for the classification algorithm, e.g., DeepPrior [128] which estimates the hand skeleton from a depth stream. The acquisition of this type of data still requires expensive or not very accessible hardware like Leap Motion cameras which extract the skeleton from infrared sensors or magnetic sensors attached to each of the hand joint to track their movements.

These input data are then analyzed and classified using methods that we categorize as Deep Learning (DL) based, Hand-Crafted (HC) based, and hybrid approaches. Figure 2.3 illustrates the proposed categorization. The feature extraction phase is quite distinct from the classification phase when using HC-based methods since the classification phase is usually based on classical machine learning techniques such as support vector machines (SVM) [10] or decision trees [11]. In contrast, when we talk about DL-based methods, we refer to methods where these two phases of feature learning and classification are grouped within one neural network used, which processes both phases simultaneously in an end-to-end manner. Hybrid methods, combine DL-based and HC-based methods.

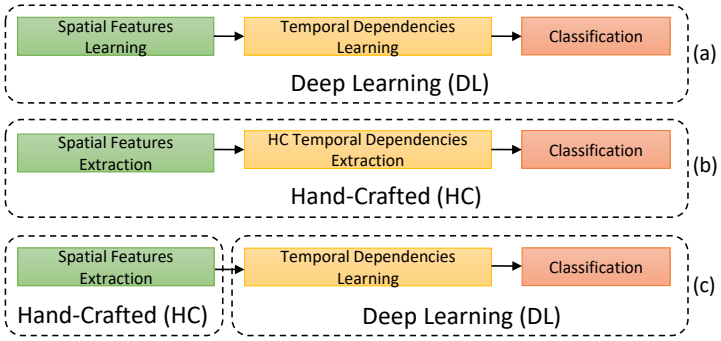

Fig. 2.3 A categorization of activity recognition methods. (a) Deep learning based methods. (b) Hand-crafted based methods . (c) Hybrid methods that combine both Deep learning and Hand-crafted methods.

In addition to the motivation of designing algorithms with better classification results, there is a need for lighter and less computationally expensive solutions, especially to justify the application of these algorithms in real-world use cases. This requirement is one of our concerns in this state-of-the-art study.

The following sections present related work for first-person hand activity recognition. We also introduce other similar activity recognition approaches: dynamic hand gesture recognition and human activity recognition. Following our categorization, sections 2.2.1, 2.2.2, and 2.2.3 introduce state-of-the-art DL-based , HC-based and hybrid methods, respectively. Section 2.2.4 introduces available hand activity recognition and related task datasets.

2.2.1 Deep Learning based methods

Deep learning methods have recently become very popular in many application areas, including human activity recognition [219]. In particular, since the beginning of 2010, the AlexNet neural network [88] has beaten all the HC based methods usually used for the ImageNet dataset classification task [164].

Convolutional Neural Network (CNN). *Lecun et al.* [92] were the first to introduce CNNs, allowing the usage of deep learning methods on image tasks, notably by extracting and recognizing patterns via convolution operations. The outstanding performance of CNNs in image classification [231] has motivated [19, 20] to formulate the recognition problem as an image classification problem by representing a sequence of 3D skeleton joints as a 2D image input for a deep CNN (Figure 2.4). However, this conversion causes an inevitable loss of data.

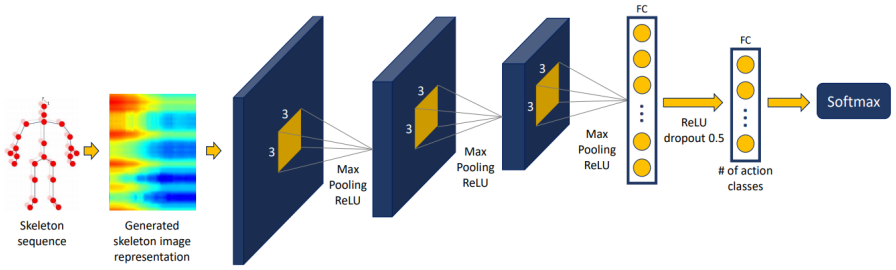

Fig. 2.4 CNN architecture employed for 3D action recognition [19].

Also, attracted by the success of CNNs, [96] proposed CNN-based neural network architecture for activity detection and classification. First, the raw skeleton coordinates, as well as skeleton motion, are fed directly into transformer modules that are designed to rearrange and select important skeleton joints automatically. Then, the output of each transformer module is fed into a CNN layer. Finally, as shown in figure 2.5, the outputs of the two-stream CNN are concatenated and followed by a Fully Connected (FC) layer equipped with Softmax loss for the classification.

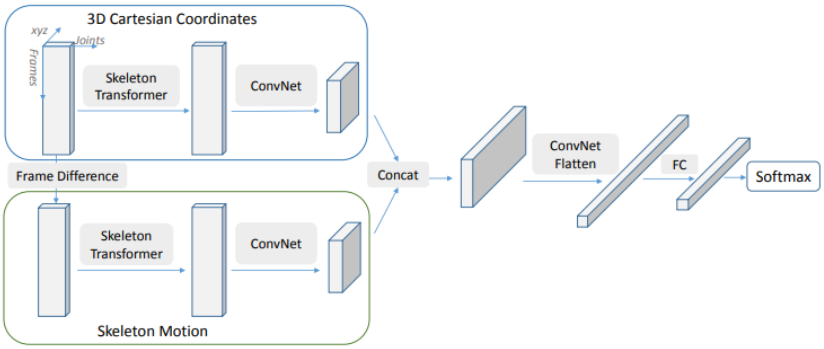

Fig. 2.5 CNN representation of skeleton sequences for action classification [96].

To perform recognition based on RGB sequences instead of 3D skeleton data, *Wang et al.* [56] proposed to cut the activity video into multiple windows of consecutive frames of RGB images. A spatial stream composed of a CNN analyzes each window. Simultaneously, the same windows are analyzed on the optical flow using a second temporal stream consisting of another CNN. This technique allows the architecture to classify the activity simply by merging the classification result of all the windows in the activity video.

Luvizon et al. [111] proposed a 2D CNN-based deep neural network to perform a multi-task prediction from an image sequence. They used an Inception-V4 [199] based architecture to estimate 2D/3D human poses and classify activities. Their architecture extracts low-level visual features from the input color video and probability maps. These two data act as an attention mechanism and thus allow to focus on pertinent regions of image frames. Their network also allows, by regression, to determine the 2D or 3D coordinates of the observed human skeleton. These two data are finally assembled to predict the class of the video as illustrated in figure 2.6.

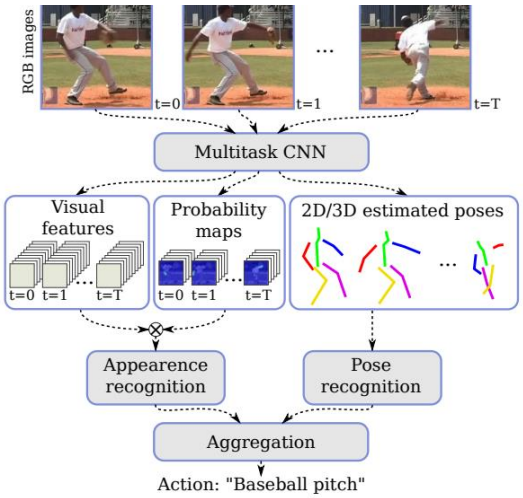

Fig. 2.6 2D/3D Pose estimation and action recognition using multitask deep neural networks [111].

Recently, *Singh et al.* [185] proposed a two-stream DL architecture, 2D and 3D CNNs fed by egocentric cues (hand mask, head motion, and saliency map). The two-streams networks are followed by a class score fusion strategy to classify activities as shown in figure 2.7. To make use of the temporal dimension, they added a temporal stream that uses stacked optical-flow as an input to capture motion information. Hence, these egocentric cues are not always available.

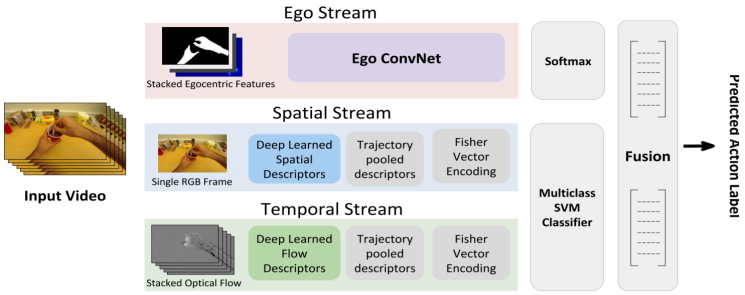

Fig. 2.7 Activity recognition neural network architecture proposed by [185].

Similarly, *Man et al.* [113] proposed a two-stream architecture: An appearance stream for an object classification task by applying hand segmentation and object location; And a motion stream for action classification using optical-flow. The activity class label is given by concatenating the action and the object class labels. However, a heavy manual data annotation was necessary for object region localization and hand segmentation. Moreover, they just used a single RGB image for encoding appearance without considering the temporal ordering. As an alternative to optical-flow-based motion information, which is also interpreted as

temporal dependencies features, *Ryoo et al.* [166] extracted features from a series of frames to perform temporal pooling with different operations, including max pooling, sum pooling, or histogram of gradients. Then, a temporal pyramid structure encodes both long-term and short-term characteristics. However, these methods do not consider the temporal order of activity sequence frames.

Graph Neural Network. Other recent works focused on the Graph Convolutional Networks (GCNs), mainly applied for 3D skeleton data to exploit the connections between the skeleton joints following their physical structure. These connections are represented as a matrix that allows convolutions operations. Based on this concept, *Yan et al.* [233] proposed an end-to-end Spatial Temporal Graph Convolutional Networks (ST-GCN). Given a sequence of 2D or 3D coordinate joints, they construct a spatial-temporal graph with joints as graph nodes. Connectivities in both human body structures and time are represented in blue and green graph edges, respectively, in Figure 2.8. The layers of the ST-GCN operations are applied to the input data generating high-level feature maps on the graph, which facilitates the activity classification.

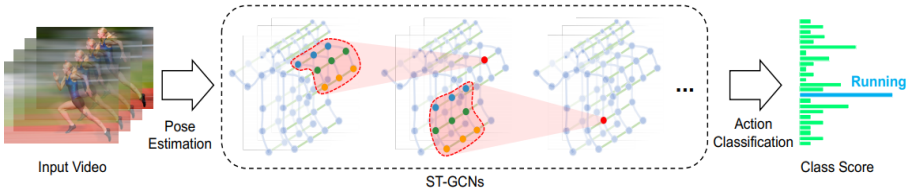

Fig. 2.8 Spatial Temporal Graph Convolutional Networks (ST-GCN) for skeleton-based activity recognition [233].

Similarly, *Li et al.* [99] proposed an Actional-Structural Graph Convolutional Networks (AS-GCN). Unlike ST-GCN, it considers not only the connections between joints directly connected via bones but also captures the latent dependencies between any joints via their proposed actional and structural links, the (A-links) and (S-links) modules, respectively. For example, hands and feet are strongly correlated while walking, which can be pertinent graph connections. Figure 2.9, shows the difference between AS-GCN and ST-GCN pertinence of feature maps.

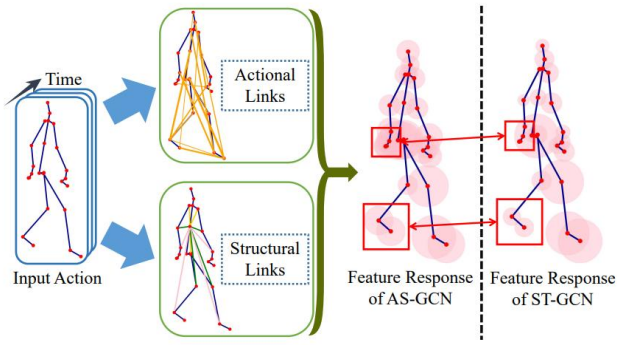

Fig. 2.9 Feature learning with generalized skeleton graphs. The actional links and structural links capture dependencies between joints. For the action *"walking"*, actional links denotes that hands and feet are correlated. The semilucent circles on the right bodies are the joint feature maps for recognition [99].

We also find the use of this type of architecture to solve dynamic hand gesture recognition. *Zhang et al.* [246] used GCNs focusing on the spatial aspect of the hand, which is composed of all the joint connections. They employed a two-stream learning paradigm: temporal graphs focused on hand joint evolution and a spatial graph focused on the displacement of these joints. Finally, each GCNs stream is fed into a FC layer. The two outputs of the FC layers are concatenated and followed by a softmax layer to perform classification as shown in figure 2.10.

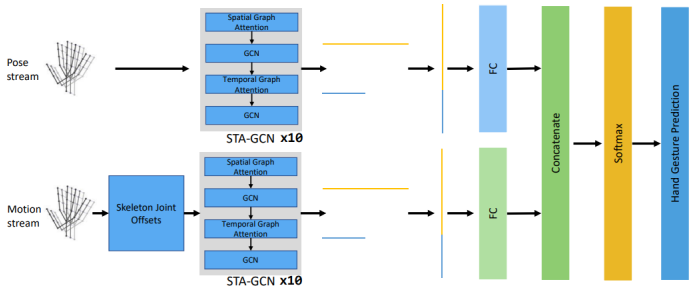

Fig. 2.10 Illustration of the two-stream graph convolutional network with spatial-temporal attention [246]

Unlike [246], *Li et al.* [95] exploited a recurrent bi-directional network to process the temporal aspect of the hand skeleton joints. In contrast, the analysis of the spatial component of the skeleton is done using GCNs architecture in a two-stream manner, as shown in figure 2.11. An attention mechanism is added to the graph architecture to mutually explore the spatial relationships between all the hand joints.

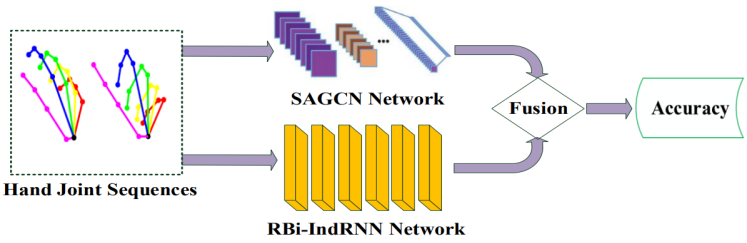

Fig. 2.11 *Li et al.* [95] proposed approach for gesture recognition based on the hand skeleton and using two branches, one to analyze the spatial component, the second to analyze the temporality.

However, these methods do not focus on a particular approach considering the hand joints, their location, and their displacement as mutually related. Especially the hand is a relatively small object whose relations between joints are pretty direct. Moreover, GCNs are ineffective in learning long-term temporal dependencies and are very sensitive to noise.

Visual Attention. A particular branch of DL approaches focused on observing and exploring spatial attention through a deep neural network to recognize activities based on visual information [195, 197]. Recently, Transformer [211] has attracted much attention because of its ability to learn temporal dependencies in conjunction with visual attention. It applies a self-attention mechanism that directly models the relationships between all temporal elements in a sequence, regardless of their respective positions. This allowed Transformer to handle longer sequences than RNNs.

Li et al. [100] proposed a Transformer-based RGB-D egocentric action recognition method called Tear that consists of two modules, an inter-frame attention encoder and mutual-attentional fusion block, which showed promising results. Hence, the learned spatial attention is not fully confident since it is learned in an unsupervised manner while learning a supervised egocentric activity recognition. This has led some researchers to supervise spatial attention learning by using Gaze information [121] or by manually annotating the data [113] which is more expensive. In all cases, this has confirmed that in first-person hand activity recognition problems, the visual points of interest are concentrated around the hands and manipulated objects. This relevant information can be used to design more robust EAR algorithms.

Recurrent Neural Networks. Many other works focused on Recurrent Neural Networks (RNNs) to better exploit information in the temporal dimension. The main difference between RNN and the feed-forward networks is the presence of feedback loops that produce the recurrent connection in the unfolded network. With the recurrent structure, RNN can

model the contextual information of a temporal sequence allowing learning of long and complex temporal dependencies along the activity sequence. Hence, it is unsafe to train deep RNNs with the standard activation functions, e.g., *Tanh* and *Sigmoid* functions, due to the vanishing gradient and error-blowing-up problems.

Hochreiter et al. [62, 31], proposed the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU) cells, respectively. LSTM and GRU were created as the solution to short-term memory. They have an internal gated circuit that regulates information flow by learning which data in a sequence is relevant to keep or throw away. To improve the long-short term temporal dependencies learning, [178] proposed the bidirectional recurrent neural network (BRNN), which presents the sequence forwards and backward. This has motivated [40] to use LSTM-BRNN hierarchically for skeleton-based action recognition, as illustrated in Figure 2.12.

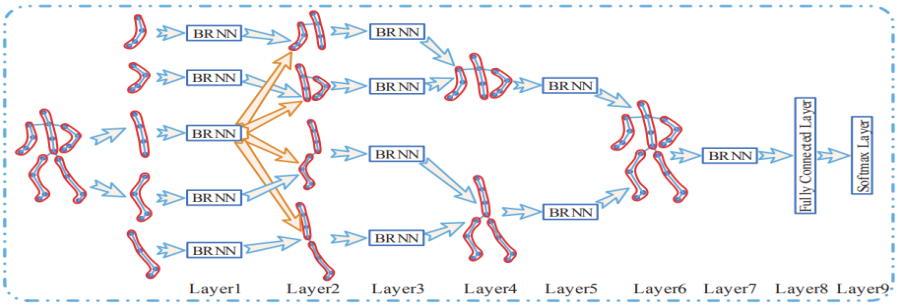

Fig. 2.12 Hierarchical bidirectional recurrent neural network for skeleton-based activity recognition [40].

For better control of the temporal aspect, *Liu et al.*[105] proposed a customized LSTM gates for 3D human action recognition. [114] exploited the GRUs to recognize 3D skeleton-based activities, which showed good results.

On the other hand, *Sudhakaran et al.* [196, 197] proposed Convolutional Long Short-Term Memory (ConvLSTM), which allows reasoning along the temporal dimension to learn the temporal dependencies in respect of temporal order while learning spatial features from image sequences. This has motivated [195] to propose a customized LSTM unit to learn visual attention along the activity sequence jointly with the temporal dependencies. LSTMs fed by learned visual features have been shown to be effective in learning long- and short-term temporal dependencies.

To make use of both the performance of CNNs on spatial features learning and LSTMs on temporal dependencies learning, *Vaudaux-Ruth et al.* [212] proposed a reinforcement learning-based neural network architecture. They used a CNN that takes an image or a video segment as input to learn the spatial features resulting in a low-dimensional feature vector,

which is then forwarded to a GRU layer. The resulting hidden state vector of the GRU is then individually processed by four parallel blocks. The first block deals with the decision of turning the current frame into a spot frame or skipping it (a detected activity or not). The second block predicts the activity class related to the spot frame, and the third block outputs the next video frame. The last block is used to ensure better convergence in the reinforcement learning framework.

The disadvantage of using recurrent neural networks is that they take low-dimensional data as input, and only skeleton data meet these criteria in the case of gesture and action recognition. However, except for particular hardware, such as the Microsoft Kinect or Leap Motion camera, it is mandatory to use pre-processing methods to extract the skeleton data, which adds an extra computing time to the recognition inference.

2.2.2 Hand-crafted methods

Despite the success of DL approaches in the last few years, classical methods that mainly use HC techniques to perform recognition still get attention.

Non-Euclidean manifold representation. It aims to give a highly relevant representation of skeleton data that facilitates the activity recognition task. A reference work was proposed by *Vemulapalli et al*. [213] where they represented the 3D skeleton activity sequence as a curve in the Lie group. Curves are mapped from the Lie group to its Lie algebra, a vector space to classify actions. Finally, the classification is performed using a combination of dynamic time warping, Fourier temporal pyramid representation, and linear SVM.

In [37], a Riemannian manifold is used as a non-Euclidean domain to formulate the recognition problem as a problem of computing the similarity between the shape of trajectories using elastic registration and matching in the shape space. Classification using k-NN is finally performed on this manifold, taking advantage of Riemannian geometry in the open curve shape space. Similarly and besides, *Zhang et al*. [249], projected activity sequences into a Riemannian manifold by using positive definite (PD) regularized Gram matrices of their Hankel matrices. Thus, the classification can be done by computing distances between matrices on the PD manifold. In this context, sequences with the same dynamic model are very close, while those corresponding to different dynamics are far apart. Figure 2.13 show the diagram of their proposed approach.

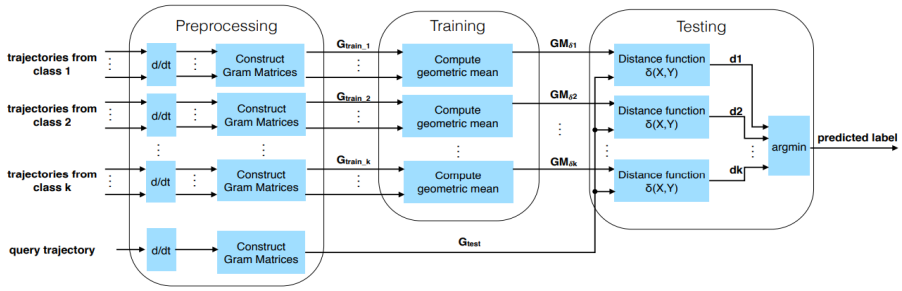

Fig. 2.13 Riemannian manifold representation for skeleton-based activity recognition [249].

Key-points features. In order to exploit the appearance information many works traditionally used local visual features such as HOF [90], MBH [215], 3D SIFT [179], HOG3D [84], and extended SURF [226] to encode appearance information so that it can be used as feature descriptors to recognize activities. The first application of key-point features for EAR was proposed by [243]. Correspondence features are extracted from sequential frames, and matches are removed based on a set of constraints imposed by the camera model (epipolar constraints). Motion histograms are defined and calculated within a frame, defining a new feature called accumulated motion distribution derived from motion statistics in each frame. A SVM classifier is trained with this feature and used to classify activities in different scenes.

Key-point-based feature methods are computationally efficient and can handle large displacements as proven in the related domain, namely Simultaneous Localization and Mapping (SLAM) [123]. Binary descriptors offer faster key-point matching, which is helpful in resource-limited platforms such as embedded devices. However, they perform poorly in first-person videos with poor texture and are often blurred due to the high Egomotion induced by the camera wearer.

Optical flow based methods. Aiming to exploit the motion information, many approaches use optical flow as the primary source of motion features [200]. Optical flow can be obtained using direct motion estimation techniques [71] to achieve frames/sub-frames sub-pixel accuracy resulting in a dense representation. Hence, this representation has a high-computational cost and suffers from redundancy. This has led *Abebe et al.* [2] to propose motion-feature that combines grid (spars) optical flow-based and video-based inertial features. They concatenate features extracted from discriminative motion patterns in the optical flow data, such as magnitude, direction, and frequency. They also include features extracted from virtual inertial data derived from the movement of intensity centroid across frames in a video without inertial sensors. Hence, it suffers from an information leak and has limited discriminative capabilities as specific motion characteristics (e.g., magnitude) are not exploited [200].

3D geometrical representations. Some recent work focused on exploiting the 3D geometrical information. *Smedt et al.* [186] proposed a set of hand crafted (HC) geometrical features based on the connection between the hand joints, namely Shape of Connected Joints (SoCJ) that represent the variation of the hand shape while performing the activity. 3D vectors between physically connected joints are computed from the wrist up to the fingertips for each finger. A fisher vector representation is computed from these SoCJ descriptors. In addition, they proposed two additional HC features, which include histograms of hand directions (HoHD) and rotations (HoWR). Similarly to [43, 242], they used a Temporal Pyramid (TP) representation to manage the temporal dimension. Finally, they used SVM with a linear kernel to classify the dynamic gestures. Figure 2.14 shows the full pipeline of their proposed approach. Similarly, in [130], joint angles similarities and a Histogram of Oriented Gradients (HOG) fed into an SVM to classify activities.

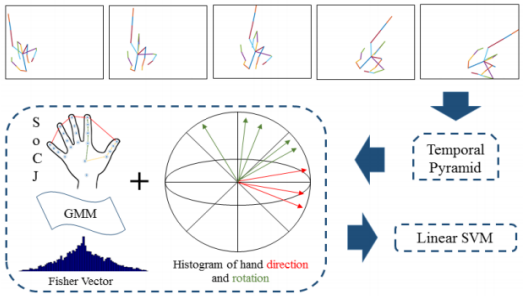

Fig. 2.14 3D geometrical-based representation for skeleton-based activity recognition [186].

This category of methods has been proven to be very effective in providing relevant spatial features, but most of them are still struggling to learn long-term temporal dependencies.

2.2.3 Hybrid methods

This category combines the previously introduced DL-based and HC-based approaches to overcome their limitations. The concept is first to extract a set of HC features from activity sequences and then train a deep neural network on these extracted features to classify the activities. The purpose is to feed the neural network with only relevant features, facilitating its convergence. This allows training with less complex architectures and fewer data and avoids the over-fitting problem.

Avola et al. [5] concatenated a set of HC features as an unified input vector to a deep LSTMs based neural network to classify American Sign Language (ASL) and semaphoric hand gestures. Similarly, aiming at classifying Human 3D Gaits, [108] concatenated relative distances and angles as a feature input vector to a LSTM based neural network to manage

the temporal dimension, while in parallel a CNN is exploited to learn spatial features from 2D Gait Energy Images. Yet, the early fusion of different features spaces (e.g., angles and distances) increases the input complexity and the learning noise [236], especially when only few training data are provided. To avoid the early fusion limitations, *Chen et al.* [29] proposed an end-to-end slow fusion based architecture. Yet, this type of complex architecture requires a lot of data.

2.2.4 Hand activity recognition and related domain datasets

In this thesis, we focused on human activity recognition datasets that involve hand-object interactions. More precisely, the hand activity recognition dedicated datasets. They are almost exclusively acquired from the first-person point of view, which is very suitable for AR applications that naturally offer this viewpoint through AR devices. In the state-of-the-art, we find that the proposed datasets follow two contexts: The first is the study of human grasp from a robotics perspective [21, 158, 18]. These studies mainly focus on the hand's positions to analyze and recognize the grasp type [22]. The second context refers to daily human activity recognition, which is more popular. The daily activities include but are not limited to cooking, sport, person-to-person interaction, and person-object interaction [45, 148, 122]. Table 2.1, shows common human activity recognition datasets acquired from the first-person viewpoint.

RGB images and depth maps are the most common provided data types. For this type of data, we can find large-scale datasets with at least several thousands of training samples, e.g., Something-Something [54], EGTEA [101], or Epic-kitchen [34], which contains several tens or even hundreds of hours of training samples. These datasets are shown to effectively train a stable method for most of the minor variations encountered. Hence, they impose difficulties to be learned on modest configurations, requiring months of training. The hand skeleton [48] or the gaze direction and hand position [101] can also be found, which provide spatial characteristics of the hand (e.g., 3D Cartesian coordinate). These data types are beneficial for hand activity recognition since they allow the analysis of the relationship of joints and their temporal evolution through the activity sequence. Hence, in the case of real-time applications, it will be necessary to extract the same high-level data very fast and robustly, which is still a challenging problem, unlike a simple RGB or depth stream which can be used directly.

Table 2.1 Commonly used egocentric activity recognition datasets.

Dataset	Num of classes	Data type	Data size	Year
GUN-71 [158]	71	RGB, Depth	12,000	2015
UT-Grasp [21]	17	RGB	20	2015
Yale [18]	33	RGB	18,210	2015
GTEA [45]	61	RGB	525	2011
WCVS [122]	10	RGB, Depth	5	2014
Ego-Hand [9]	4	RGB, Depth	48	2015
THU-READ [203]	40	RGB, Depth	1,920	2017
EGETA [101]	106	RGB, Gaz	22Hours	2018
Epic-kitchen [34]	149	RGB	39,596	2018
FPHA [48]	45	RGB, Depth, 3D	1,175	2018
SLS France (our private dataset)	8	RGB	24	2021

In the following, we introduce the dataset that we used to evaluate our proposed methods for hand activity recognition:

- **FPHA dataset.** Proposed by [48]. It is the only publicly available dataset for 3D skeleton-based first-person hand activity recognition. This dataset provides RGB and depth images with the 3D annotations of the 21 hand joints, the 6 Dof object poses, and the activity classes. It is a diverse dataset that includes 1175 activity videos belonging to 45 different activity categories, in 3 different scenarios performed by 6 actors with high inter-subject and intra-subject variability of style, speed, scale, and viewpoint. It represents a real challenge for activity recognition algorithms.

- **Dynamic Hand Gesture (DHG) 14/28 dataset.** Proposed by [186], which is basically devoted to hand gesture recognition. It contains 14 gestures performed in two ways: using one finger and the whole hand. Each gesture is performed 5 times by 20 participants in 2 ways, resulting in 2800 sequences. Sequences are labelled following their gesture, the number of fingers used, the performer and the trial. Each frame contains a depth image, the coordinates of 22 joints both in the 2D depth image space and in the 3D world space forming a full hand skeleton.

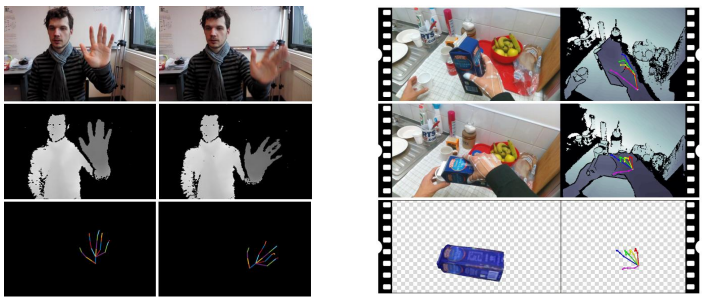

Fig. 2.15 In the left, we show examples from the DHG 14/28 dataset [186]. In the right, we present examples from the FPHA dataset [48].

- **EgoHand dataset.** Proposed by [9]. It has 48 videos recorded with a Google glass. Each video has two actors doing one of the 4 activities: playing puzzle, cards, jenga or chess. These videos are recorded in 3 different environments: office, courtyard and living room. We chose this small dataset to evaluate our methods in case there is not enough training data.

Fig. 2.16 Examples from the EgoHand dataset [9].

- **SLS France dataset.** We have created this private dataset to evaluate our approaches in a more realistic industrial context. The dataset consists of 8 activity classes of a complex cleaning procedure of an industrial production machine. The activities are represented by RGB image sequences acquired with two instruments, an augmented reality headset (MS HoloLens 2) and smart glasses (Vuzix M400). Figure 2.17 shows the used acquisition instruments

Fig. 2.17 Devices used for the SLS France dataset acquisition.

The dataset presents a real challenge for activity recognition since the operator is equipped with protective gloves, which complicate the detection of the hands. In addition, the movement of the operator's head, hands, and handled objects simultaneously makes the activity recognition more challenging. Figure 2.18 shows examples of the proposed dataset.

Fig. 2.18 Examples from the proposed SLS France dataset.

2.2.5 Conclusion

In this section, we presented state-of-the-art hand activity recognition-related methods. The introduced methods are categorized according to their input data type and the adapted techniques: deep learning-based, hand-craft-based, or hybrid-based. Finally, we presented widely used datasets for hand activity recognition and related tasks and our proposed SLS France dataset.

The disadvantage of deep learning-based methods is that, in most cases, the spatial and temporal feature learning parts of the complex neural network act as a black box. Even if the classification can be perfectly correct, it is not easy to know how and which features were extracted by the network. Furthermore, the deep learning method has proven effective when a large amount of data is available. Therefore, it is still difficult and expensive for some industrial applications to provide large-scale labeled datasets due to manual data annotation. Hand-crafted-based methods can be well mastered, allowing debugging and deep analysis,

and can deal with a limited amount of data. However, they still struggle to learn temporal dependencies along the activity sequence. Hybrid methods combine deep learning-based and pure hand-crafted-based methods to overcome their limitations. They can be seen as a tuning alternative between performance and data acquisition cost. In chapter 3, we introduce two hand activity recognition solutions that fall in the category of hybrid approaches.

The next section formulates the problem of detecting unknown activities and presents state-of-the-art existing methods that address this problem.

2.3 Unknown hand activity detection

In the previous section, we introduced state-of-the-art related methods to hand activity recognition. The introduced methods are based on the classical closed-set recognition paradigm, where the train and test activity classes are supposed to be known. However, one of the essential components of our desired hand activity recognition framework aims to detect unknown activities from previously unseen classes during the training procedure. Such a capability allows guidance and feedback to AR users, e.g., to warn of bad maneuvers by the case of assembling assistance usage. Moreover, the detected unknown activities are expected to be exploited in the next component to overcome the data scarcity problem.

Detecting and rejecting unknown activities while recognizing the known ones is a pure open-set recognition problem [52]. It is a realistic scenario where incomplete knowledge of the world exists at training time, and unknown classes can be presented during testing, requiring the classifiers to classify the seen classes accurately and effectively deal with unseen ones [52]. The unknown classes, i.e., classes without any information regarding them during training, are unseen and have no side information (e.g., semantic/attribute information) during the training procedure. For simplicity, let us take an example of a binary image classification of cats and dogs. Classical closed-set image classification and advanced open-set image classification models are trained with the same dataset containing cat and dog images. Figure 2.19 shows that the open-set recognition model can reject the horse image as an unknown while perfectly classifying the cat and dog image samples. However, the classical closed-set classification model cannot reject the unknown horse image sample.

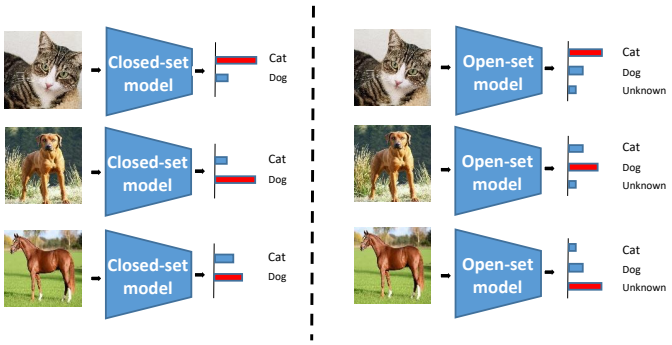

Fig. 2.19 Illustration of closed-set and open-set recognition models behavior during an open-set testing phase. The two models are trained with the same dataset, which contains annotated dog and cat images. In an open-set testing phase, we may have samples from unknown classes (e.g., horse image), which are unseen during the training phase.

2.3.1 Open-set recognition problem formulation

In the open-set recognition, we find three basic notations, which help to formulate the problem: (1) the *open space* that we denote by $\mathscr{O}$. The space of all samples that are far away from the known classes. (2) The *open space risk* that we denote by $R_{\mathscr{O}}$. This risk is incurred when labeling any sample in the $\mathscr{O}$ as an arbitrary known sample. Thus, the more samples from the $\mathscr{O}$ are labeled, the more the $R_{\mathscr{O}}$ incurs. To approximate the quantity of the $R_{\mathscr{O}}$, *Scheire et al*. [176] proposed a qualitative description, where they formalized the risk $R_{\mathscr{O}}$ as the relative measure of open space $\mathscr{O}$ compared to the overall measure space $S_{\mathscr{O}}$, a large ball containing both the positively labeled open space $R_{\mathscr{O}}$ and all of the positive training samples. The quantitative description is formulated as follows:

$$R_{\mathscr{O}} = \frac{\int_{\mathscr{O}} f(x)dx}{\int_{S_{\mathscr{O}}} f(x)dx} \tag{2.1}$$

where f denotes the measurable recognition function, with $f(x) = 1$ indicates that the sample x is recognized, otherwise $f(x) = 0$. (3) the third basic annotation is the *openness* that we denote by O for a particular problem or data. Larger *openness* corresponds to more open problems, while the problem is completely closed when the openness equals 0. *Scheire et al*. [176] formulated this concept by the following definition:

Definition 1. (The openness defined in [176]) *Let C_{TA} , Let C_{TR}, and Let C_{TE} respectively represent the set of classes to be recognized, the set of classes used in training and the set of classes used during testing. Then the openness of the corresponding recognition task O is:*

$$O = 1 - \sqrt{\frac{2\,|C_{TR}|}{|C_{TA}| + |C_{TE}|}} \tag{2.2}$$

where $|.|$ denotes the number of classes in the corresponding set.

The formulation of the *open space risk* and the *openness*, allowed [176] to define the open-set recognition problem as the following minimization problem:

Definition 2. (The open-set Recognition Problem [176]) *Let V be the training data, and let $R_{\mathscr{O}}$, R_{ε} respectively denote the open space risk and the empirical risk. Then the goal of open-set recognition is to find a measurable recognition function f, where $f(x) > 0$ implies correct recognition, and f is defined by minimizing the following open-set risk:*

$$\arg\min_{f} \{R_{\mathscr{O}}(f) + \lambda_r R_{\varepsilon}(f(V))\} \tag{2.3}$$

where λ_r is a regularization constant.

Based on the above theoretical definition of the open-set recognition problem, a series of open-set recognition solutions have been proposed, generally tried to tunning the empirical risk and the open space risk over the space of allowable recognition functions.

The open-set recognition is highly confused with various related techniques that involve a classification with rejection capabilities, such as one-class classification for anomaly detection and zero/one/few-shot learning. These techniques seem to be related in some sense, but they are fundamentally different from the open-set recognition. These methods work under a closed-set assumption. In other words, the corresponding classifier rejects recognizing samples due to low confidence, avoiding false-positive classification.

The introduction of open-set recognition first appeared in face recognition applications, where evaluation datasets contain unseen face instances as imposters that should be rejected [145]. Such open-set protocols are widely used to evaluate face recognition [97]. The generalization of an open-set scenario for multi-object classification was first introduced by *Schreier et al.* [173]. As a step towards a solution, they introduced a *1-vs-Set Machine*, which sculpts a decision space from the marginal distances of a 1-class or binary SVM. Based on this principle, [174] propose multi-class classifiers that detect unknown instances by learning SVMs that assign probabilistic decision scores instead of class labels. More recently, *Bendale et al.* [13] addressed the more realistic case of a finite set of known objects mixed with many unknown objects. They adapted a DNN for open-set recognition by introducing a new model

layer, OpenMax, which computes the probability of a test samples being from an unknown class. Their key element of estimating the unknown probability is adapting Meta-Recognition [175] concepts to the activation patterns in the high-level feature space layer of the network. Generally, in the state-of-the-art, we observe two branches of open-set recognition methods that aim to solve the minimization problem given in the formula 2.3, traditional machine learning and DNN-based methods, which we discuss in the following sections.

2.3.2 Traditional ML-based open-set recognition methods

Following the method proposed by *Schreier et al.* [173], many SVM-based approaches were proposed to solve the open-set recognition problem. *Cevikal et al.* [24] added another constraint on the samples of target classes based on the SVM and proposed the Best Fitting Hyperplane Classifier (BFHC) model, which directly formed a slab in feature space. In addition, BFHC can be extended to the nonlinear case using specialized nonlinear kernels. Hence, as reported in [52], the slab models decrease the region of the known sample for each binary SVM, and the space occupied by each known class remains unbounded. Thus the open space risk still exists. To overcome this challenge and control this risk, *Scheirer et al.* [174] incorporated nonlinear kernels into a solution that further reduced open space risk by positively labeling sets with finite measures. They introduced a new formal model of probabilistic class association for open-set recognition called Compact Abating Probability (CAP). In a CAP model, the probability of class membership abates as points move from known data to open space, which accounts for the unknowns without the need to explicitly model them. They also introduced a novel technique called the Weibull-calibrated SVM (W-SVM), which combines CAP with the statistical extreme value theory (EVT) [85] for improved multi-class open-set recognition.

On the other hand, some other works focused on distance-based approaches to solve open-set recognition problems. *Scheirer et al.* [176] proposed a Nearest Non-Outlier (NNO) algorithm for open-set recognition by proposing an extention to the Nearest Class Mean (NCM) classifier [119]. NNO performs classification based on the distance between the testing sample and the mean of each known class, where it rejects an input sample when all classifiers reject it. Further, based on the traditional Nearest Neighbor classifier, *Junior et al.* [73] introduced an open-set version of the Nearest Neighbor classifier (OSNN) to deal with the open-set recognition problem. Their approach extends upon the traditional closed-set neural network to decide whether or not a test sample can be identified as unknown. Instead of using a fixed threshold on the similarity score of the most similar classes, their method uses the ratio of similarity to the two most similar classes by based on a threshold. One advantage of *Junior et al.* [73] approach compared to other existing methods for open-set

scenarios, is that it is inherently multi-class, i.e., the efficiency of the OSNN is not affected as the number of training classes increases.

The main issue with these traditional methods is that they have difficulty learning a highly discriminative feature space that allows distinguishing the unknown from known samples.

2.3.3 DNN-based open-set recognition methods

As discussed in section 2.2.1, DNNs have gained significant benefits for various tasks, especially in the image classification task. Yet, fooling/rubbish unknown images, which are, to human observers, clearly not from a class of interest, present a difficult challenge to DNN classifiers. So deep networks produce high confidence but incorrect classification. This meets the open-set recognition problem.

Bendale et al. [13] proposed the first solution toward open-set Deep Networks. They replaced the SoftMax layer in DNNs with an OpenMax layer. A key insight in their method is that they measured the *open space risk* in feature space rather than in pixel space. Indeed, a deep neural network is first trained with the standard SoftMax layer with known labeled samples by minimizing the cross-entropy loss. The pre-trained DNN is then used to map all the correctly classified training samples into the feature space (the last layer before the softmax layer). Based on these mapped samples, denoted by Activation Vectors (AV), each class is represented as a mean activation vector (MAV) . Next, based on the concept of NCM [119], the training sample distances from their corresponding class MAVs are computed and used to fit the separate Weibull distribution model for each class. Further, the activation vector's values are redistributed according to the Weibull distribution fitting score and then used to compute a pseudo-activation for unknown samples. Finally, the class probabilities of known and unknown samples are computed by using SoftMax again on these new redistributed activation vectors.

Using a generative adversarial network (GAN) to synthesize mixtures of unknown samples, *Ge et al.* [51] proposed the generative version of OpenMax denoted by (G-OpenMax), which can provide explicit probability estimation over the generated unknown samples, enabling the classifier to locate the decision margin according to the knowledge of both known and unknown generated samples. This is done by using synthetic samples as an extra training label apart from known labels. Such generated unknown samples are driven from the originally known data space. Yet, even if G-OpenMax effectively detects unknown samples in monochrome digit datasets, it has no significant performance improvement on natural images.

Oza et al. [135] adapted a class conditioned auto-encoders with specialized training and testing methodology. They separated the training procedure into two sub-tasks: closed-set

classification and open-set identification (i.e., identifying a class as known or unknown). The encoder learns the first task following the closed-set classification training pipeline, whereas the decoder learns the second task by reconstructing conditioned on class identity. Furthermore, as in [13], they modeled reconstruction errors using the EVT of statistical modeling to find the threshold for identifying known/unknown class samples. Similarly, *Ryota et al.* [237] proposed Classification-Reconstruction learning for open-set recognition. As shown in the figure 2.20, their training procedure provides label prediction for the Closed-set classification and a latent space for unknown samples identification. They further design deep hierarchical reconstruction nets (DHRNets) to provide effective probability prediction and discriminative low-dimensional latent space simultaneously. The critical idea in DHRNets is the bottlenecked lateral connections, which is helpful to learn rich representations for classification and compact representations for detection of unknowns jointly. DHRNets learn the reconstruction of each intermediate layer in classification networks using latent representations.

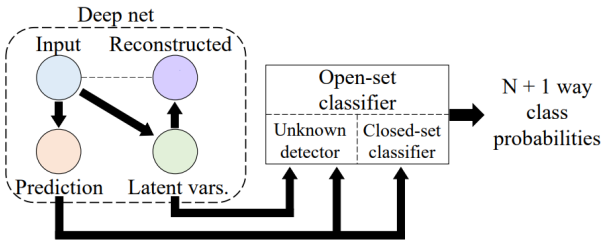

Fig. 2.20 *Ryota et al.* [237] proposed Classification-Reconstruction learning for open-set recognition.

Lei et al. [183] proposed a framework with four components to address the open-set recognition: (1) An Open Classification Network (OCN), which is used for open classification (traditional classification with rejection capability), that can produce rejected samples when tested on both seen and unseen classes. (2) A Pairwise Classification Network (PCN) classifies whether two input examples are from the same or different classes. (3) An auto-encoder that is used to learn representations from unlabeled examples. (4) A hierarchical clustering that clusters the rejected samples from OCN using PCN as the distance measure. It gives the number clusters or classes embedded in the rejected examples.

DNN-based open-set recognition methods showed promising improvements to open-set recognition. In contrast to traditional methods, DNNs allow learning a highly discriminative feature space, which facilitates the identification of unknown samples. Moreover, the learned feature space gives a low-dimensional representation, which allows using ML-based

traditional methods in a hybrid manner to enforce the identification of unknown samples. Our adopted method for open-set hand activity recognition is based on this hybrid concept following *Prakhya et al.* [149] method, which we detail in chapter 4, section 4.2.

2.3.4 Open-set activity recognition

In the literature, we find that most open-set recognition applications were devoted to image recognition tasks. While only a few existing literatures explored it for the activity recognition task.

Shu et al. [182] was the first to propose an open-set solution for activity recognition, assuming that applying closed-set recognition methods will lead to unseen-category errors. Since it is impossible to know all activity classes beforehand and consequently infeasible to prepare sufficient training samples for those emerging classes, they proposed a multi-class Triplet thresholding method combining the inter-class association for detecting unknown samples, which captures how different classes vary in the activation level (the high-level features vectors). Based on the initial training data, they calculated a Triplet threshold for each category, the acceptance, the rejection, and the distance rejection thresholds.

Bao et al. [10] proposed a deep evidential activity recognition method to recognize actions in an open-set context. Given a activity video as input, the evidential neural network head on top of an activity recognition backbone predicts the class-wise evidence, which formulates a Dirichlet distribution so that the input's multi-class probabilities and predictive uncertainty can be determined. High uncertainty videos can be rejected as unknown actions for the open-set inference, while the learned categorical probabilities classify low uncertainty videos. *Krishnan et al.* [87] utilized the stochastic variational inference technique while training a Bayesian DNN to infer the approximate posterior distribution around model parameters and perform Monte Carlo sampling on the posterior of model parameters to obtain the predictive distribution with uncertainty score. This uncertainty score allows unknown activity rejections. They showed that the Bayesian inference applied to DNNs provides reliable confidence measures for visual activity recognition task. Similarly, *Subedar et al.* [194] proposed an uncertainty-aware multimodal Bayesian fusion framework for open-set activity recognition. They focused on audiovisual activity recognition and used Bayesian DNN with stochastic variational inference to estimate the uncertainty associated with the individual modalities for multi-modal fusion. According to the resulted uncertainty, activities can be rejected as unknown.

To address the online activity recognition in an open-set context, *Wentao et al.* [11] proposed a multi-task general framework by decoupling the overall objective into three related tasks: uncertainty-aware action classification, actions prediction, and temporal location

regression. In essence, the foreground actions are distinguished from the background by the action prediction and localized by the temporal localization, while, as in [194, 87], the known and unknown foreground actions are discriminated by the learned evidential uncertainty from the classification module.

2.3.5 Conclusion

In section 2.3, we presented state-of-the-art methods related to the detection of unknown hand activities. First, we formulated the problem of open-set recognition. Then, we presented two categories of existing methods commonly used in the literature, traditional ML-based and DNN-based open-set recognition. Finally, we gave an overview of existing open-set activity recognition methods.

From the presented state-of-the-art study, we conclude that, generally, ML-based methods can be very efficient for identifying unknown samples when the provided data is quite discriminating to the point that it allows a minimum of distinction between the known and unknown samples. However, the central issue of these methods is that they struggle to learn a highly discriminative data representation. In contrast, thanks to sophisticated DNNs (e.g., CNN-based), DNN-based methods can learn highly discriminative data representation, which facilitates the identification of unknown samples. Nevertheless, standard DNN classifiers, such as the *Softmax*, can not effectively exploit this representation to identify unknown samples and perform open-set recognition. A hybrid combination of DNN-based and traditional ML-based methods can be envisaged as a tunning solution. Thus, the DNN-based methods can be used to learn a suitable data representation; and the traditional ML-based methods can exploit this representation and perform unknown sample detection and open-set recognition.

Following our objectives introduced in the previous chapter, section 1.2, the detected unknown activities are expected to be exploited to extend the recognition model capability. This is done by clustering and annotating these unknown activities and then integrating them into the detection and recognition models. In this regard, in the next section, we formally define the problem of clustering unknown (unlabeled) activities and present existing state-of-the-art related approaches.

2.4 Unlabeled hand activity clustering

The third component of our desired hand activity recognition framework introduced in the previous chapter, section 1.2 aims to annotate the detected unknown activities semi-automatically. Once the number of detected unknown activities reaches a fixed threshold, they will be semi-automatically annotated (labeled). First, we categorize (cluster) them based on similarities; then, a human expert assigns a class label for each category. By exploring the state-of-the-art, we found that this clustering procedure overlaps with the field of unsupervised clustering. Clustering consists in finding the underlying distribution of samples in their feature space. In other words, the clustering algorithm aims at forming homogeneous groups (or clusters) from an unlabeled dataset according to a specific notion of similarity. Observations considered similar are associated with the same group, while those considered different are associated with different groups.

Clustering high-dimensional unlabeled samples requires mapping them in a lower-dimension disseminative manifold. This requirement is addressed in Semi-Supervised Learning (SSL) methods [55, 253], which uses prior knowledge from a labeled dataset to exploit unlabeled samples. Generally, this is done by pre-training a model in a supervised manner on labeled data, then using its feature space as a mapping low-dimensional manifold to categorize unlabeled samples. The primary purpose of SSL is to boost the performance of an existent recognition model based on the assumption that both the labeled and unlabeled datasets come from the same domain. This means that the data distribution and the initial feature space remain the same. However, in real-world applications, the learning process is continuous. We may need to label samples belonging to new activity classes from a new related domain to extend the model recognition capability.

Unlike the SSL, in the Unsupervised Domain Adaptation (UDA), the target and source domain differ [137]. This means that the distribution, the initial features space, or both differ in the target and the source domains. Our clustering problem can approximate a UDA problem. Thus, the labeled (known) activity samples can be seen as the source domain; and the unlabeled (unknown) ones as the target domain. This allows the handling of unlabeled samples that belong to a new related target domain by leveraging the knowledge of the source domain. Better usage of prior knowledge from the source domain is essential for UDA. Therefore, the pre-trained model, which relies on the source domain, must map unlabeled samples to a highly discriminative feature space. This requirement is highly discussed in the field of open-set face recognition [107, 217, 35]. Thus, we concluded that the desired feature space must satisfy two main objectives of metric learning: (1) maximizing inter-class distances and (2) minimizing intra-class distances for the mapped samples.

The remainder of this section is organized as follows: subsection 2.4.1 formally defines the problem of unlabeled sample annotation to facilitate the introduction of state-of-the-art methods. In subsection 2.4.2, we present state-of-the-art UDA approaches for general application, particularly for human activity recognition. Subsection 2.4.3 extensively presents existing deep metric learning approaches, where we discuss their advantages and disadvantages since one of the major contributions of this thesis work falls in this research field. Subsections 2.4.4 and 2.4.5 are devoted to introducing six clustering algorithms we selected for their wide utilization for evaluating metric learning methods and consensus clustering techniques, respectively. In subsection 2.4.6, we introduce widely used external clustering evaluation metrics, and subsection 2.4.7 presents the evaluation datasets. Finally, subsection 2.4.8 concludes this section.

2.4.1 Unlabeled sample annotation problem formulation

Let $\mathscr{D}_s = \{\mathscr{X}, P_s(x^l)\}$ be the source domain that consist of two components: the initial features space $\mathscr{X}$ and the probability distribution $P_s(x^l)$ where $x^l \in \mathscr{X}$ is a particular labeled sample. Now, let $\mathscr{T}_s = \{\mathscr{Y}, f_\phi \circ g_w(.)\}$ be the source activity recognition task, which consists of two components: $\mathscr{Y}$ is the space of all the labels of samples in $\mathscr{X}$, and $f_\phi \circ g_w(.)$ is the recognition model, where f_ϕ is the features learner backbone, and $g_w(.)$ is the classification layer with a learnable parameters ϕ and w respectively. The subset of labels $\mathscr{Y}_s \subset \mathscr{Y}$ of the source task is known, so the labeled data of the source domain is formulated as a set of pairs $\{x^l, y^l\}$ where $y^l \in \mathscr{Y}_s$.

For a given set of unlabeled samples $\mathscr{X}^u \subset \mathscr{X}$, we try to exploit the knowledge from $\mathscr{D}_s$ and $\mathscr{T}_s$ to find the correspondent unknown subset of labels $\mathscr{Y}_u \subset \mathscr{Y}$. We formulate the target domain as $\mathscr{D}_u = \{\mathscr{X}, P_u(x^u)\}$ where $x^u \in \mathscr{X}^u$. In our UDA application, we assume that the source and the target domain differ $\mathscr{D}_s \neq \mathscr{D}_u$ since only the distributions differ $P_s \neq P_u$. While the initial features space $\mathscr{X}$ and the recognition source/target tasks remain the same [137].

2.4.2 Unsupervised Domain Adaptation (UDA)

UDA methods are based on the assumption that there are no available labels in the target domain. The concept is that only the label information from the source domain can be exploited to disseminate shared knowledge across domains and improve model transfer capability. In recent years, this concept of UDA has attracted increasing attention from the computer vision community, especially in the field of image classification [209], object detection [168], and semantic segmentation [172]. A reference work was proposed by [136], which aims at aligning the source and the target domains by projecting data onto a set of

learned transfer components. Recently, the success of adversarial learning has influenced UDA's proposed methods. They mainly aim at minimizing the domain discrepancy between the source and the target domain while learning a domain discriminator by maximizing its loss concerning the discriminant features learner [208].

Generally, most of the introduced UDA methods try to find the best mapping manifold $\mathscr{Z}$ that discriminantly represents unlabeled target domain samples $\mathscr{X}^u$. To this end, several metric learning-based UDA methods have been proposed [147]. *Laradji et al.* [91], proposed a metric-based adversarial discriminative domain adaptation for image tasks. First, they used a metric learning approach to train a supervised model on the source dataset by optimizing the Triplet loss function [110], which results in a pre-trained model that provides a discriminative embedding space. Next, they used the adversarial approach (same as in [208]) to make the extracted features from the source and target datasets indistinguishable.

In this thesis work, we do not need to align the target with the source domain in our UDA application. We assume that the learning process is continuous, and we may discover new class categories in the target domain which do not match those in the source domain.

Despite the progress in UDA for third-person activity recognition [98, 154], there have been few works for the first-person hand activity recognition, which are based on RGB images modality [14]. Most of these introduced methods share the same goal of learning discriminative features space by focusing on low-level spatial-temporal feature learning. However, to enforce the discriminative power of the neural network, which is crucial for UDA, we must also consider the high-level feature space.

2.4.3 Deep Metric Learning Losses (DML)

Metric learning techniques [72, 223, 207] aim at learning semantic distance measures and embeddings that help map samples into a highly discriminative features space $\mathscr{Z}$ that facilitate these tasks. The desired feature space is expected to satisfy two main objectives of metric learning: (1) maximizing inter-class distances and (2) minimizing intra-class distances.

With the development of DL, deep metric learning methods showed many improvements in recent years by employing neural networks as a low-dimensional mapping function, which is nothing but $f_\phi(.)$. The employed neural networks are optimized with specialized loss functions to satisfy metric learning objectives on the high-level features space. In this regard, we found two main categories of loss functions in the state-of-the-art, Contrastive and penalty-based Softmax losses.

Contrastive approaches. The idea of contrastive approaches is to design a loss function that directly applies the two metric learning objectives, hence the name "Contrastive." These

methods are regarded as "Direct" because they directly apply the metric learning definition. The distance measure in the embedding space for this type of approaches is fixed as the l_2 Euclidean distance. For $x_1^l, x_2^l \in \mathscr{X}$ two labeled samples, the distance is formulated as follows:

$$\rho_{f_\phi}(x_i^l, x_2^l) = \left\| z_1^l - z_2^l \right\|_2 \tag{2.4}$$

where $f_\phi(x_1^l) = z_1$ and $f_\phi(x_2^l) = z_2$ with $z_1^l, z_2^l \in \mathscr{Z}$. We denote $\rho_{f_\phi}(x_i^l, x_2^l)$ as a shortcut for $\rho(f_\phi(x_1^l), f_\phi(x_3^l))$.

Chopra et al. [32] are the first to propose a reference constrastive loss to solve the face identification problem. For $x_1^l, x_2^l \in \mathscr{X}$ two labeled samples, where $y_1^l, y_2^l \in \mathscr{Y}$ are the corresponding labels. let us denote $\mathbb{I}_A$ as the identity function that is equal to 1 if A is true, and 0 otherwise. The constrastive loss is then defined as follows:

$$\mathscr{L}_{contrastive} = \mathbb{I}_{y_1=y_2}\rho_{f_\phi}^2(x_1^l, x_2^l) + \mathbb{I}_{y_1\neq y_2} max(0, m - \rho_{f_\phi}^2(x_1^l, x_2^l)) \tag{2.5}$$

where m is the margin parameter that avoids the network f_ϕ to map all samples to the same point, making distances between samples equal to zero.

Another popular contrastive loss function was proposed by Schroff et al, [177], denoted by *Triplet loss*. It enforces the distance of a negative pair to be larger than that of a positive pair by imposing a given margin manually. More formally, let $x_1^l, x_2^l, x_3^l \in \mathscr{X}$ and their corresponding labels $y_1^l, y_2^l, y_3^l \in \mathscr{Y}$ with $y_1 = y_2$ and $y_1 \neq y_3$. Typically, in this case, x_1^l is called *anchor* sample, x_2^l is called *positive* sample because it has the same label as x_1^l, and x_3^l is called *negative* sample because it has a different label. The *Triplet loss* is defended as follows:

$$\mathscr{L}_{Triplet} = max\left(0, m - \rho_{f_\phi}^2(x_1^l, x_3^l) + \rho_{f_\phi}^2(x_1^l, x_2^l)\right) \tag{2.6}$$

The fundamental problem with the *Triplet loss*, as noted in [187], is that it only compares a sample with one negative sample while disregarding negative data from the other classes. As a result, we may end up distinguishing an example from only a small number of negative classes while keeping a small distance from many other classes.

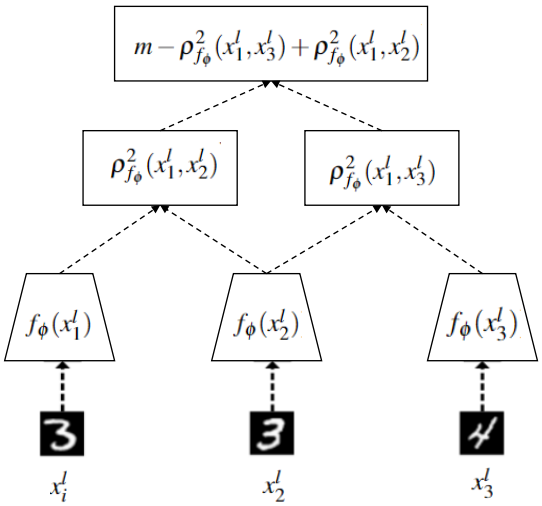

Fig. 2.21 Illustration of the *Triplet* network using samples from the MNIST dataset [36].

To tackle the *Triplet loss* limitations, *Chen et al.* [28] proposed a *Quadruplet loss* that requires training the network with four inputs instead of three. The *Quadruplet loss* aims at enhancing inter-class variation and minimizing the intra-class variation, unlike the *Triplet Loss* that does not focus on the class variation in the feature space. For samples $x^l_1,x^l_2,x^l_3,x^l_4 \in \mathscr{X}$ and their corresponding labels $y^l_1,y^l_2,y^l_3,y^l_4 \in \mathscr{Y}$ with $y_1 = y_2 = y_4$ and $y_1 \neq y_3$ the *Quadruplet loss* can be defined as:

$$\begin{aligned}\mathscr{L}_{Quadruplet} &= max\left(0, m_1 - \rho^2_{f_\phi}(x^l_1,x^l_2) + \rho^2_{f_\phi}(x^l_1,x^l_4)\right) \\ &+ max\left(0, m_2 - \rho^2_{f_\phi}(x^l_1,x^l_4) + \rho^2_{f_\phi}(x^l_1,x^l_3)\right)\end{aligned} \tag{2.7}$$

The authors in [127] compared the *Triplet* and *Quadruplet* loss functions in a healthcare application. They obtained a better degree of compactness between patients of the same class while using *quadruple loss* samples in each training batch compared to *Triplet loss*.

There have been other attempts to design a better metric learning objective based on the concept of the *Triplet Loss*. *Wang et al.* [218] suggested focusing on angular constraint at the negative point of Triplet triangles. Their *angular loss* pushes the negative point away from the center of the positive cluster and then brings positive points closer to each other while using an angle that is a rotation and scale-invariant metric [78]. In the same scope, *Song et al.* [189] proposed *Structured Loss*, which improves the sampling effectiveness of *Triplet Loss* and makes full use of the samples in each batch of training data. Similarly, N-pair loss [187] extends the Triplet loss by separating more than one negative sample from the anchor compared with the positive sample. Multi-similarity loss [220] considers both self-similarity

and relative similarity for weighting informative pairs in an iterative manner. Other loss functions includes lifted structured loss [129], *Clustering loss* [188], *Hierarchical Triplet loss* [50], *Ranked list loss* [221], and *Tuplet margin loss* [238], and *Magnet Loss* [157].

Many research works focused on improving the basic idea of *Triplet loss* in supervised deep metric learning. Yet, it has become clear that learning to directly minimize/maximize the Euclidean distance between samples with identical/different labels may not be the right solution. There are three main problems with these approaches:

- **Expansion problem.**

  We have no guarantee that samples with the same label be mapped closely to a common region in space, as reported by [187]. On the other hand, *Quadruplet Loss* [28] only improves the variability and enhances the inter-class distances. *Structured Loss* can impose the structure just locally for the samples in the batch, not globally for all samples. Attempts to handle this problem directly with a global aim, *Magnet Loss* [157], and *Clustering Loss* [188], were mostly unsuccessful owing to high complexity and scalability concerns.

- **Sampling problem.** All the deep metric learning approaches that try to minimize/maximize the distance between samples require sophisticated sample mining procedures that select the "most useful" samples for learning for each training batch. These mining procedures are time-consuming, performance-sensitive, and can become problematic in a distributed training setting.

- **Beyond metric learning applications.** All these methods can not be used simultaneously for another task (e,g., a classification task), while it is very suitable to exploit trained networks to perform classification tasks.

Penalty-based Softmax approaches. A series of deep metric learning methods were proposed to solve the issues mentioned earlier. Most of these methods are inspired by the well-known classification *Softmax loss* function. Generally, they exploit the weight vectors of the last layer (see figure 2.22) to force the network to learn a discriminative feature space that satisfies metric learning objectives.

Before getting into the details of these methods, let us formally describe the *Softmax loss* function that we denote by $\mathscr{L}_{Softmax}$, and we formulate it in combination with the cross-entropy loss as follows:

$$\mathscr{L}_{Softmax} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{w_{y_i}z_i+b_{y_i}}}{\sum_{j=1}^{C} e^{w_j z_i+b_j}} \tag{2.8}$$

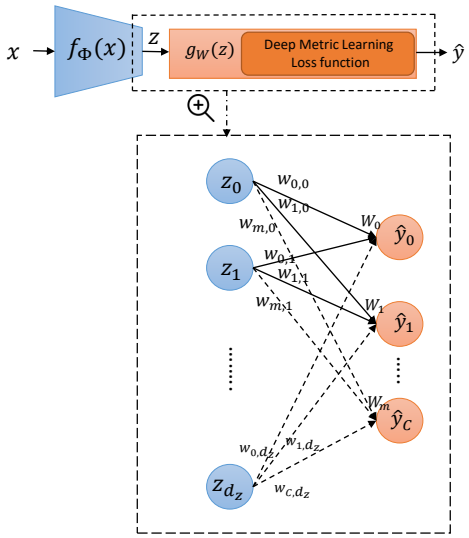

Fig. 2.22 Illustration of the fully connected classification layer that relies on deep metric learning loss functions.

Where N is the training batch size, C is the number of labeled classes. $z_i \in \mathscr{Z}$ is the embedding vector of the i-th sample x_i where $f_\phi(x_i) = z_i$. w_j is the j-th weight of the last FC layer and $b_j \in \mathbb{R}^C$ is the bias. In other words, the inner product-based logit, which is formulated as $w_{y_i} z_i + b_{y_i}$, is passed through a softmax function that normalizes and results in a soft probabilistic affinity score. The *Softmax loss* separates embedding vectors from different classes by maximizing the posterior probability of the positive ground-truth class. However, it does not explicitly imply any constraints on the learning procedure to satisfy the metric learning objectives.

In response to *Softmax loss* limitations,*Wen et al.* [224] proposed *Center Loss*. Their main idea is to add a new regularization term to the Softmax Loss to pull the features to corresponding class centers:

$$\mathscr{L}_{Center} = \mathscr{L}_{Softmax} + \frac{\lambda}{2} \sum_{i=1}^{N} \left\| z_i - c_{y_i} \right\|_2^2 \tag{2.9}$$

Where c_{y_i} denotes the y_i class center of deep features. A scalar λ is used for balancing the two loss functions. The conventional *Softmax loss* can be considered a special case of this joint supervision if λ is set to 0. During the training procedure, the class center c_{y_i} is also updated using gradient descent and can be thought of as the moving mean vector of the set of feature vectors of class y_i. As a result, this will minimize the intra-class variations while keeping the features of different classes separable. However, there is no guarantee that this

will result in a good inter-class separation since the clusters closer to zero will benefit less from the regularization term.

To address the *Center loss* limitation, *Liu et al.* [107] proposed the *SphereFace* loss function for face recognition. In contrast to *Center loss*, they enforce the class centers to be at the same distance from a center by mapping them to a hypersphere. The authors also proposed a modified version of Softmax. First, they fixed $b_i = 0$ and normalized embedding vectors $\|z_i\| = 1$ and the weights $\|w_j\| = 1$. Then, the logit is modified as $\|z_i\| cos(\theta_{y_i})$ where $\theta_{y_i} = arcos(\frac{w_{y_i} z_i}{\|w_{y_i}\| \|z_i\|})$ is the angle between the embedding z_i and the weight vectors w_j thus $(0 \leq \theta_{j,i} \leq \pi)$. The modified Softmax objective is defined as follows:

$$\mathscr{L}_{MSoftmax} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{\|z_i\| cos(\theta_{y_i,i})}}{\sum_{j=1}^{C} e^{\|z_i\| cos(\theta_{i,j})}} \tag{2.10}$$

This means that the sample x_i is assigned to the class j if the angle between w_j and its embedding vector z_i is the smallest among all the class centers $\{w_{j=1:C}\}$. In this case, the decision boundary for the modified Softmax is thin, and it will not make the features discriminative enough since, for very similar classes, the inter-class distance will be too small. To address this issue, the authors proposed a multiplicative margin m to penalize the target logit, enhancing inter-class separation and intra-class closeness. Thus the *SphereFace* objective can then be expressed as follows:

$$\mathscr{L}_{SphereFace} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{\|z_i\| cos(m\theta_{y_i,i})}}{e^{\|z_i\| cos(m\theta_{y_i,i})} + \sum_{j \neq y_i}^{C} e^{\|z_i\| cos(\theta_{i,j})}} \tag{2.11}$$

with the requirement that $\theta_i \in \left[0, \frac{\pi}{m}\right]$. For e.g., for two classes C_1 and C_2, the decision boundary is given by :

$$\begin{aligned} C_1 &: cos(m\theta_1) \geq cos(m\theta_2) \\ C_2 &: cos(m\theta_2) \geq cos(m\theta_1) \end{aligned} \tag{2.12}$$

In order to make the optimization possible via neural network, the authors expand the definition range of $cos(\theta_{y_i,i})$ by generalizing it to a monotonically decreasing angle function $\psi(\theta_{y_i,i})$, which can be defined as $\psi(\theta_{y_i,i}) = (-1)^k cos(m\theta_{y_i,i}) - 2k$ with $\theta_{y_i,i} \in [k\pi/m, (k+1)\pi/m]$ and $k \in [0, m-1]$. The final *SphereFace* loss can be expressed as follows:

$$\mathscr{L}_{SphereFace} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{\|z_i\| \psi(\theta_{y_i,i})}}{e^{\|z_i\| \psi(\theta_{y_i,i})} + \sum_{j \neq y_i}^{C} e^{\|z_i\| cos(\theta_{i,j})}} \tag{2.13}$$

During the training procedure, the optimization using the *SphereFace* loss will force the network to map embedded samples z_i closer to their corresponding w_j, which makes the feature space more discriminative. However, the posterior Softmax probability will merely rely on the cosine of $m\theta_{y_i,i}$. As a result, different margins apply to different classes. As a result, in the decision space, certain inter-class traits have a more significant margin of error than others, reducing discriminating capability. Nevertheless, the success of *SphereFace* has led to a series of new methods based on the same idea of using the angular distance with the angular margin.

Inspired by the *SphereFace* idea, *Wang et al.* [217] proposed the *CosFace* loss function to overcome the *SphereFace* limitation. Figure 2.23 illustrates the geometrical interpretation of the *CosFace* loss. The authors defined the margin in the cosine space rather than the angle space to penalize the *Softmax* logit by modifying it as $s\,(cos(\theta_{y_i,i}) - m)$, where s is the scaling parameter. Thus the *CosFace* objective is then defined as:

$$\mathscr{L}_{CosFace} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\,(cos(\theta_{y_i,i})-m)}}{e^{s\,(cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i}^{C} e^{s\,cos(\theta_{i,j})}} \tag{2.14}$$

The choice of scale s and the margin m values is crucial to obtain the expected performance.

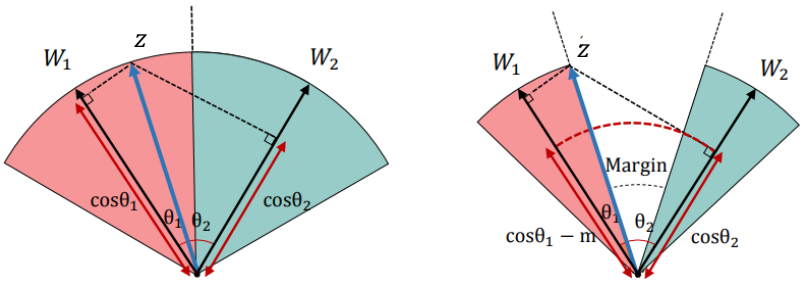

Fig. 2.23 A geometrical interpretation of *CosFace* (Right figure) from feature perspective. Different color areas represent feature space from distinct classes. *CosFace* has a relatively compact feature region compared with *Modfied Softmax* in left figure.

The *CosFace* authors proposed a lower bound for the scale parameter. Let P_W denote the expected minimum posterior probability of class center (i.e. W). The lower bound of s is given by:

$$s \geqslant \frac{C-1}{C} log\frac{(C-1)P_W}{1-P_W} \tag{2.15}$$

They also proposed a variable scope of margin m based on two assumptions: (1) In order to conduct perfect classification, the softmax loss aims to maximize the angle between any two weight vectors from two different classes; (2) The best solution for the softmax loss should evenly distribute the weight vectors over a unit hypersphere. The variable scope is defended as follows:

$$\begin{array}{l} 0 \leqslant m \leq 1 - cos\frac{2\pi}{C}, \quad (k=2) \\ 0 \leqslant m \leq \frac{C}{C-1}, \quad (C \leqslant d_z + 1) \\ 0 \leqslant m \ll \frac{C}{C-1}, \quad (C > d_z + 1) \end{array} \tag{2.16}$$

Following the same path, *Deng et al.* [35] proposed the *ArcFace*, a very simular loss function to *CosFace*. However, instead of defining the margin in the cosine space, it defines the margin directly in the angle space as follows:

$$\mathscr{L}_{ArcFace} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y_i,i}+m))}}{e^{s(cos(\theta_{y_i,i})+m)} + \sum_{j\neq y_i}^{C} e^{s\,cos(\theta_{i,j})}} \tag{2.17}$$

Angular distance optimization has become a natural approach employed by most of the loss functions mentioned above. Indeed, during the training procedure, the weight vectors of the classification layer that rely on the classification loss function are used as centers of attraction to the embedded feature vectors regarding the ground-truth supervision. This kind of optimization encourages the network to enhance the intra-class compactness in the feature space. However, it does not explicitly force the neural network to maximize inter-class separation.

2.4.4 Clustering algorithms

In the previous subsection, we introduced a deep metric learning loss function that allows learning a mapping function f_θ that maps unlabeled samples $\mathscr{X}^u$ of the target domain into a clustering-friendly manifold $\mathscr{Z}$. This subsection introduces state-of-the-art clustering algorithms usually used to cluster these mapped samples. In this thesis work, clustering algorithms are mainly seen and used as tools to evaluate the quality of the learned features space $\mathscr{Z}$. There exist several clustering approaches. For a comprehensive study of existing clustering algorithms, we refer the reader to [232]. Each approach is best suited to particular data distribution. In the following, we introduce six selected widely used clustering algorithms to validate metric learning methods' performance:

K-means clustering algorithm. *K-means* is the most widely used clustering algorithm for its simplicity and low computational cost. It is a centroid-based algorithm [232]. The variable *K* represents the number of groups or classes the algorithm seeks to create. First, the *K* points are placed in the data space representing the group of initial centroids. Then, each data point is assigned to the nearest *K*. Once all points are assigned to their corresponding centroids, the positions of the *K* centroids are recalculated. These last two steps are repeated until the positions of the centroids do not move.

With the increasing size of datasets, the computational time of *K-means* increases due to its constraint of needing to load dataset in the main memory. For this reason, several *K-means* variants have been proposed to reduce the temporal and spatial cost, such as *Mini Batch K-means* [180] or *Nested Mini-Batch K-Means* [124].

The disadvantage of the *K-mean* clustering algorithm is that it deals very poorly with high-dimensional data. When the data dimension increases, the distance-based similarity measure converges to a constant.

DBSCAN clustering algorithm. (Density-based spatial clustering of applications with noise) is a data clustering algorithm proposed by *Ester et al.* [42]. It is a density-based algorithm [232] that relies on the estimated density of clusters to perform the clustering. *DBSCAN* searches, for each point, the points that are part of its immediate neighborhood based on a initialy given radius value, and it groups all the points that can be reached. Once all the points are grouped, the groups of points are considered clusters if the number of points in the group is superior or equal to a initialy given minimum number of points. Otherwise, the points are considered outliers.

One of the advantages of *DBSCAN* is that it is not necessary to know the number of desired clusters in advance. Moreover, the algorithm detects and isolates outliers by itself. The concept of density does with all data distributions. The density may vary in the data space and may not be the same at all data points. The algorithm may perfoms poorly when there are no "holes" between the clusters, which makes clusters hardly separable.

OPTICS clustering algorithm. (Ordering Points to Identify the Clustering Structure) is a density-based clustering algorithm proposed by *Ankers et al.* [4]. It addresses *DBSCAN* limitation of detecting meaningful clusters in data of varying density, while it shares the same limitations and advantages.

BIRCH clustering algorithm. (Balanced Iterative Reducing and Clustering using Hierarchies). It is a hierarchy-based clustering algorithm [232] proposed by [245] to process large datasets. It splits the data into small summaries grouped in place of the original data points. The summaries contain as much information as possible about the original data points,

which makes it commonly used with other clustering algorithms to cluster its generated summaries.

BIRCH only requires a single dataset scan, making working with large datasets fast. It can incrementally cluster incoming, multi-dimensional data points in respect of a given set of computational resource constraints. Another advantage is that it does not require knowing the number of clusters in advance.

Affinity Propagation clustering algorithm. It is a message-passing based clustering algorithm proposed by *Frey et al*. [47]. This algorithm is entirely different from the others in how it clusters data. Each data point corresponds with all other data points by passing message information. This communication between data points allows each point to know how similar they are, which starts to form the clusters in the data.

The advantages of *Affinity Propagation* clustering is that it is insensitive to the outliers, and the number of clusters does not need to be known. However, it has a high computational complexity, which makes it unsuitable for a large dataset.

Agglomerative Hierarchical Clustering. *Agglomerative clustering* is generally used as a bottom-up clustering process. Initially, every data point forms its cluster. Then, in each iteration, the two 'closest' clusters will be merged until only one cluster remains. The objective is to construct a cluster hierarchy such that given any two distinct clusters, A and B, from conceivably different levels of the hierarchy, we obtain either $A \cap B = \varnothing$, $A \sqsubset B$, or $B \sqsubset A$ [191]. Such a hierarchy is useful in many application areas. In particular, for applications that are interested in the hereditary qualities of clusters, as in some bioinformatics applications. It can also be very helpful when the precise number of clusters is a priori unknown.

2.4.5 Consensus clustering

Consensus clustering aggregates the results of multiple clustering algorithms or the same algorithm with multiple runs with varying parameters. It is also called cluster ensemble [192] or aggregation of clusters and refers to a situation where several different predicted clusters have been obtained for a given data set, and a single consensus clustering is to be found that in some sense fits better than the existing clustering results. Figure 2.24 illustrates the clustering idea.

The most referenced work in this field was proposed by *Strehl et al.* [192]. Based on graph representation, the authors propose three consensus clustering algorithms: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-Clustering Algorithm (MCLA).

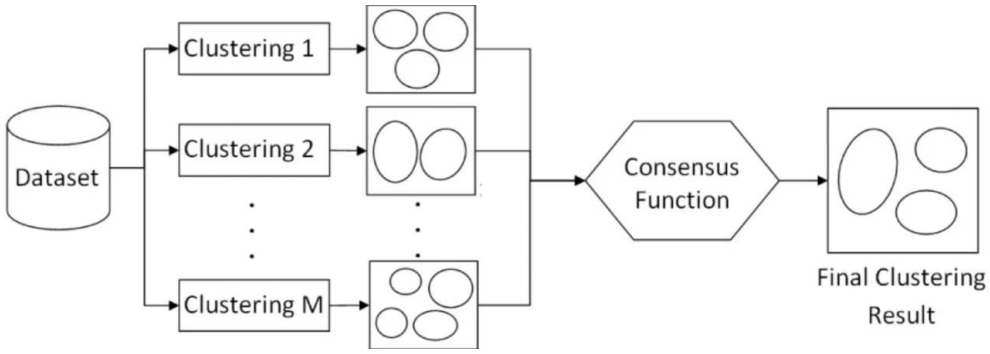

Fig. 2.24 Illustration of a consensus clustering for M clustering results, which the consensus takes as an input to generate the final clustering result.

- **CSPA.** A clustering implies a relationship between samples in the same cluster and can be used to create a pairwise similarity measure. Then, samples are re-clustered using this similarity measure, resulting in a combined clustering. First, a similarity matrix is computed. Then, an induced similarity graph is constructed. Vertices are represented by data points, and edges by similarity measures, which are partitioned into K clusters using METIS [76].
- **HGPA.** The hypergraph is constructed using the ensemble of clustering results, where the vertices correspond to the data points. Each hyperedge is a cluster from one of the clustering results in the ensemble. All hyperedges have the same weight. This algorithm searches for a hyperedge separator that partitions the hypergraph into K disconnected components of approximately equal size. The hypergraph partitioning package HMETIS [75] is used for this partitioning.
- **MCLA.** This algorithm is based on clustering the clustering results in the ensemble. Each cluster is represented as a hyperedge. The algorithm packs and collapses related hyperedges into the desired K number of clusters and then assigns each data point to the cluster (collapsed hyperedge) in which it is most involved.

In the context of our work, we used the *Strehl et al.* [193] proposed method because of its efficiency and the availability of its source code. We refer the reader to [25, 125] for more details on consensus clustering methods.

2.4.6 Clustering evaluation metrics

The evaluation of classification methods (supervised or unsupervised) requires independent and reliable measurements. It turns out that there is always a gap between theory and practice.

The data size, the details of its representation, and classification algorithms make intuitive assessments impossible. There is no absolute measure for the evaluation of clustering methods, but a variety of methods depend on the characteristics of the data or the algorithms. These methods can be divided into two families, internal and external evaluation methods. Internal methods are used to measure the goodness of a clustering structure without external information (ground-truth) [206]. External methods measure the agreement between two partitions where the first partition is the a priori known clustering structure (ground-truth), and the second results from the clustering procedure [41]

In this thesis work, we used only external methods since we have the ground-truth information for all the evaluated datasets. In the following, we present the most frequently used external evaluation methods:

Adjusted Rand Index (ARI). The Rand Index *(RI)* is a generic clustering evaluation metric proposed by *Randet al.* [235]. It is based on the computation of *TP*, *TN*, *FP*, and *FN*, where :

- *TP*: number of true positives (objects correctly grouped and belonging to the same group).
- *TN*: number of true negatives (objects not belonging to the same group and annotated as not belonging to the same group).
- *FP*: number of false positives (objects grouped together but not belonging to the same group in the ground truth).
- *FN*: number of false negatives (objects not grouped but belonging to the same group in the ground truth).

Thus the *RI* evaluation metric is formulated as follows:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.18}$$

It is a performance measure corresponding to the proportion of sample pairs for which the ground-truth labels and the predicted clustering labels meet. The advantage of *RI* is that it considers the number of *TN*, and thus a system that groups all samples s into a single cluster has a low *RI*.

In [68], the authors propose to perform an adjustment of the Rand Index: the Adjusted Rand Index *(ARI)*. This adjustment uses a calculation of μ, representing the Rand Index's mathematical expectation that a random grouping of samples can reach compared to the ground truth. The *ARI* is formulated as follows:

$$ARI = \frac{RI - \mu}{1 - \mu} \tag{2.19}$$

Normalised Mutual Information (NMI). Mutual information is another evaluation measure based on information theory that measures the entropy between the ground-truth labels and the clustering algorithm predicted labels. Its normalized version [192] is defined as follows:

$$NMI(G,M) = 2\frac{I(G,M)}{H(G)+H(M)} \tag{2.20}$$

where I is the mutual information, and H ik the entropy.

V-Measure. It is an entropy-based clustering evaluation method proposed by *Rosenberg et al*. [160]. It explicitly measures how successfully satisfied the criteria of homogeneity and completeness are. *V-measure* is computed as the harmonic mean of distinct *Homogeneity* and *Completeness* score.

- ***Homogeneity***: To satisfy the homogeneity criteria, a clustering must assign only those data points that belong to a single class to a cluster. Within each cluster, the class distribution should be skewed to a single class, resulting in zero entropy. Within each cluster, the class distribution should be shifted to a single class, resulting in zero entropy [160].

- ***Completeness*** : Completeness is symmetrical to homogeneity. To satisfy the completeness criteria, a clustering must assign all of those data points that are members of a single class to a cluster. To evaluate the completeness, we examine the distribution of cluster assignments within each class. Each of these distributions will be completely asymmetric with respect to a single group in a perfectly complete clustering solution [160].

V-measure can be weighted to favor the contributions of homogeneity or completeness.

Clustering Accuracy (ACC). For clustering algorithms that require the ground-truth number of classes k (e.g., *K-means*), the *ACC* [234] can be used to validate the clustering quality. Let us denote by $\hat{y}_i$ as the clustering result from the clustering algorithm and y_i the ground truth label. The *ACC* can be defined as follows:

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(\hat{y}))}{n} \tag{2.21}$$

where n is the number of samples to be clustered. $\delta(a,b) = 1$ if $a = b$ and $\delta(a,b) = 0$ otherwise. $map(.)$ is the optimal mapping function that permutes clustering labels to match

the ground truth labels. The best mapping can be obtained by using the Kuhn-Munkres algorithm [139].

2.4.7 Deep Metric learning-based clustering datasets

In the context of this thesis work, we have applied metric learning and clustering techniques for hand activity recognition using the previously introduced FPHA and SLS France datasets. Nevertheless, in the literature, we find other commonly used datasets to validate the methods of metric learning methods that we introduce in the following:

CUB200-2011 [214]. Contains 200 classes of birds with 11,788 images. It is usually divided into the first 100 classes for training (5,864 images) and the remaining classes for testing (5,924 images). In the training set, the maximum, minimum, mean and standard deviation of the number of images in each class are 60, 41, 58.6 and 3.5, respectively.

Cars196 [86]. Contains 196 classes of cars with 16,185 images. It is usually divided into the first 98 classes for training (8,054 images) and the remaining classes for testing (8,131 images). In the training set, the maximum, minimum, mean and standard deviation of the number of images in each class are 97, 59, 82.2 and 7.2, respectively.

2.4.8 Conclusion

In this section, we have formulated the problem of clustering unlabeled samples. Then, we presented the state-of-the-art UDA approaches and explained the tendency of researchers to use metric learning loss functions to solve UDA problems. In subsection 2.4.3, we have presented existing metric learning approaches in detail and discussed their advantages and disadvantages. We have also presented a selection of clustering algorithms and evaluation metrics. Our selection of the algorithms is based on their extensive use in evaluating metric learning methods. Finally, we presented the commonly used datasets for evaluating metric learning methods.

From this state-of-the-art study, we retained that approximating the clustering of unlabeled hand activities to an UDA problem can be a good solution. Thus, the unlabeled classes we seek to identify can be considered as a target domain, and the labeled classes as the source domain. Suitable exploitation of the prior knowledge from the source domain is crucial for UDA. Therefore, the pre-trained model, which relies on the source domain, must map unlabeled samples into a highly discriminative feature space. We have also concluded that the desired feature space must satisfy two main objectives of metric learning: (1) maximizing inter-class distances and (2) minimizing intra-class distances for the mapped hand activities. A lot of deep metric learning methods have been proposed. They are used to

optimize neural networks, learning a mapping function that maps unlabeled samples into a highly discriminative low-dimensional feature space. The resulting feature space facilitates measuring similarities between pairs of samples in such a discriminative manifold. The most these existing methods usually try to boost the discriminatory power of neural networks by enhancing intra-class compactness in the high-level features space. However, they do not explicitly impose constraints to improve inter-class separation. Based on these observations in chapter 4, section 4.3, we propose a new composite deep metric learning loss function that, in addition to the intra-class compactness, explicitly implies regulations to enforce the best inter-class separation.

In the next section, we present state-of-the-art methods related to incrementally integrating these clustered and annotated unlabeled samples, which is the goal of the final component of our desired user activity recognition framework.

2.5 Incremental Learning for hand activity recognition

In the previous section, we presented state-of-the-art methods related to the semi-automatic annotation of detected unknown activities. Following our objectives introduced in the previous chapter, section 1.2, the final component of our desired user activity recognition framework aims to integrate these annotated activities into the recognition and the unknown activity detection models. In other words, thanks to this final component, the framework incrementally learns and extends its multi-class classifier, making each new class "known" to the models. Observing the state-of-the-art related approaches, we found that this integration (update) procedure meets the incremental learning paradigm. More precisely, it meets class-incremental learning, where the system learns progressively newly arrived classes.

In incremental learning, the dataset is not necessarily entirely available at the beginning of the training. The system receives sets of training samples progressively and must be able to learn from each of these sets of samples separately. The system must therefore be able to modify itself and adjust its parameters after observing each set of classes to learn from it, but without forgetting the knowledge acquired from the previously learned classes. This learning procedure is used either when the dataset is too large with many classes to be used at once; or when the entire training set is unavailable, and the training data arrives incrementally.

Many methods have been adapted to the problem of supervised classification to address incremental learning challenges. They include but are not limited to SVM-based methods [251], neural networks [140], k-nearest neighbor [230], and decision trees [33]. The choice of an algorithm depends strongly on the task to be solved and the desired model interpretability. Following the thesis objectives introduced in the previous chapter, section 1.2, we focus on neural networks based class-incremental learning approaches. Moreover, these methods are the most commonly used to solve the incremental learning.

The main challenge in class-incremental learning is learning from data of the current set of classes in a way that prevents forgetting previously learned classes. The drastic drop in performance in previously learned classes is a phenomenon known as *catastrophic forgetting* [53]. Incremental learning aims to prevent *catastrophic forgetting* while at the same time avoiding the problem of intransigence, which inhibits adaptation to new tasks [26]. There are several causes of *catastrophic forgetting*, in the following, we present the most common ones:

Weight drift: When a network has learned to recognize an initial set of patterns, it has found a point in weight-space, $W_{initial}$, for which the network can recognize all the patterns it has seen. If the same network learns a new set of patterns, even if the new set is small, it will move to a new solution point in weight-space, W_{new}, corresponding to a set of weights that

allows the network to recognize the new patterns. As a result, performance on previous tasks often suffers dramatically.

Activation drift: Related to weight drift, activation drift changes lead to changes in neuron activation and, therefore, in the network output. Focusing on activation rather than weights can be less restrictive because it allows weights to change as long as they result in minimal changes in layer activation [118].

Inter-class confusion: The incremental learning objective is to distinguish all classes from all iterations. However, since classes are not jointly trained, the network weights cannot optimally discriminate all classes. As a result, the new classes will be highly confused with the old ones.

Task-recency bias: Separately learned tasks may produce incomparable classifier network results. Thus, the network is biased towards classes of newer learning iterations. This effect is observed in confusion matrices which illustrate the tendency to miss-classify inputs from the most recently seen classes, as reported by [118].

In the following section we introduce neural network based class incremental learning methods.

2.5.1 Fine-tuning based approaches

They are the most popular type of approach in class incremental learning.. Fine-tuning consists of initializing the model weights with those of the previous model to benefit from previously acquired knowledge and quickly start learning convergence on new data. Following [89], these methods can be categorized into three main categories: (1) replay-free methods approaches that do not use the memory of the past, (2) replay-based methods approaches that use limited memory of the past, and (3) pseudo-replay-based approaches that generate past images instead of storing them in memory.

Replay-free methods. This family of methods aims to prevent the weight *drift of weights* important to the previously learned classes. It also aims at preventing *activation drift* problems present in the classification layer. These methods do not re-use past model memory (initial training samples). They are often based on regularization losses.

A series of weight regularization methods have been proposed to tackle the *drift of weights* problem. Generally, these proposed methods compute the importance of each parameter in the network after each incremental learning iteration. When learning new classes, the more important the weights are, the more penalized the network is for these changes. Different techniques were proposed to measure the importance of weights that define the network changes. *Lee et al.* [93] used batch normalization layers and proposed a quadratic penalty method with a Hessian approximation. In [83], the authors proposed an elastic weights

consolidation technique, representing the weight importance as a diagonal approximation of the Fisher Information Matrix.

On the other hand, to solve the *activation drift*, researchers focused more on knowledge distillation, which was originally designed to solve teacher-to-student learning problems [61]. A reference work was proposed by *Li et al.* [102], denoted by Learning without Forgetting (LwF). It is the first state-of-the-art method that does not require memory of past classes. It leverages knowledge distillation to minimize the discrepancy between representations of past classes from the previous and current incremental learning iterations. It freezes the past network parameters, trains only the new ones, and then jointly trains all network parameters until convergence.

Learning without Memorizing (LwM) [39] is a distillation-based approach that does not need a memory of past classes. The authors of LwM proposed an information preserving penalty using attention distillation loss that captures the changes in the classifier attention maps to preserve past knowledge. *Lu et al.* [239] proposed a semantic drift compensation. Instead of preventing drift, which most existing methods do, their method estimates the drift of previous tasks when training new tasks to compensate for it to improve performance. The drift is computed at the class-mean-embedding level. This approach is based on a Nearest-Class-Mean (NCM) classifier that does not need a memory of past classes since the past class-mean embeddings are estimated using new data only.

Recently, *Zhang et al.* [244] tackled the *activation drift* problem by using two models. Given a pre-trained model on existing classes and labeled data of new classes, they first train a new model to recognize instances of the new classes; then, they combine the old model and the new model using the Deep Model Consolidation (DMC) module, which exploits the external unlabeled auxiliary data. Thus, the final model suffers less from forgetting the old classes and achieves high recognition accuracy for the new classes. Figure 2.25 illustrates the functioning of this method.

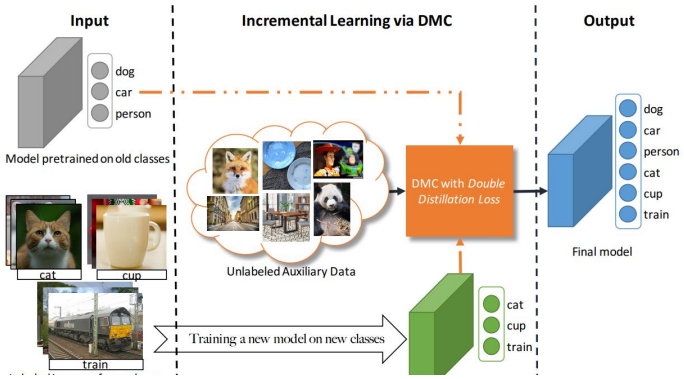

Fig. 2.25 *Zhang et al.* [244] class-incremental learning proposed method.

Replay-based methods. Here we introduce methods that use limited memory of the past. By observing the state-of-the-art, we distinguish two categories: regularization-based methods, which rely on the loss function to regularize the weights update, and bias-removal-based methods, that address recency bias by removing the bias between past and new class scores.

One of the most popular **regularization-based method** is the Incremental Classifier and Representation Learning (iCaRL) [156]. It combines the use of distillation and memory for past class exemplars. The authors used the nearest mean of samples for the classification decision instead of the network probability score. Their classification method aims to reduce the prediction bias due to the imbalance between old and new classes.

Castro et al. [23], addressed the challenge of *catastrophic forgetting* with an end-to-end incremental learning approach (E2EIL). They proposed a representative memory component that can be realized with any deep learning architecture, akin to a sample set for maintaining a small set of samples corresponding to old classes. Their end-to-end model is learned by minimizing the cross-distilled loss, combining two loss functions: cross-entropy to learn the new classes and distillation to retain the previous knowledge corresponding to the old classes. Similarly, [65] adapted the E2EIL concept and added a sophisticated data augmentation, which showed more improvement.

In [252], the authors proposed a multi-model and multi-level knowledge distillation strategy to tackle the *catastrophic forgetting* problem. Instead of sequentially distilling knowledge only from the penultimate model, they directly leverage all previous model snapshots. In addition, they incorporate an auxiliary distillation to preserve further knowledge encoded at the intermediate feature levels. They adapt mask-based pruning to reconstruct all previous models with a small memory footprint to make the model more memory efficient.

Similar to DMC [244], *Lee et al.* [94] proposed a global distillation method that takes advantage of an external dataset to tackle *catastrophic forgetting* using triple distillation. This method has three main stages. In the first stage, a teacher model is trained on new data, then calibrated using the exemplars memory and the external dataset. Then, in the second stage, a triple distillation is applied using the teacher, previous, and ensemble models, which are used with external data only. Finally, fine-tuning is performed in the last stage to tackle the recency bias problem. In contrast, *Hou et al.* [64] proposed an approach that trains a teacher model separately with new data while distilling knowledge using exemplars memory to preserve accuracy in past classes.

A set of **bias-removal-based methods** was proposed to target and directly tackle the recency bias problem. In [227], a bias correction is proposed. It is a recently proposed method that uses a classical knowledge distillation term and adds a linear layer after the prediction layer of the deep model to reduce the bias in favor of new classes.

Maintaining Discrimination and Fairness (MDF), [250] uses distillation loss to maintain Discrimination between past classes. The method consists of two phases: The first phase aims to Maintain Discrimination (MD), where they train a new model on the new and rehearsal data. Thanks to knowledge distillation, this first step allows knowledge transfer from the old model to the new model and maintains Discrimination within old classes. Knowledge distillation loss still cannot help the model treat old and new classes fairly, so they designed a second phase denoted by Maintaining Fairness (MF). In this phase, they propose a method named Weight Aligning (WA) to correct the model trained in the first phase. The corrected model treats old and new classes fairly, improving overall performance.

Pseudo-replay-based methods. These approaches do not store exemplars for past classes in the memory. Instead, they generate synthetic data to represent past classes in the current incremental state. Attracted by the success of Generative Adversarial Networks (GANs), *Wu et al.* [228] used a GAN to create artificial images for past classes. Generated and real examples are mixed to obtain slightly better performance than that of iCaRL [156]. However, the accuracy drops significantly when relying exclusively on artificially generated images. *Ostapenko et al.* [134] introduced a Dynamic Generative Memory (DGM), a synaptic plasticity-driven framework for continual learning. DGM employs conditional generative adversarial networks with learnable connection plasticity achieved by neural masking. They specifically investigate two neural masking variants: applied to layer activation and weights directly. Moreover, they proposed a dynamic network expansion mechanism that ensures the model capacity to accommodate for continually incoming tasks. The amount of added capacity is determined dynamically from the learned binary mask. Instead of storing images

of past classes, DGM uses previous distributions to learn a single generator of past class images incrementally.

2.5.2 Fixed-representation based approaches

Parameter-isolation based methods [165] address *catastrophic forgetting* by allocating specific model parameters to each learning iteration. They can be seen as a basic variant of fine-tuning-based methods.

Inspired by biological brain functioning, *Kemker et al.* [79] proposed a fixed-representation method based on dual-memory system. Separated networks are used for long- and short-term memories to represent past and new classes. A decision mechanism is implemented to decide which network should be used for each test sample. However, memory increases significantly with time since the algorithm stores detailed statistics for each class learned.

Also inspired by the human brain, *Hayes et al.* [57] proposed an incremental learning method called REMIND. The method is also based on an initial representation which is only partially updated afterward. The approach uses a vector quantization technique to store compressed intermediate representations of images, which are more compact than the images themselves. The stored vectors are reconstructed and replayed for memory consolidation.

Deep Streaming Linear Discriminant Analysis (Deep-SLDA) [58] is a recently proposed method based on SLDA [138] algorithm. The network is trained on the first batch of classes and is frozen afterward. During the training, a class-specific running mean vector and a shared covariance matrix are updated. The predictions are made by assigning the label to the closest Gaussian in the feature space defined by the class-mean vectors and covariance matrix.

2.5.3 Parameter-isolation based approaches

This group of incremental learning approaches addresses *catastrophic forgetting* by allocating specific model weights to each learning iteration. Here we distinguish between two types of approaches: (1) those where the architecture of the model extends to accommodate new knowledge (dynamic networks), and (2) those where the complexity of the model is constant (fixed networks).

Dynamic networks. It increases the size of the neural network model to acquire new knowledge. Progressive Neural Networks [165] prevents *catastrophic forgetting* by instantiating a new neural network (a column) for each learning iteration, while deep features are shared through lateral connections between previously learned columns.

Roy et al. [163] proposed an adaptive hierarchical network structure consisting of deep CNN that can extend and learn as new data arrives incrementally. The network grows tree-like to accommodate new classes of data while retaining the ability to classify the previously learned classes. The network organizes incrementally available data into feature-driven superclasses and enhances existing hierarchical CNN models through the ability to self-grow.

Rosenfeld et al. [161] proposed deep network architecture for lifelong learning, which they refer to as Dynamically Expandable Network (DEN). It can dynamically decide its network parameters as it trains on a sequence of tasks to learn a compact overlapping knowledge-sharing structure among tasks. DEN is trained online by performing selective retraining and dynamically expands network capacity upon arrival of each iteration with only the necessary number of units (neurons). This prevents semantic drift by splitting/duplicating units and time-stamping them.

Fixed networks. Unlike dynamic networks, these methods do not modify the neural network architecture. *PackNet* [116] is based on a pruning technique that identifies redundant free weights and uses them to train the network on new classes. The approach cannot learn a large number of classes since the network can not be strongly compressed without significant performance loss.

Based on the *PackNet* idea, *Mallya et al.* [115], proposed a similar solution *Piggyback*. This solution learns to selectively mask a base network's fixed weights to improve performance on a new task. They achieve this by maintaining a set of real-valued weights passed through a deterministic thresholding function to obtain binary masks that are then applied to existing weights. They learn binary masks appropriate for the task by updating the real-valued weights through backpropagation. The approach increases the model complexity because extra parameters are added each time to include new learning iterations.

2.5.4 Incremental activity recognition

Unlike many incremental learning applications that can be found for the image classification task, few incremental learning applications focus on activity recognition. Incremental learning of activities is more complex than static images. This is mainly because the spatial and temporal features of the activity sequence must not be forgotten when learning new classes. On the other hand, the representations of activity data (e.g., videos) require more computational resources for processing and storage, which makes the replay or pseudo-replay-based incremental learning methods more challenging.

Ryoo et al. [167] have proposed a method for incrementally learning new activity classes from videos. They represented activities using visual words to describe their videos and incrementally learned new words for existing/new activities while providing training

examples. Each word is a set of local spatiotemporal features, and the activity is learned by modeling the distribution of these words in their videos. They also proposed an algorithm to learn an optimal set of words per class gradually. New visual words are generated when videos from an existing or new activity class are provided, while old words are updated or merged based on the new observations. A histogram of the visual words needed for recognition is also incrementally generated for each activity class. The histograms are created or updated after each video arrival, preserving the features of the previous distribution.

Ma et al. [112] proposed a *Grow When Required network (GWR)* based video class incremental learning framework for action classification. *GWR* learns knowledge incrementally by modeling the manifold of video frames for each encountered activity class in feature space. They also introduce a knowledge consolidation method to separate the feature manifolds of the old class and the new class.

More recently, based on the previously introduced LwM [39], which is a knowledge distillation-based incremental learning method, *Park et al.* [141] proposed a class incremental learning method for action recognition in videos. To reduce the memory for past class exemplars storage and the training costs, they adopted a frame-based feature representation method to store exemplars for the tasks learned in the past. This representation is based on the Temporal Shift Module (TSM) [103]. In addition to LwM, they exploited an attention method over a time-channel space to facilitate action recognition in a class-incremental learning scenario.

The first and the only work that addressed the *catastrophic forgetting* problem for egocentric hand gesture recognition was proposed by *Wang et al.* [222]. Their proposed method, denoted by (CatNet), is based on the previously introduced iCaRL method [156] and 3D CNN. The key idea of *CatNet* is that they selected some previous class samples that are the most pertinent of the old class. Selected video samples and their predictions are stored. This is achieved by mainly learning the feature representation from RGB and depth hand gesture sequences. The cached samples play two roles during class-incremental learning. First, the selected samples are used to compute a feature vector for each class that is used for inference. Second, following the *iCaRL* concept, the prediction is used to compute the distillation loss during the training.

2.5.5 Conclusion

In this section, we presented state-of-the-art methods related to the final component of our targeted activity recognition framework, which aims at incrementally integrating annotated unknown samples. We first described the class incremental learning problem. We also introduced the central issue of neural network-based incremental learning denoted by the

catastrophic forgetting problem and highlighted its causes. Then, we presented state-of-the-art approaches to overcome this phenomenon. Finally, we presented state-of-the-art methods related to incremental human activity recognition.

We can retain from this state-of-the-art study that the most investigated and promising incremental learning methods are these based on fin-tuning the initial model to learn both old and new classes simultaneously. The primary issue with this category of methods is that they require storing information about old training classes to fight the *catastrophic forgetting* problem. However, in the long-term continual learning, a naive storing strategy can cause memory capacity problems. We presented existing solutions to this problem, such as no-reply memory or pseudo-memory replay methods. Nevertheless, these methods are still struggling with the catastrophic forgetting problem, and they do not apply to some specific tasks. Based on these observations, in chapter 4, section 4.4, we present our adopted fin-tuning-based method for incremental hand activity recognition, which uses a new memory replay solution. The proposed solution aims to maximize information about old training data to avoid the *catastrophic forgetting* problem while minimizing the storage memory usage.

2.6 Conclusion

Following the four components we propose in this thesis (chapter 1, section 1.2), we introduced in this chapter each component's general idea and presented its related existing state-of-the-art approaches. Thus, we began by reviewing state-of-the-art existing first-person hand activity recognition methods, which we divided into three distinct categories. We provided a description, advantages, and limitations for each category; and presented the available state-of-the-art datasets for hand activity recognition evaluation. We also gave an overview of the state-of-the-art methods related to the rest of the proposed components. Thus, we explored the following research fields:

- **Open-set recognition.** To detect unknown hand activities while correctly classifying the known ones.
- **Unsupervised domain adaptation and metric learning.** To automatically cluster, then manually annotate detected unknown hand activities.
- **Incremental learning.** To integrate annotated unknown activities into the recognition and unknown activity detection models.

As we mentioned in the conclusion of each section of this chapter, these state-of-the-art studies allowed us to make the right choices, proposing efficient solutions that satisfy our objectives. In the following chapters, we introduce our proposed methods.

Chapter 3

Efficient and low-cost Learning Pipelines for Hand Activity Recognition

Contents

3.1 Introduction

As we explained in the first chapter, subsection 1.2.1, hand activity recognition is one of the main components of the desired user activity recognition framework. The activity recognition component will enable the framework to comprehend the AR user's hand activity to help, direct, and orient him while he performs complicated tasks.

To this end, in this chapter, we present our two solutions for hand activity recognition from the first-person viewpoint that meet the AR context and address the challenges introduced in chapter 1, section 1.2.1. The remainder of the chapter will be organized as follows:

- **Section 3.2** details our proposed learning pipeline for 3D skeleton-based hand activity recognition. First, we introduce the approach and its main contributions. Then we detail each step of the proposed approach. Finally, we present the experiments that evaluate and validate the approach by discussing and comparing the obtained results with existing state-of-the-art approaches.

- **Section 3.3.** introduces our proposed learning pipeline for RGB-based hand activity recognition. This second solution aims at overcoming the previous one's imitations. First, we define the approach and highlight its contributions. Then we detail each step of the proposed method. Next, we introduce the combination of our RGB-based and 3D skeleton-based solutions for hand activity recognition. Finally, we present the experiments that evaluate and validate the proposed approaches by comparing the results obtained with existing state-of-the-art methods.

- **Section 3.4** concludes the chapter.

3.2 3D skeleton-based hand activity recognition

As discussed in the previous chapter, section 2.2, the 3D skeleton data provide a robust high-level description of common problems in RGB imaging, such as background subtraction and light variation. To this end, many skeleton-based approaches have been proposed. Most of them are based on end-to-end DL process [40, 216], which have been shown to be effective when a large amount of data is available. Nevertheless, providing large labeled datasets for some industrial applications is still difficult and expensive due to manual data labeling. On the other hand, pure HC feature-based approaches [186, 37, 74, 249, 69] can deal with limited data. However, they have difficulty learning temporal dependencies along the activity sequence. As an alternative to balance performance and data acquisition cost, hybrid methods combine DL and pure HC methods [5, 29, 108, 248].

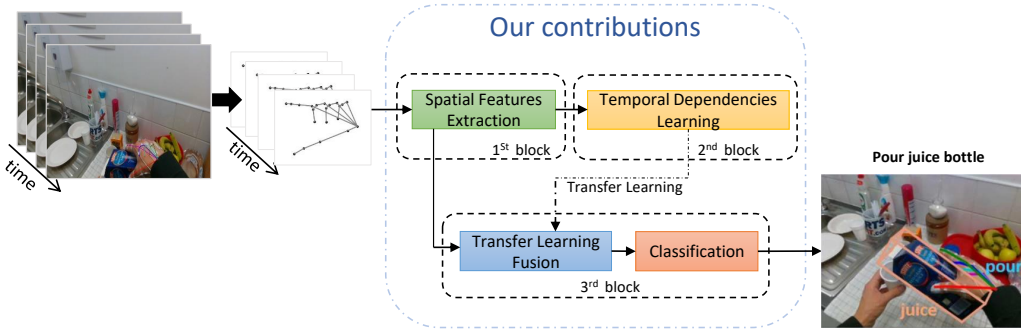

Fig. 3.1 This figure illustrates our proposed learning pipeline for 3D skeleton-based first-person hand activity recognition. For a given 3D hand skeleton activity sequence, the spatial features are extracted using existing and new hand-crafted methods. Then, in the second block, the temporal dependencies are learned. In the last block, a hand activity sequence classifier is learned using a post-fusion strategy applied to the previously learned temporal dependencies. Only the first and last blocks are involved in the test predictions.

Motivated by all these observations, in this section we present our proposed hybrid approach for 3D skeleton-based first-person hand activity recognition (see figure 3.1). We highlight the contributions as follows:

- A novel hybrid learning pipeline for first-person hand activity recognition consists of three sequential blocks (as shown in figure 3.1): In the first block, we extract the spatial features using our proposed selection of existing and new HC feature extraction methods. Then, in the second block, we differ from the existing methods by learning the temporal dependencies independently from each HC feature using a simplified

separate neural network to avoid the overfitting problem. Finally, we use the knowledge from the previous block to classify the activities using a tuning post-fusion strategy. Once learning is complete, only the first and last blocks are used for predictions. This multi-stage learning pipeline allows training with a limited number of samples while ensuring good accuracy.

- A combination of three local and global HC feature extraction methods for 3D skeleton-based first-person hand activity recognition, which we summarize as follows: (1) Inspired by [186], we use a Shape of Connected Joints (SoCJ) that characterizes the activity sequence by the variation of the physical hand shape at each time step. (2) Our proposed Intra/Inter Finger Relative Distances (IIFRD), which also relevantly characterizes the activity sequence at each time step, by the variation of inter-finger relative distances between physically adjacent finger pairs, and Intra-Finger Relative Distances, which belong to the distance between two opposite joints of a pair of directly connected finger segments. (3) Since SoCJ and IIFRD focus only on the local features of the hand at each time step, we proposed a complementary HC method called Global Relative Translations (GRT), which focuses on the global features of the whole sequence by using the displacement of the hand during activity.

The remainder of this section details our proposed hybrid approach following the illustration of figure 3.1. Subsection 3.2.1 presents the first block of our proposed pipeline, which extracts the HC features from 3D hand skeleton data. In subsection 3.2.2, we present the second block that learns the temporal dependencies. Subsection 3.2.3 presents the last block of the proposed pipeline. Once the temporal dependencies learning is ended, we transfer and exploit the knowledge from the previous block to learn classifying activities in the last block. Subsection 3.2.4 is devoted to experiments and the evaluation of the proposed learning pipeline. Subsection 3.2.5 concludes the section.

3.2.1 Hand-crafted features extraction

The proposed pipeline inputs a 3D hand skeleton activity sequence that we denote by $x(t)$. At each time step t, the hand is represented by a configuration of physically connected n joints $\{J_j^t\}_{j=1:n}$. Each joint is represented by 3D Cartesian coordinates forming a set of segments that yields the hand bones, the phalanges, and metacarpals (figure 3.2). We define and formulate the activity sequence as follows:

$$x(t) = \{\{J_j^t\}_{j=1:n}\}_{t=1:T} \tag{3.1}$$

where T is the max length of the sequence.

In order to exploit the 3D geometrical information in the first block, we use three HC feature extraction methods that provide relevant features for first-person hand activity recognition. Before proceeding to feature extraction, the 3D hand skeleton data are normalized, such as all the hands of all the subjects are adjusted to the same average size while keeping the angles intact. In the following descriptions, we model the hand skeleton by fixing the number of joints to $n = 21$.

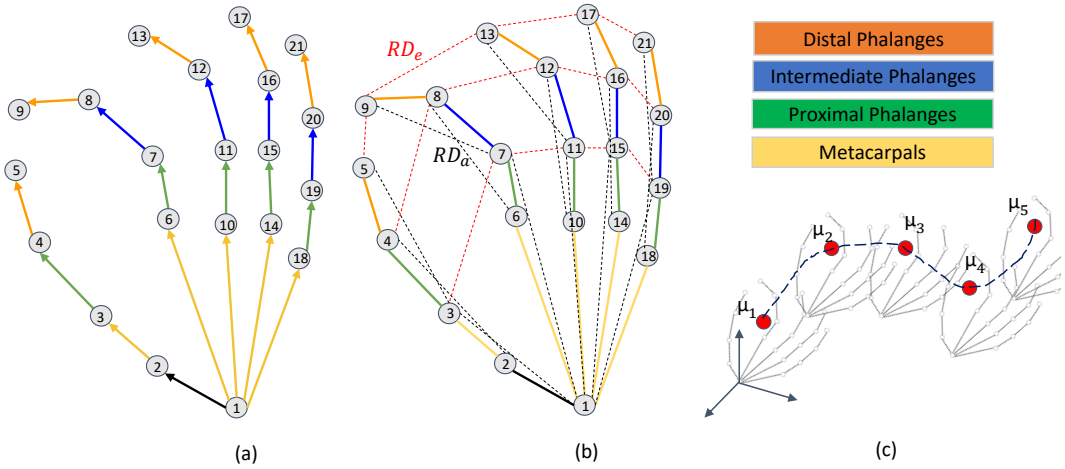

Fig. 3.2 The proposed selection of hand-crafted features. (a) 3D vectors between physically connected joints represent the shape of Connected Joints (SoCJ), the hand shape [186]. (b) Our proposed Intra/Inter Finger Relative Distances (IIFRD) characterize the activities by the high variation of intra-finger distances RD_a (in black) and inter-finger distances RD_e (in red). (c) Our proposed Global Relative Translation aims to characterize the activity sequence by translating the joints centroid at each time step.

Shape of Connected Joints (SoCJ). Inspired by [186], we use the SoCJ to represent the variation of the hand shape during the activity. For each finger, we compute the 3D vectors between physically connected joints, from the wrist up to the fingertips (Figure 3.2 (a)). Let $Finger_1 = \{J_j\}_{j=1:5}$ be a set of joints that are ordered such as they represent the physical connections of the thumb finger and wrist joint J_1 as shown in figure 3.2 (a), the $SoCJ(Finger_1)$ can be computed as follows:

$$SoCJ(Finger_1) = \{J_j - J_{j-1}\}_{j=5:2} \tag{3.2}$$

By applying the SoCJ to all the fingers, as a result, for each time-step t, we obtain a feature descriptor $\{SoCJ(Finger_l^t)\}_{l=1:5} \in \mathbb{R}^{4\times5\times3}$, where $Finger_l$ is the l-th finger. $\Psi_{SoCJ}(.)$

denote the SoCJ method applied to the entire activity sequence $S(t)$ that we define as follows:

$$\Psi_{SoCJ}(x(t)) = \{\{SoCJ(Finger_l^t)\}_{l=1:5}\}_{t=1:T} \tag{3.3}$$

Intra/Inter Finger Relative Distances (IIFRD). We exploit the periodic variation of the intra-finger and inter-fingers relative distances, which relevantly characterizes the activity sequence (figure 3.2 (b)).

- The **intra-finger** relative distances that we denote RD_a give strong internal dependencies between the finger's connected segments. It represents the distance between two opposite joints of a pair of directly connected segments from each fingertip down to the wrist (Figure 3.2 (b) in black). Let us take $Finger_1$ as described previously. The $RD_a(Finger_1)$ can be computed as follows:

$$RD_a(Finger_1) = \{\rho(J_j, J_{j-2})\}_{j=5:3} \tag{3.4}$$

  where ρ is the Euclidean distance. By applying the RD_a to all fingers, for each time-step t, we get a feature descriptor $a(t) = \{RD_a(Finger_l^t)\}_{l=1:5} \in \mathbb{R}^{15}$.

- The **inter-finger** relative distances that we denote by RD_e (figure 3.2 (b) in red), give external dependencies between adjacent fingers pairs. For instance, let us take $Finger_1$ as described previously and $Finger_2 = \{J_j\}_{j=7:9}$, two sets of connected joints that refers to the thumb and the index fingers, respectively. The $RD_e(Finger_1, Finger_2)$ is computed as follows:

$$RD_e(Finger_1, Finger_2) = \{d(J_j, J_{j+4})\}_{j=3:5} \tag{3.5}$$

  By applying the RD_e to the four pairs of adjacent fingers, for each time-step t, we obtain a feature descriptor $e(t) = \{RD_e(Finger_l^t, Finger_{l+1}^t)\}_{l=1:4} \in \mathbb{R}^{12}$.

Finally, by concatenating the two descriptors $a(t)$ and $e(t)$, for each time-step t, we obtain a final feature descriptor $\{a(t), e(t)\} \in \mathbb{R}^{15+12}$. We denote by $\Psi_{IIFRD}(.)$ the IIFRD method applied to the entire activity sequence $S(t)$ that we define as follows:

$$\Psi_{IIFRD}(x(t)) = \{a(t), e(t)\}_{t=1:T} \tag{3.6}$$

Global Relative Translations (GRT). Unlike the IIFRD and SoCJ descriptors, which only consider the local features that belong to the fingers motion at each time step, the GRT characterizes the activity sequence by computing the relative displacement of all the hand

joints along the sequence time steps (figure 3.2 (c)). To this end, for each sequence, we fix the wrist joint J_1^0 of the first time-step $t = 0$ as the origin. Then, we transform all remaining joints of the sequence to this new coordinate system as follows:

$$\acute{J}_j^t = J_j^t - J_1^0 \tag{3.7}$$

where the $\acute{J}_j^t$ the new transformed j-th joint at the time-step t. Once the transformation is done, at each time-step, we compute the centroid of the transformed joints $\mu_t = \frac{1}{21}\sum_{j=1}^{21}\acute{J}_j^t$. We denote by $\Psi_{GRT}(.)$ the application of GRT to the entire sequence $S(t)$, which we define as follows:

$$\Psi_{GRT}((x(t)) = \{\mu_t\}_{t=1:T} \tag{3.8}$$

The GRT gives discriminate complementary information to the IIFRD and the SoCJ by considering the global trajectory of the hand along the activity. In subsection 3.2.4 we quantitatively show the benefit of this complementary information.

3.2.2 Temporal Dependencies Learning

Learning long and complex activities requires considering the temporal dimension to benefit from the long-term dependencies between sequence time steps. To this end, we use LSTM cells to learn these long/short-term dependencies for their outstanding success and capabilities. Moreover, unlike traditional RNNs, LSTMs overcome the vanishing gradient problem by using a specific circuit of gates [62].

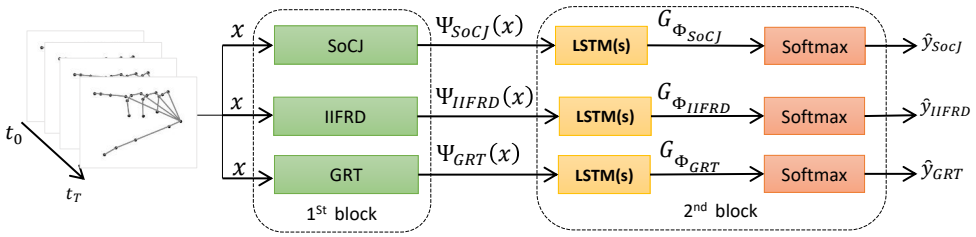

Fig. 3.3 Illustration of the first and second blocks of the proposed learning pipeline. For each hand-crafted feature descriptor (SoCJ, RD, and GRT) seen in figure 3.2, a Neural Network composed of staked LSTM layers and a softmax layer is trained independently to learn temporal dependencies.

[5, 108] concatenate different types of feature spaces as one input vector, which may complicate the input and confuse the neural network. In contrast, for each HC features descriptor (seen in subsection 3.2.1), we train a simple neural network separately that

consists of staked LSTM layers followed by a softmax layer to classify activities. Therefore, we train three neural network separately, as shown in figure 3.3.

More formally, lets $\{\Psi_k(x)\}_{k=1:3}$ be the set of the three feature descriptors corresponding to Eq.3.3, Eq.3.6 and Eq.3.8 defined in subsection 3.2.1, where x is the activity sequence input. For each feature descriptor $\Psi_k(x)$, we model the temporal dependencies with a composite function $G_{\phi_k}(\Psi_k(x))$, where $G_{\phi_k}(\cdot)$ is the k-th LSTM sub-network with ϕ the learnable parameters, while the output of $G_{\phi_k}(\cdot)$ refers to the last hidden state of the last LSTM unit. For each network, we define a cross-entropy loss function $\mathscr{L}_k$ as follows:

$$\mathscr{L}_k = -\sum_{j=1}^{C} y^j log(\hat{y}_k^j) \tag{3.9}$$

where C is the number of classes, y^c the target label, and $\hat{y}_k^c$ the softmax output that refers to the predicted label. The temporal learning parameters are optimized by minimizing over a labeled dataset:

$$\phi_k^* = \arg\min_{\phi_k} \mathscr{L}_k(y, \hat{y}_k) \tag{3.10}$$

At the end of the training, as a result, we have a set of three trained LSTM sub-networks with ϕ_k^* an optimized parameters:

$$\{G_{\phi_k^*}(\Psi_k(x))\}_{k=1:3} \tag{3.11}$$

We note that this second block aims to learn the temporal dependencies, and all the classification results $\hat{y}_k$ are ignored. Only the results shown in Eq.3.11 are needed for the next block.

This pre-training strategy of multiple networks avoids the fusion of different feature spaces, reducing input complexity and noise learning. It also allows the LSTM to focus only on independently learning over one specific feature input, which also helps to avoid the over-fitting problem [236].

3.2.3 Post-fusion Strategy and Classification

Once the temporal dependencies are learned in the second block, we proceed to the final classification. To this end, we train another neural network, a multi-input one this time that exploits the resulted three pre-trained LSTM layers introduced in subsection 3.2.2 that we transfer with fixed optimized parameters ϕ^* as illustrated in figure 3.4.

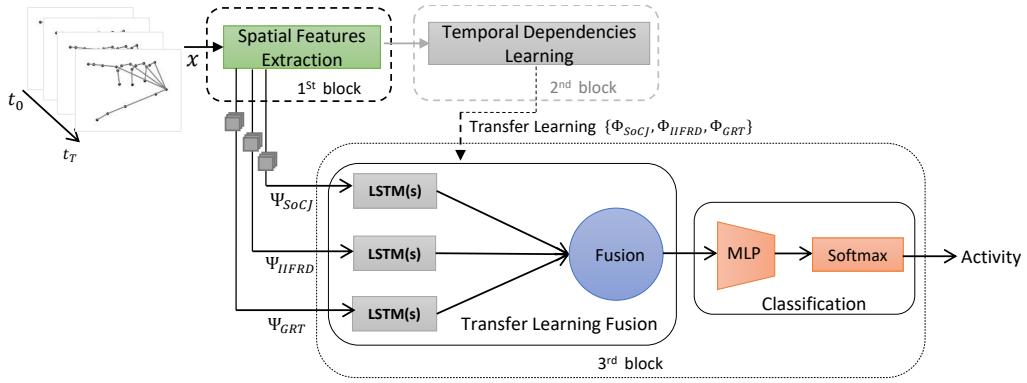

Fig. 3.4 Illustration of the third block of the proposed pipeline. Once the temporal dependencies are learned in the second block. The LSTM layers are transferred to the third block with fixed parameters. Their outputs are fused and fed into a MLP followed by a softmax layer for the final classification.

Seeking to ensure the best classification accuracy, the three parallel outputs branches of the transferred LSTMs are concatenated, then fed into a Multi Layers Perception (MLP) that consists of two (FC) layers, followed by a softmax layer (Figure 3.4). We model this network as shown in Eq.3.12, where F_γ is a MLP+softmax with learnable parameters γ, and *Concat* is the concatenation function:

$$F_\gamma(Concat(\{G_{\phi_k^*}(\Psi_k(x))\}_{k=1:3})) \tag{3.12}$$

The learnable parameters γ are optimized using the same loss function as in the previous block by minimizing over the same training dataset. Note that for the test predictions, only this network is involved using the three HC features descriptors as a multi-input. The proposed post-fusion strategy ensures a good accuracy score through tuning between the outputs of the pre-trained network. In subsection 3.2.4, we quantitatively show the efficiency of this strategy compared to other traditional fusing and classification methods.

3.2.4 Experiments

Datasets

To evaluate the proposed learning pipeline we used the two previously introduced FPHA [48] and DHG 14/28 [186] datasets.

- **FPHA dataset.** For the proposed method, we only need the 3D coordinates of the hand joints. For all the experiments, we used the setting proposed in [48], with the same data distribution: 600 activity sequences for training and 575 for testing.

- **Dynamic Hand Gesture (DHG) 14/28 dataset.** For our experiment, we only need the 3D hand skeleton joints. We ignored the palm center and only considered the remaining 21 hand joints. For the training and testing data split, we used the same configuration proposed in [186].

Implementation details

We detail the implementation used in the experiments as follows:

- **The learning of temporal dependencies.** For every extracted HC feature, we trained different configurations of separated neural networks that consist of 1,2,3, and 4 staked LSTM layers followed by a softmax. We selected the best configuration that gives the best accuracy score: only one LSTM layer of 100 units for the FPHA dataset and two staked LSTMs of 200 units for the DHG 14/28 dataset. We set the probability of dropout to 0.5 (outside and inside the LSTM gates). We use Adam with a learning rate of 0.001 for the optimization. All the networks are trained with a batch size of 128 for 2000 to 3000 epochs. We also padded all the sequence lengths to 300-time steps per sequence.

- **Post-fusion and classification.** Once all the temporal dependencies are learned (end of block 2), in the post-fusion step, we recover the pre-trained LSTM networks, fix all their weights, and discard the Softmax layers. Then, the three outputs branches from the three parallel transferred LSTMs are concatenated and followed by an MLP that consists of two dense layers of 256 and 128 neurons, respectively, equipped with a *Relu* activation function. At the end of the network, a Softmax layer is used for the final classification. This network is trained until 100 epochs, with the same batch size and optimization parameters as the previous networks. Our implementations are based on the Keras framework.

State-of-the-art comparison

Table 3.1 shows the accuracy of our approach compared with the state-of-the-art approaches on the FPHA dataset. We note that the accuracy results of [46, 131, 133, 151, 240, 213, 40, 249, 44] and [66, 67] are reported by [48] and [126] respectively, where the recognition may need the full body joints instead of hands and some of them might not be tailored for hand activities.

Table 3.1 Test accuracy comparison of our proposed approach and the state-of-the-art approaches on the FPHA dataset. The bests results are marked in bold.

Method	**Color**	**Depth**	**Pose**	**Acc.(%)**
Feichtenhofer et al., 2016 [46]	✓	✗	✗	61.56
Feichtenhofer et al., 2016 [46]	✓	✗	✗	69.91
Feichtenhofer et al., 2016 [46]	✓	✗	✗	75.30
Ohn-Bar and Trivedi, 2014 [131]	✗	✓	✗	59.83
Ohn-Bar and Trivedi, 2014 [131]	✗	✓	✓	66.78
Oreifej and Liu, 2013 [133]	✗	✓	✗	70.61
Rahmani and Mian, 2016 [151]	✗	✓	✗	69.21
Garcia-Hernando et al., 2018 [48]	✗	✗	✓	78.73
Garcia-Hernando et al., 2018 [48]	✗	✗	✓	80.14
Zanfir et al., 2013 [240]	✗	✗	✓	56.34
Vemulapalli et al., 2014 [213]	✗	✗	✓	82.69
Du et al., 2015 [40]	✗	✗	✓	77.40
Zhang et al., 2016 [249]	✗	✗	✓	85.39
Garcia-Hernando et al., 2018 [48]	✗	✗	✓	80.69
Fang Hu et al., 2015 [44]	✓	✗	✗	66.78
Fang Hu et al., 2015 [44]	✗	✓	✗	60.17
Fang Hu et al., 2015 [44]	✗	✗	✓	74.60
Fang Hu et al., 2015 [44]	✓	✓	✓	78.78
Huang and Gool, 2016 [66]	✗	✗	✓	84.35
Huang et al., 2016 [67]	✗	✗	✓	77.57
Tekin et al., 2019 [205]	✓	✗	✗	82.26
Zhang et al., 2019 [248]	✗	✗	✓	82.26
Lohit et al., 2019 [109]	✗	✗	✓	82.75
Nguyen et al., 2019 [126]	✗	✗	✓	**93.22**
Rastgoo et al., 2020 [155]	✓	✗	✓	**91.12**
Our	✗	✗	✓	**96.17**

The best performing approaches among state-of-the-art methods are the neural network based on SPD manifold learning [126], and the multi-modal approach proposed by *Razieh et al.* [155], which gives 93.22% and 91.12% of accuracy respectively, 3.26% inferior to our proposed approach. The remaining methods are outperformed by our approach by more than 11% of accuracy.

Table 3.2 Accuracy comparison of our proposed approach and the state-of-the-art approaches on the DHG-14/28 dataset. The bests results are marked in bold.

Method	**Color**	**Depth**	**Pose**	**Accuracy (%)**	
				14 gest	**18 gest**
Oreifej and Liu, 2013 [133]	✗	✓	✗	78.53	74.03
Devanne et al., 2015 [37]	✗	✗	✓	79.61	62.00
Huang and Gool, 2016 [66]	✗	✗	✓	75.24	69.64
Ohn-Bar and Trivedi, 2014 [131]	✗	✗	✓	83.85	76.53
Chen et al., 2017 [29]	✗	✗	✓	84.68	80.32
Smedt et al., 2016 [186]	✗	✗	✓	88.24	81.90
Devineau et al., 2018 [38]	✗	✗	✓	91.28	84.35
Nguyen et al., 2019 [126]	✗	✗	✓	94.29	89.40
Maghoumi and LaViola, 2018 [114]	✗	✗	✓	94.50	**91.40**
Avola et al., 2019 [5]	✗	✗	✓	**97.62**	**91.43**
Our	✗	✗	✓	**95.21**	90.10

Table 3.2 shows that our proposed approach is achieving the state-of-art results on the DHG-14/28 dataset, even though our selected HC features methods are adapted to the first-person hand activity recognition and not to the hand gesture recognition problem. The approach proposed by [5] outperforms all the state-of-the-art approaches, including ours, thanks to their proposed HC features which are well adapted to American Signe Language (ASL) and semaphoric hand gestures. Furthermore, the time sampling strategy used in [5] allows them to better classify the least dynamic and shortest gestures, unlike our proposed approach, which is adapted to deal with the hand activities where the hand is supposed to be more dynamic.

Hand-crafted features analysis

SoCJ Vs. IIFRD Vs. GRT. In order to analyze the effectiveness of the selected HC features, we evaluated each one independently using a simplified end-to-end neural network architecture composed of one LSTM layer of 100 units with a dropout of 0.5 (outside and inside the LSTM gates) and a softmax layer. We also evaluated possible HC feature combinations using our approach with the same configuration introduced in subsection 3.2.4.

The results in table 3.3 show that the SoCJ and IIFRD alone can achieve a good accuracy of 89.91% and 88.17%, respectively. As expected, the GRT alone cannot classify activities

Table 3.3 Test accuracy results on the FPHA dataset. The selection of hand-crafted features independently and the combinations using our proposed approach.

Hand-crafted Feature	**Acc.(%)**
Shape of Connected Joints (SoCJ)	89.91
Intera/Inter Finger Relative Distance (IIFRD)	88.17
Global Relative Translations (GRT)	58.26
IIFRD + GRT	93.73
SoCJ + GRT	92.17
SoCJ + IIFRD	93.91
SoCJ + IIFRD + GRT	**96.17**

by achieving only 58.26% of accuracy. However, it boosts the performance if combined with the SoCJ, the IIFRD, or both, by achieving the best accuracy of 96.17%. This can be explained by the fact that the SoCJ and the IIFRD focus on the local features based on the motion of the fingers, ignoring translations between the activity sequence time-steps, while the GRT focuses on the global feature based on the displacement of the hand during the activity, which provides a piece of crucial complementary information. The combination of the three selected HC features allowed us to overcome the commonly confused classes *"open wallet"* and *"use calculator"* even if the hand poses are dissimilar but more subtle [48]. Nevertheless, we still get confusion between *"open wallet"* and *"flip sponge"* classes due to the limited displacement of the hand, the shortness of the activities, and the limited number of samples in the dataset compared to the other classes [48]. Figure 3.5 shows the confusion matrices.

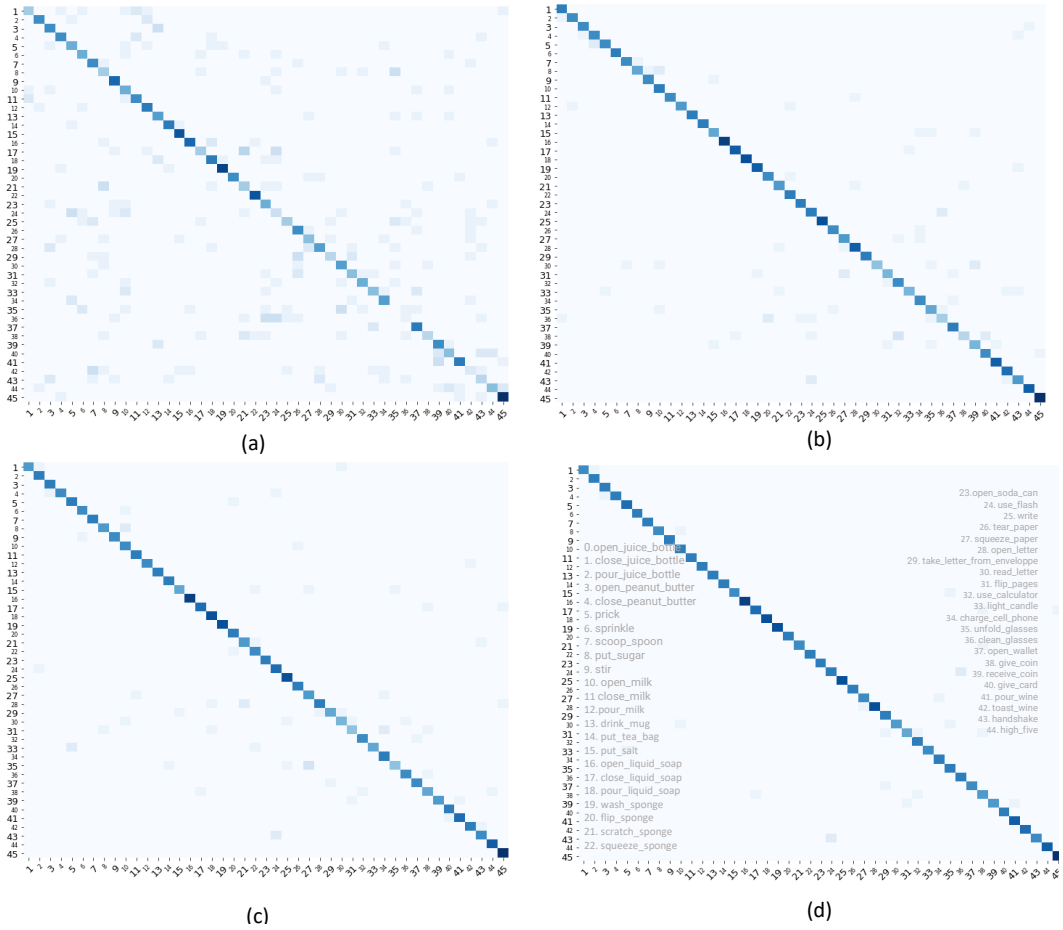

Fig. 3.5 Confusion matrices. (a) using the GRT only. (b) using the IIFRD only. (c) using the SoCJ only. (d) using the combination of the three hand-crafted features the GRT, the IIFRD, and the SoCJ.

Confusions made by SoCJ and IIFRD. Figure 3.5 (b) and (c) give the confusion matrices for the IIFRD and SoCJ, respectively. As we can see, the two local features are complementary to each other since most of the confusions seen in the IIFRD (e.g., *"give coin"* with *"take lettefrom enveloppe"*, and *"wash sponge"* with *"unfold glasses"*) are solved in the SoJC. On the other hand, the confusion is seen in the SoJC (e.g., *"tear paper"* with *"charge cell phone"*, and *"light candle"* with *"read letter"*) is solved in the IIFRD.

Hand-crafted Feature best combination. In figure 3.5 (d), we give the confusion matrix that refers to the combination of the three HC features (the IIFRD, the SoCJ, and the GRT) using our proposed learning pipeline. As we can see, the matrix is very sparse, ensuring the best test accuracy score. This combination solved most of the confusion. Nevertheless, we still get confused about some challenging classes we discussed above. These confusions may

also occur due to the shortness of the activities and the limited number of samples in the dataset compared to the other classes [48].

GRT results explanation. The GRT cannot achieve good accuracy on its own because it does not consider the local features related to the fingers' motion. Figure 3.6 gives the variation of the GRT as the total average distance covered by the hand joints centroid for each class of the FPHA data set that we computed as follows:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} d(\mu_{t-1}^{i}, \mu_{t}^{i}) \tag{3.13}$$

where M is the number of samples of the class, T is the maximum sequence length, and μ_t is the hand joints centroid as defined in subsection 3.2.1. Figure 3.5 (a), and figure 3.6 show that most confused classes have a short total distance covered by the hand joints centroid during the activity.

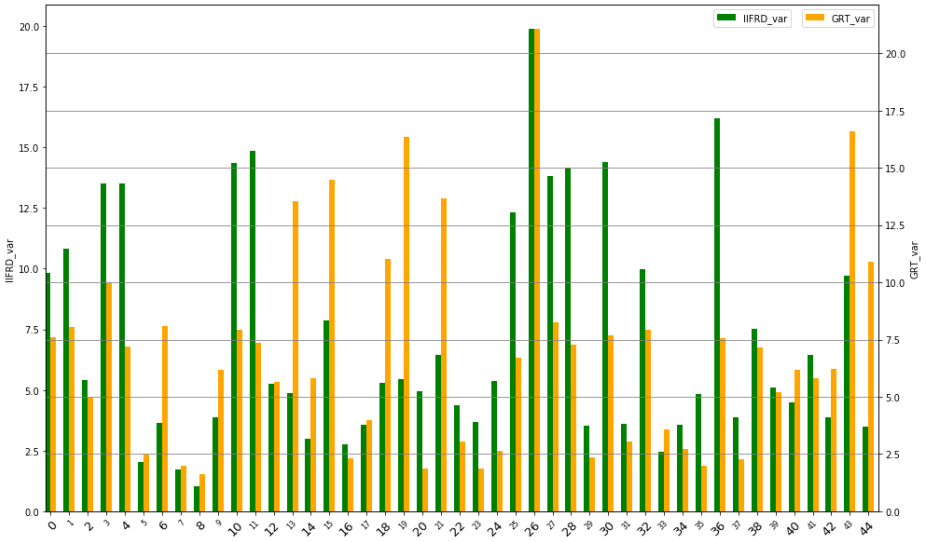

Fig. 3.6 Bar graphs of the IIFRD and the GRT variations according to each class of the FPHA dataset.

IIFRD results explanation. Figure 3.6 gives the IIFRD variation for each class of the FPHA data set, that we computed as follows:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} |SumDist_{t+1} - SumDist_{t}| \tag{3.14}$$

Where $SumDist_t = \sum a(t) + \sum e(t)$. From figure 3.6 and figure 3.5 (b), we can see that the classes with less variation are the same ones with more confusion.

Post-fusion Strategy and Classification analysis

We compared our post-fusion strategy with three traditional baselines, the early, the slow, and the late fusion that we define as follows:

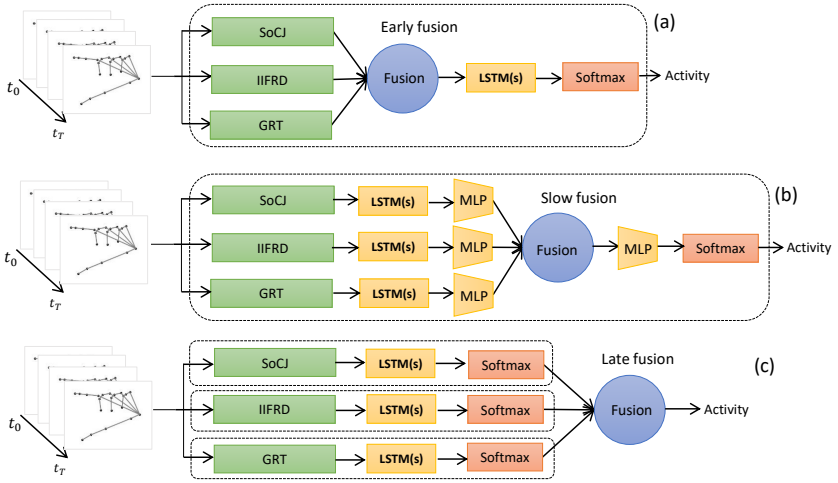

Fig. 3.7 Hybrid fusion and classification baselines, (a) **Early fusion** is an end-to-end architecture where the extracted hand-crafted features are concatenated and fed into a temporal learning network followed by a classifier. (b) **Slow fusion** is an end-to-end architecture where, for each extracted hand-crafted feature, the temporal dependencies are learned separately, then concatenated and fed into a classifier. **(c) Late fusion** is multi-stream learning, where an end-to-end temporal network is trained separately for each extracted feature, and at the end, a majority vote is applied to their classifier outputs.

- **Early fusion.** (Figure 3.7. (a)) As in [5, 108], we concatenate our extracted HC features descriptors in one unified vector that we fed into deep-staked LSTM layers of 200 units followed by a softmax layer. We evaluated the baseline with a configuration of 2, 3, and 4 staked layers, and then the best accuracy results were selected for the comparison.

- **Slow fusion.** (Figure 3.7. (b)) As in [29], for each extracted HC features descriptor, we used two stacked LSTM layers of 200 units, followed by a Fully Connected layer (FC) of 128 neurons. The outputs from the three parallel FC branches are concatenated and followed by two sequential FC layers of 256 and 128 neurons, respectively. A softmax layer is used for the classification at the end of the network. A dropout layer follows all the layers, and all the FC layers are equipped with a *relu* activation function.

- **Late fusion.** (Figure 3.7. (c)) In contrast to the previously introduced end-to-end baselines, in this architecture, a neural network composed of an LSTM layer of 100

units and a softmax layer is trained separately for each HC feature. At the end of the training, a majority vote is applied by summing the softmax outputs scores.

Table 3.4 Accuracy results on 50% and 100% of the 600 FPHA dataset training samples. For the test, all the 575 testing samples are kept. We compared our proposed approach with three traditional fusion and classification baselines.

Architecture	300 samples **Acc.(%)**	600 samples **Acc.(%)**
Early fusion	75.65	90.95
Slow fusion	63.47	86.43
Late fusion	76.26	93.73
Our	**79.78**	**96.17**

We trained our network architecture and the selected baselines with 100%, then only 50% of the 600 training samples of the FPHA dataset that belongs to subjects 1, 3, and 6. For the test, we kept all the 575 testing samples.

Our proposed approach outperforms the baselines with more than 3.52% using 50% and 2.44% of accuracy using 100% of 600 training samples, respectively, which confirms the effectiveness of our fusion strategy. Moreover, our approach is achieving state-of-the-art performance by using only 300 samples (half) of the FPHA dataset training samples, confirming the capability of our method to provide a good recognition result while learning on a limited amount of data. Thanks to its simplified architecture, the early fusion outperforms the slow fusion, which is more complex and implies the over-fitting problem. The late fusion outperforms both, thanks to its simplified neural networks trained independently, which helps to overcome the over-fitting problem. The late fusion performs well, but its naive fusion can not ensure good tuning between the neural network outputs.

Inference run-time. Table 3.5 shows that the early and the late fusion are more efficient than the architecture we propose because their neural network architectures are less complex and have fewer parameters. The slow fusion performs poorly because the architecture is very complex with more parameters.

Table 3.5 Average prediction run-time comparison of the three selected baselines with our proposed architecture using FPHA data set test samples.

Architecture	Early fusion	Slow fusion	Late fusion	Our
Run-time (s)	0.33	0.64	0.45	0.58

3.2.5 Conclusion

In this section, we presented a novel learning pipeline for first-person hand activity recognition. The pipeline consists of three blocks. The first block is a novel combination of hand-crafted feature extraction methods. The second block is our multi-stream learning strategy for temporal dependencies. In the last block, we present our proposed post-fusion strategy, which is shown to be more efficient than other traditional fusion methods. The proposed approach was evaluated on two real datasets and showed good accuracy results.

In the next section, we present our second approach for first-person hand activity recognition, which uses color information in addition to skeletal data to avoid the ambiguous case where the manipulated objects in different activities may have the same dimension but different colors.

3.3 RGB-based and Multi-modal-based (RGB and 3D hand skeleton) hand activity recognition

In addition to its low-cost acquisition, RGB image sequences consider appearance and motion information, unlike depth maps or 3D skeletal data, which focus only on the motion. Whereas most egocentric activities are centered around hand-object interactions, appearance is essential to perform inter-objects and inter-scenarios differentiation.

As discussed in the previous chapter, section 2.2, a particular branch of DL-based activity recognition approaches focused on observing and exploring spatial attention through RGB images using deep neural networks to recognize activities based on visual information [195, 197]. However, the learned spatial attention is not entirely confident since it is learned in an unsupervised manner while training a supervised egocentric activity recognition (EAR) model. This has led some researchers to supervise spatial attention learning by using Gaze information [121] or by manually annotating the data [113] which is more expensive. In all cases, this has confirmed that the visual points of interest are concentrated around the hands and manipulated objects in first-person hand activity recognition problems. This relevant information can be used to design more robust EAR algorithms.

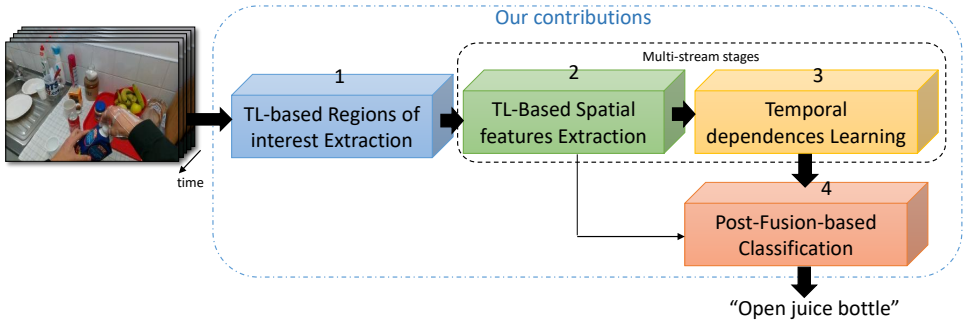

Fig. 3.8 Our proposed learning pipeline for RGB-based first-person hand activity recognition. The regions of interest are extracted for a given RGB image activity sequence using a pre-trained neural network in the first stage. Then, high-level spatial features are extracted in the second stage using a pre-trained deep neural network. Sequentially, in the third stage, the temporal dependencies are learned. In the last stage, a hand activity sequence classifier is learned using a post-fusion strategy applied to the previously learned temporal dependencies.

Motivated by all these observations, we introduce in this section a new learning pipeline for RGB-based first-person hand activity recognition that aims at overcoming the data scarcity problem while ensuring a low-cost good and accurate recognition, overcoming the limitations

of our previously introduced method (section 3.2). It is a novel four-stage learning pipeline, such as each stage is described as follows:

1. **Regions of Interest Extraction (RoIE)**. Unlike existing methods that use DL-based visual attention and require a large amount of data, we propose directly using the right and left hands as pertinent regions of interest that give information about manipulated objects and actions being performed. These regions of interest are extracted using a transfer learning technique. Our experiments showed that this information is the key to first-person hand activity recognition. In order to robustify the recognition model, we propose a data augmentation process, which is specifically adapted to these regions of interest.

2. **Spatial Features Extraction (SFE).** Here, we also use transfer learning instead of end-to-end DL methods. This stage exploits the visual information of the regions of interest from the previous stage. Adapting transfer learning for RoIE and SFE allows learning with a limited number of training samples while providing a good accuracy score. Furthermore, it decreases the training cost since the transferred neural network is already pre-trained.

3. **Temporal Dependencies Learning (TDL).** For each extracted deep visual descriptor (right and left) resulting from the previous stage, we learn the temporal dependencies in a multi-stream manner, which also avoids the over-fitting problem.

4. **Post-Fusion classifier (PFC).** In this final stage, we use the post-fusion strategy proposed in subsection 3.2.3 to learn classifying activities.

In the remainder of this section, we explain our proposed pipeline following the figure 3.8. In the first stage, we extract the regions of interest (subsection 3.3.1). Then, in the second stage, we extract the spatial features (subsection 3.3.2). In the third stage, we learn the temporal dependencies (subsection 3.3.3). Once the temporal learning is finished, in the last stage we transfer and use the knowledge from the previous stage to learn the classification of activities (subsection 3.3.4). In subsection 3.3.5 we introduce the combination of RGB-based and 3D skeleton-based solutions for hand activity recognition. Subsection 3.3.6 presents our experiments on the proposed approach. It also contains the discussion and comparison of the obtained results. Subsection 3.3.7 concludes the section.

3.3.1 Transfer Learning-based Regions of Interest Extraction (RoIE) and Data Augmentation

We use as unique input a sequence of images (frames) representing a first-person hand activity that we denote by $x(t) = \{I_1, I_2, .., I_T\}$, where I_t is an image frame at time-step t and T the sequence max length.

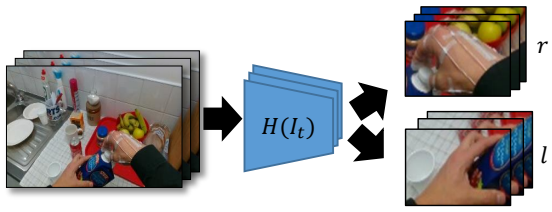

Fig. 3.9 The first stage of the pipeline is Transfer Learning-based Regions of interest Extraction (RoIE). Each image frame I_t is fed into a pre-trained neural network $H(I_t)$ resulting in two hand region sequences l and r that refer to the left and the right-hand regions respectively.

The main focus of the first-person hand activity is centered around the hands and manipulated objects. To this end, we propose to directly extract and use the left and the right-hand regions as regions of interest. Let denoting $H(I_t) = \{h_t^l, h_t^r\}$ where $H(.)$ is the pre-trained neural network that takes an image frame I_t as an input and outputs two sub-images h_t^l and h_t^r that refers to the left and the right hand respectively. So, by applying this to all image frames, the activity sequence will be reformulated by two sequences l and r that belong to the left and right hand, respectively, such as:

$$left = \{h_t^l\}_{t=1:T} \text{ and } right = \{h_t^r\}_{t=1:T} \tag{3.15}$$

Figure 3.9 illustrates the hand region extraction process. The proposed regions of interest characterize the hand activity sequence in a relevant way since the hand's visual information contains information about the type of grasp and the shape of objects being manipulated (noun), e.g., *"Juice bottle."* Moreover, passing this information through the time dimension allows retrieving relevant information about the performed action (verb), e.g., *"Open."* In subsection 3.3.6, we quantitatively show the efficiency of the proposed regions of interest. On the other hand, unlike visual attention methods based on end-to-end neural network [195], using transfer learning to extract the regions of interest helps to avoid the over-fitting problem and allows training with a limited number of samples while ensuring a good accuracy score. In subsection 3.3.6, we give details about the adopted pre-trained neural network.

In daily/industrial hand activities, one of the two hands, left or right, can be dominant. It depends on whether the participant is right- or left-handed. This may cause an imbalanced training dataset and make the model less generalizable. To this end, we proposed an adapted data augmentation process to balance the training dataset. It is applied to the RoIE stage's outputs. If only one hand is detected, e.g., the left hand, we augment the extracted sub-image of the right-hand h_t^r with the mirror effect of the detected left-hand h_t^l. The figure 3.10 illustrates the data augmentation process.

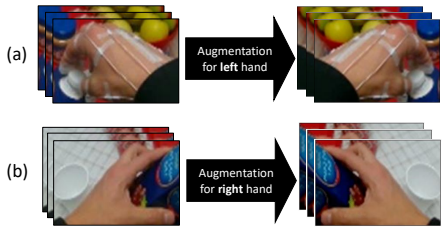

Fig. 3.10 Illustration of our data augmentation process. (a) the mirror effect of extracted right-hand sub-images h_t^r is used as augmentation for those of the left hand. (b) the mirror effect of extracted left-hand sub-images h_t^l is used as augmentation for those of the right hand.

In subsection 3.3.6, we quantitatively show this proposed data augmentation process's effectiveness.

3.3.2 Transfer Learning-based Spatial Features Extraction (SFE)

One of the problems where deep learning excels is image classification [231]. Image classification aims to classify a specific picture according to a set of possible categories by deeply exploring and learning spatial information. This motivated us to use a pre-trained neural network classifier to extract learned spatial features from the sub-images Eq.3.15 resulted from the previous stage.

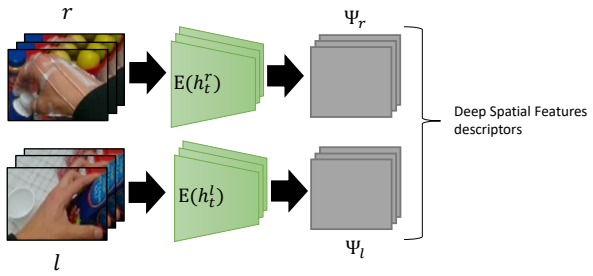

Fig. 3.11 The second stage of the pipeline: Transfer Learning-based Spatial features Extraction (SFE). Each extracted sub-image $h_t^l \in l$ and $h_t^r \in r$ is fed into a pre-trained neural network $E(.)$. This stage results in two deep spatial feature descriptor sequences Ψ_l and Ψ_r for the right and left hand, respectively.

We denote by $E(.)$ this pre-trained neural network. And we formulate the spatial feature descriptor sequences by Ψ_l and Ψ_r referring to the left and the right hands' regions as follows:

$$\Psi_l = \{E(h_t^l)\}_{t=1:T} \text{ and } \Psi_r = \{E(h_t^r)\}_{t=1:T} \tag{3.16}$$

This stage allows to exploiting the hand visual information resulting from the previous stage. Using a sophisticated pre-trained neural network reduces the dimension while keeping pertinent high-level spatial features. Adding to that, all transfer learning benefits, it decreases the learning cost and avoids the over-fitting problem while learning on a limited number of training samples. In subsection 3.3.6, we give details about the adopted pre-trained neural network.

3.3.3 Temporal Dependencies Learning (TDL)

As we do not have a learned neural network for this specific task, we train LSTM-based neural networks for their great success and capabilities to learn these long/short-term dependencies, as we have already mentioned in subsection 3.2.2.

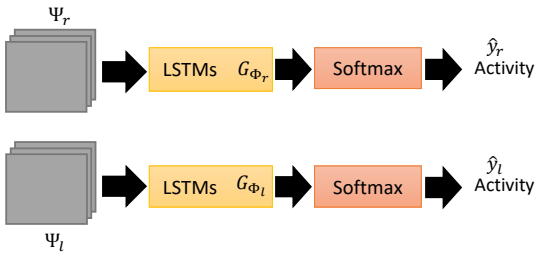

Fig. 3.12 The third stage of the pipeline: Temporal dependencies Learning. For each feature descriptor sequence Ψ_l and Ψ_r, a neural network composed of stacked LSTM layers followed by a softmax layer is trained independently to learn temporal dependencies by classifying activities.

Similarly, as we did in the previous approach (section 3.2), for each spatial feature descriptor Ψ_l and Ψ_r (seen in subsection 3.3.2), we train separately a simple neural network that consists of staked LSTM layers followed by a softmax layer to classify activities. Therefore, we train two neural networks separately, as shown in figure 3.12.

More formally, for each descriptor sequence Ψ_l and Ψ_r, we model the temporal dependencies with a composite function $G_{\Psi_l}(\Psi_l)$ and $G_{\Psi_r}(\Psi_r)$ respectively, where $G_{\phi_\cdot}(\cdot)$ is a LSTM network with Ψ_l and Ψ_r learnable parameters, while the output of $G_{\phi_\cdot}(\cdot)$ refers to the last hidden state of the last LSTM unit. For each network we define a cross entropy loss functions $\mathscr{L}_l$ and $\mathscr{L}_r$ as follows:

$$\mathscr{L}_l = -\sum_{j=1}^{C} y^j log(\hat{y}_l^j) \text{ and } \mathscr{L}_r = -\sum_{j=1}^{C} y^j log(\hat{y}_r^j) \tag{3.17}$$

where C is the number of classes and y the target label. The $\hat{y}_l$ and $\hat{y}_r$ are the softmax outputs that refers to the predicted label using left and right hand descriptor sequence respectively. The temporal learning parameters are optimized by minimizing over a labeled dataset:

$$\phi_l^* = \arg\min_{\Psi_l} \mathscr{L}_l(y, \hat{y}_l) \text{ and } \phi_r^* = \arg\min_{\Psi_r} \mathscr{L}_r(y, \hat{y}_r) \tag{3.18}$$

At the end of the pre-training, as a result, we have a set of two trained stacked LSTM layers, with optimised parameters ϕ_l^* and ϕ_r^*:

$$G_{\phi_l^*}(\Psi_l), G_{\phi_r^*}(\Psi_r) \tag{3.19}$$

We note that the purpose of this third stage is to learn the temporal dependencies, and all the classification results $\hat{y}_l$ and $\hat{y}_r$ are ignored. Only the results shown in Eq.3.19 are needed for the next stage.

3.3.4 Post-fusion-based Classification (PFC)

Once the temporal dependencies are learned, we proceed to the final classification similarly as we did in the previous approach (section 3.2). We train another multi-input neural network that exploits the resulted two pre-trained stacked LSTM layers introduced in (subsection 3.3.3) that we transfer with a fixed optimized parameters ϕ_l^* and ϕ_r^* as illustrated in figure 3.13.

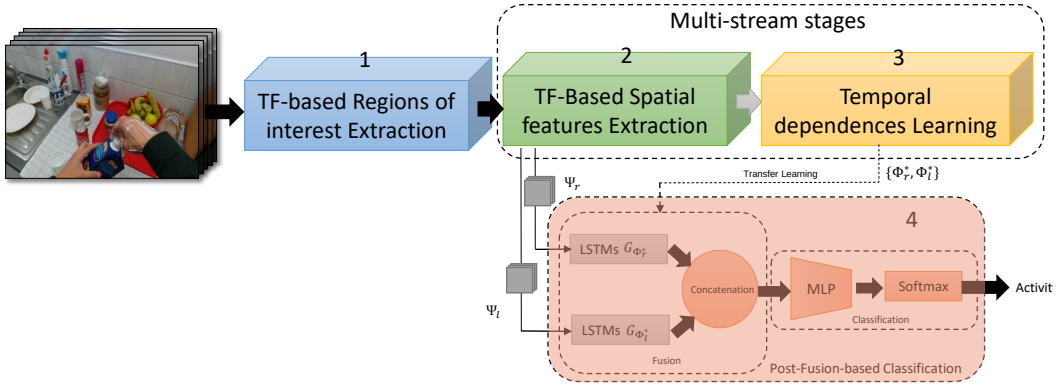

Fig. 3.13 The fourth stage of the pipeline: post-fusion-based Classification. Once the temporal dependencies are learned in the third stage. The LSTM layers are transferred to the fourth stage with fixed parameters Ψ_l^* and Ψ_r^*. Their outputs are concatenated and fed into a MLP+softmax for the final classification.

The two parallel output branches of the transferred LSTMs are concatenated, then fed into a Multi Layers Perceptron (MLP) that consists of two Fully Connected (FC) layers, followed by a softmax layer (figure 3.13). We model this network as shown in Eq.3.20, where F_γ is a MLP+softmax with learnable parameters γ, and *Concat* is the concatenation function:

$$F_\gamma(Concat(\{G_{\phi_l^*}, G_{\phi_r^*}\})) \tag{3.20}$$

The learnable parameters γ are optimized using the same loss function as in the previous stage by minimizing over the same training dataset.

This post-fusion strategy aims at ensuring a good accuracy score by tuning between the pre-trained LSTMs outputs.

3.3.5 Multi-modal RGB and 3D hand skeleton for first-person hand activity recognition

Both the RGB images and 3D hand skeleton-based methods have limitations and advantages. One of the 3D hand skeleton-based method's inter-class confusion occurs when different classes use objects of the same shape but different colors, e.g., *"open juice bottle"* and *"open milk bottle."* The exploitation of RGB images can surpass this limitation since they provide color information for manipulated objects. On the other hand, an example of inter-class confusion that can occur while using RGB-based methods is when the handled object in different classes has the same color. Still, the shape is different. The 3D hand skeleton data usage can help overcome this limitation since the handled object's shape can be driven from the 3D hand skeleton coordinates.

These observations motivated us to use the two modalities, RGB images and 3D hand skeleton data, for hand activity recognition. This is done by combining our two previously introduced methods. The spatial features extraction and the temporal dependencies learning steps are kept unchanged for both methods. Only the last step of post-fusion classification is changing, as shown in figure 3.14:

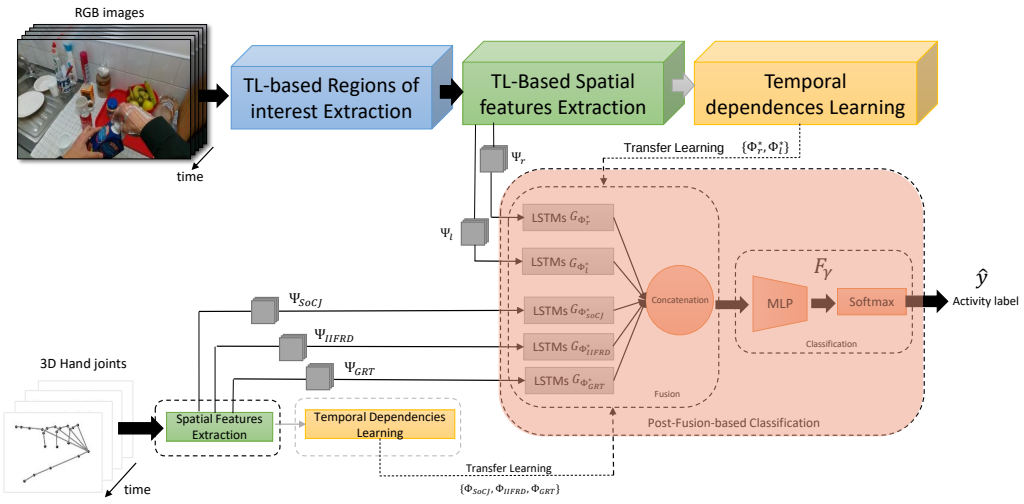

Fig. 3.14 The combination of our proposed RGB-based and 3D skeleton-based methods for hand activity recognition.

In the following subsection, we quantitatively show the improvement of this combination.

3.3.6 Experiments

Datasets

Several large-scale datasets have been proposed for EAR, e.g. EGTEA [184] and CharadesEgo [45]. In this work, we try to solve a sub-problem of the EAR, namely first-person hand activity recognition, while activities are supposed to be performed with the hands, which is not the case for some activity categories of these datasets. With this in mind, to validate our approach, we used the previously introduced real-world datasets:

- **FPHA dataset [48].** It represents a real challenge for activity recognition algorithms. For all the experiments, we used the setting proposed in [48], with exactly the same distribution of data: 600 activity sequences for training and 575 for testing.
- **EgoHand dataset [9].** We chose this dataset to evaluate our method in case there is not enough training data. We used the setting proposed by [9] that randomly splits these videos into 36 samples for training, 4 for validation and 8 for the test.

Implementation details

- **Spatial features extraction.** We deliberately chose VGG16 [106] for its widespread use as a standard foundation for transfer learning [202] and domain adaptation [27]. It is a powerful convolutional neural network, mainly designed for large-scale image recognition. The VGG16 model contains a stack of convolutional layers that capture basic features like spots, boundaries, and color patterns, followed by three fully-connected layers (FCL) that provide complex higher-level feature patterns. To this end, we extracted features from the last FCL, which provides an output vector of dimension 1x4096. VGG16 has shown good results. However, it is highly computational due to its complex architecture and a large number of parameters. Moreover, the size of its last FCL output is very large, and multiplying this size by the length of the activity sequence results in a large input dimension (200x4096) for the LSTM network. This requires high computing resources and time for the training process. Indeed, we experimented with a lighter pre-trained model, namely MobileNetV2, based on an inverted residual structure [171]. Table 3.6 shows the comparison between VGG16 and MobileNetV2. By using MobileNetV2, the accuracy dropped by 1.5%, but we achieved a gain in inference/training time and computational resources. The two models VGG16 and MobileNetV2 are pre-trained for image classification tasks on the ImageNet dataset [164] achieving 92.7% and 90% accuracy respectively. Keras framework is used for the implementation.

- **Temporal dependencies learning.** For each spatial descriptor sequence that refers to the right and the left hands, we trained different configurations of separated neural networks that consist of 1, 2, 3, and 4 staked LSTM layers followed by a softmax. We selected the best configuration with the best accuracy score: 2 staked LSTM layers of 100 units. We set the dropout probability to 0.5 (outside and inside the LSTM gates). We used Adam with a learning rate of 0.003 for the optimization. All the networks are trained with a batch size of 64 for 400 epochs. We also padded all sequence lengths to 200 and 100 time-steps per sequence for the FPHA and EgoHand datasets, respectively.

- **Post-fusion-based classification.** Once all the temporal dependencies are learned (end of stage 3), we recover the pre-trained LSTM networks in the PFC stage, fix all their weights, and discard Softmax layers. Then, the two outputs branches from the two parallel transferred LSTMs are concatenated and followed by an MLP that consists of two dense layers of 256 and 128 neurons, respectively, equipped with a *relu* activation function. At the end of the network, a Softmax layer is used for the final classification. This network is trained until 100 epochs, with the same batch size and optimization parameters as the previous networks. The implementation is based on the Keras framework.

Table 3.6 Performance comparison of our method on FPHA dataset using two different pre-trained neural networks for spatial features extraction, namely VGG16 and MobileNetV2.

Model	Inference time (ms)	Parameters (millions)	Last FCL size	Acc.(%)
VGG16	5.17	138	1x4069	96.52
MobileNetV2	3.34	3.5	1x1028	95.01

State-of-the-art comparison

Table 3.7 shows the accuracy of our approach compared with state-of-the-art methods on the FPHA dataset. The best performing approach among state-of-the-art methods is Tear [100], a transformer-based that consists of two modules, an inter-frame attention encoder and mutual-intentional fusion block. By exploiting RGB and depth modalities, they achieved 97.04% of accuracy, which is equivalent to our achievement (97.91%) while using the RGB modality only. The 3D skeleton-based approach that we proposed in the previous section gives good results, but they used the ground truth of 3D hand joints, which is not always available. This may conclude that RGB image sequences can provide the necessary elements to recognize hand activities.

Table 3.7 Activity recognition accuracy comparison of our proposed approach and the state-of-the-art on FPHA dataset. Our method outperforms all RGB-based methods, including end-to-end visual attention methods.

Methods	**Year**	**Modality**	**Accuracy(%)**
Two stream-color [46]	2016	RGB	61.56
H+O [205]	2019	RGB	82.26
Rastgoo et al. [155]	2020	RGB	91.12
Trear [100]	2021	RGB	94.96
HON4D [133]	2013	Depth	59.83
HOG2-depth [132]	2014	Depth	70.61
Novel View [151]	2016	Depth	69.21
Trear [100]	2021	Depth	92.17
Lie Group [213]	2014	3D Pose	82.69
Gram Matrix [249]	2016	3D Pose	85.39
TF [48]	2017	3D Pose	80.69
Nguyen et al. [126]	2019	3D Pose	93.22
Our	-	3D Pose	**96.17**
HOG2-depth+pose [132]	2014	Depth+3D Pose	66.78
JOULE-all [44]	2015	RGB+Depth+3D Pose	78.78
Tear [100]	2021	RGB+Depth	**97.04**
Our	-	RGB	**97.91**

Table 3.8 shows the accuracy of our approach compared to state-of-the-art methods on the EgoHand dataset. The proposed work by [80] and [9] focused on hand segmentation from an egocentric viewpoint. Nevertheless, they used the estimated and ground-truth hand masks to recognize activities. We outperformed their results by more than 5% accuracy, confirming the effectiveness of the proposed regions of interest over the hand mask. Since the EgoHand contains only 48 samples, this can also prove the ability of our method to learn on a limited amount of data.

Table 3.8 Activity recognition accuracy results in the EgoHand dataset that contains only 48 samples. Results show that our method performs better on a limited amount of data.

Method	**Acc (%)**
Khan et al. [80] + Ground truth hand mask	71.1
Khan et al. [80]	68.4
Bambach et al. [9] + Ground truth hand mask	**92.9**
Bambach et al. [9]	73.4
Babu et al. [6]	89.0
Our	**98.79**

Figure 8 shows the confusion matrix while testing our method on FPHA dataset [48]. By achieving 97.91% of activity recognition accuracy, we overcome the commonly confused classes *"open wallet"* and *"use calculator"* [48]. Nevertheless, we still get confusion between *"open wallet"* and *"receive coin"* due to the high appearance similarity between the two activities.

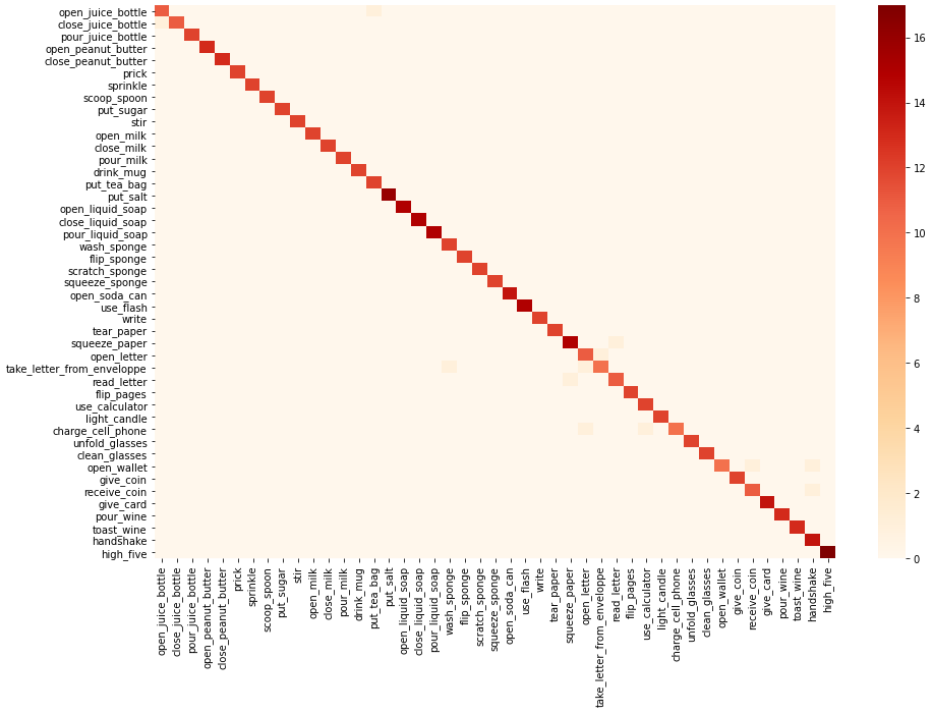

Fig. 3.15 Confusion matrix while evaluating our method on FPHA dataset testing samples.

Contribution of proposed regions of interest

To better show the contribution of left and right hands regions of interest, we skipped the RoIE stage. Instead, we used full-image frames. As expected, results presented in table 3.9 show that without our regions of interest, the accuracy dropped by more than 14%, which confirms RoIE effectiveness. Moreover, we overcome most state-of-the-art methods by using only the right hand as the region of interest.

Table 3.9 Activity recognition accuracy results on FPHA dataset with and without using our proposed regions of interest. Results show the significant impact of these regions of interest.

Extracted region of interest	**Acc.(%)**
Full image	82.01
Left hand bounding box	85.00
Right hand bounding box	**91.82**
Left+Right hands bounding boxes	**96.52**

Highly relevant information related to manipulated objects (nouns), e.g., *"juice bottle,"* can be derived from the visual data of the hand boxes, such as grasp type and object shape. Furthermore, by learning the temporal dependencies through this information, we can also relevantly characterize the actions (verbs), e.g., *"open."* For more ablation studies, we experimented our method on object and action recognition separately. Table 3.10 shows that our proposed method gives a good object and action recognition score by achieving 97.56% and 94.26% of accuracy, respectively.

Table 3.10 Object (noun) and Action (verb) recognition accuracy on FPHA dataset using our proposed pipeline. The accuracy results show that the proposed regions of interest allow object and action recognition which facilitates hand activity recognition.

Task	**Number of classes**	**Region of interest**	**Acc(%)**
Objects (nouns)	27	Left hand	88.69
		Right hand	95.82
		Left+Right hands	**97.56**
Actions (verbs)	27	Left hand	85.56
		Right hand	92.17
		Left+Right hands	**94.26**

Data augmentation

The results in Table 3.11 show that the accuracy is increased by 1.39% when we used our adapted data augmentation process. Furthermore, using only the right-hand regions of interest, we outperform most state-of-the-art methods by achieving 94.26% of accuracy.

Table 3.11 Activity recognition accuracy results in the FPHA dataset. (*) without data augmentation, (**) using data augmentation.

Extracted region of interest	**Acc(*) (%)**	**Acc(**) (%)**
Left hand bounding box	85.00	88.00
Right hand bounding box	91.82	94.26
Left+Right hands bounding boxes	96.52	**97.91**

RGB images Vs. 3D hand skeleton data

To experiment with the combination of our proposed RGB-based and 3D hand skeleton-based methods on first-person hand activity recognition, we used the same implementations settings previously introduced in subsections 3.2.4 and 3.3.6.

Table 3.12 shows the results for hand activity recognition trained and tested on the FPHA dataset. As expected, the combination of the two methods provides the best results by achieving 98.10% of accuracy. This can be explained by the fact that both modalities complement each other.

Table 3.12 Activity recognition accuracy results in the FPHA dataset. The combination of the proposed RGB-based and 3D hand skeleton-based methods achieves the best results.

Method	**Accuracy (%)**
3D hand skeleton based method	96.17
RGB images based mehtod	97.91
3D hand skeleton based + RGB images based method	**98.10**

3.3.7 Conclusion

In this section, a novel learning pipeline for first-person hand activity recognition has been introduced. The proposed pipeline is composed of four stages. In the first stage, we presented our Transfer Learning-based regions of interest extraction, the left and right hands regions, which have proven effective. The second stage is the Transfer Learning-based deep spatial feature extraction method that exploits the regions of interest in visual information. To manage the temporal dimension, in the third stage, we trained the temporal neural network in a multi-stream manner. Then, in the last stage, we applied a post-fusion strategy to classify

activities. The pipeline was evaluated on two real-world datasets and showed good accuracy results.

We also presented an experiment combining RGB images and 3D skeleton data based on the concept of our proposed methods introduced in this section and section 3.2. The combination achieved the best results, avoiding the ambiguous case of high intra-class dissimilarity, which occurs when manipulated objects in the same activity class may have different shapes, grip types, and colors.

3.4 Conclusion

We presented in this chapter our two solutions for hand activity recognition from the first-person viewpoint that meet our AR use case context and address its challenges. First, we detailed our proposed method for 3D skeleton-based hand activity recognition. Then, we introduced our proposed method for RGB-based and the combination of RGB-based and 3D skeleton-based solutions for hand activity recognition. We gave extensive experiments that evaluated and validated each proposed approach by discussing and comparing the obtained results with existing state-of-the-art methods. The experiments have proven the effectiveness of our proposed methods on real-world datasets. They also showed the advantages and disadvantages of the RGB images and 3D hand skeleton modalities on hand activity recognition. We also experimented with combining the two modalities for hand activity recognition, which significantly improved the recognition accuracy.

The introduced methods allow the recognition of hand activities, which are expected to be from known classes, seen during the training procedure. However, one of our desired activity recognition framework's main components is to detect and identify unknown activities unseen by the recognition model during its training procedure. Thus, in the next chapter, we present the remaining components of our targeted framework: unknown hand activity detection, unlabeled hand activity clustering, and incremental hand activity recognition.

Chapter 4

Continual Learning for Hand Activity Recognition

Contents

4.1 Introduction

In the previous chapter, we presented our proposed methods for hand activity recognition. The introduced methods are based on the classical closed-set recognition paradigm, where train and test activity samples are supposed to be from known classes. However, one of the components of our desired user activity recognition framework is to perform recognition in an open-set setting. Thus, it recognizes activities from known classes while, at the same time, detecting and rejecting unknown activities from unknown classes previously unseen during the training procedure. Such a capability allows guiding and feedbacks AR users, e.g., to warn of bad maneuvers in the case of assembling assistance usage.

Moreover, another component of our framework is to collect the detected unknown activities, automatically grouping and annotating them. Thus, the framework exploits these new annotated hand activity samples to incrementally learn and extend the multi-class classifier, making each new class "known" to the models. These capabilities allow overcoming the data sacristy problem and make the user activity recognition framework scalable and easily adaptable.

This chapter presents our originally proposed and adopted approaches to address the above-mentioned issues. We organize the remainder of the chapter as follows:

- **Section 4.2.** In this section, we introduce our adopted consensus-based open-set recognition that groups three approaches to deciding whether a test activity sample is from a known or unknown class. Then, we present and discuss the results of our experiments performed on real-world hand activity datasets.

- **Section 4.3. This section presents our main contributions in this thesis**. We introduce our proposed method for automatic grouping and annotating hand activity samples. From a general perspective, first, we pre-train a recognition model on labeled hand activity samples. Then, we use the pre-trained model feature space as a low-dimensional mapping manifold to categorize unlabeled activity samples based on classical clustering methods. To reach this goal, learning a highly discriminative feature space is crucial to facilitate the clustering task. Thus, in this section, we propose a new original loss function that encourages neural networks to learn a mapping function that maps samples into highly discriminative feature space. Finally, we show the benefit of the proposed approaches by presenting and discussing the experimental results.

- **Section 4.4.** Based on the state-of-art study presented in chapter 2, section 2.5, in this section, we present our adopted method for incremental hand activity recognition. Then, we present the experiments that evaluate and validate the adopted method. The

experiments also cover a comparison of our results with those obtained with an existing state-of-the-art method to enhance the validation.

4.2 Open-set Hand activity recognition

The previous chapter introduced our proposed methods for recognizing first-person hand activities. The recognized activity classes are known to the recognition model, which is trained and tested in closed-set settings. However, one of the main components of our desired activity recognition framework aims to detect unusual, unknown activities and recognize known ones. The recognition model must be learned and trained in open-set settings, a more realistic and challenging scenario. With the open-set assumption, incomplete knowledge of the world exists at training time, and unknown activities can be seen during testing. We refer the reader to chapter 2, section 2.3 for more details about the closed- and open-set recognition settings.

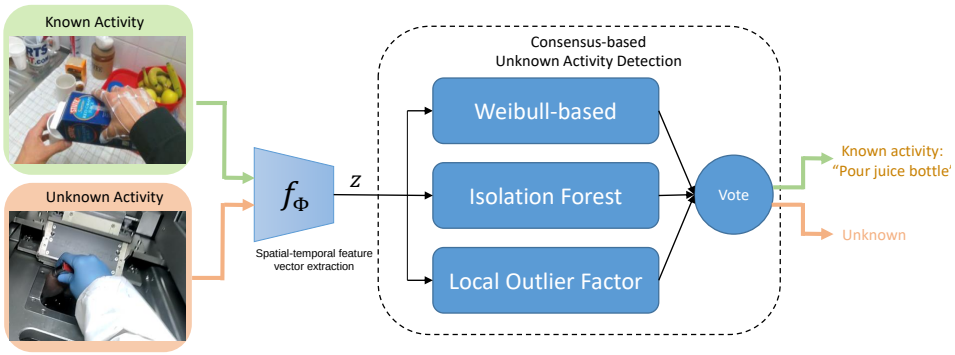

Fig. 4.1 Illustration of our adopted approach for unknown activity detection. For a given hand activity sample, first, we compute its associated spatial-temporal feature vector using a pre-trained hand activity recognition model. Then, based on the computed feature vector, we use our adopted consensus-based unknown activity detection to decide whether the sample is known or unknown.

Ensemble or consensus-based approaches have attracted the machine learning community in recent years. Using a consensus of classifiers is a machine learning approach in which classification decisions are merged to improve the overall classifier performance. The most straightforward approach to combining decisions is through voting or weighted voting. Thus, inspired by *Prakhya et al.* [149], in this section, we introduce our adopted consensus-based open-set hand activity recognition that groups three approaches to deciding whether a test activity sample is from a known or unknown class. To increase overall open-set recognition performance, we employ a consensus of three outlier detection approaches and aggregate their decisions via voting. Figure 4.1 illustrates the adopted approach.

The remainder of this section is organized as follows. In subsection 4.2.1, we formulate the problem of unknown activity detection. Then, following our adopted method, which consists of three outlier detection methods, subsections 4.2.2, 4.2.3, 4.2.4 and 4.2.5 present the probabilistic Weibull-based, the Isolation Forest (IF), the Local Outlier Factor (LOF) methods and their consensus-based decision, respectively. Subsection 4.2.6, presents our proposed method to perform open-set hand activity recognition. In subsection 4.2.7 we present the experiment and discuss the results. Subsection 4.2.8 concludes the section.

4.2.1 Adopted method formulation

Following the mathematical formulation in the previous chapters, we denote by $g_{w^*}(f_{\phi^*}(.))$ the pre-trained hand activity recognition network, where $g_{w^*}(.)$ is the final classification layer and $f_{\phi^*}(.)$ is the spatial-temporal feature learner backbone that we define as follows:

$$\begin{aligned} &f_{\phi^*} : \mathscr{X} \rightarrow \mathscr{Z} \\ &x \mapsto f_{\phi^*}(x) = z \end{aligned} \tag{4.1}$$

with w^* and θ^* are the optimised (learned) parameters. $\mathscr{X}$ is the space of all labeled samples and $x \in \mathscr{X}$ is a particular known (labeled) activity sample, and $z \in \mathscr{Z}$ is its associated feature vector in the feature space $\mathscr{Z}$.

Now, let us take $x^u \in \mathscr{X}^u$ as an unknown activity sample with $\mathscr{X}^u$ the space of unknown samples. The recognition neural network model $g_{w^*}(f_{\phi^*}(.))$ is learned in a closed-set paradigm, and its classifier layer $g_{w^*}(.)$ is not able to identify unknown samples $\{x^u\}$. Thus, the adopted method exploits the pre-trained network backbone f_{ϕ^*} to generate feature vectors $\{z^u\}$, which are used to identify $\{x^u\}$ as unknown samples.

As shown in figure 4.1, for a given hand activity sample, the adopted method computes its associated feature vector z^u using the pre-trained backbone f_{ϕ^*}. Then, a consensus decision is performed based on three unknown activity detection models.

4.2.2 Weibull-based model

The Weibull distribution [77] is a continuous probability distribution commonly used for reliability life data analysis. Weibull-based models in reliability theory clarify various observable component failures and phenomena. The formula for the two-parameter Weibull distribution, which is frequently used in failure analysis, is defined for a variable input A as follows:

$$\omega(A,\kappa,\lambda) = \frac{\kappa}{\lambda}\left(\frac{A}{\lambda}\right)^{\kappa-1} e^{-\left(\frac{A}{\lambda}\right)^{\kappa}} \tag{4.2}$$

Where κ is the shape parameter that determines the failure rate, and λ is the scale parameter of the distribution. Following [149], we use the two-parameter Weibull distribution in our work. Figure 4.2 presents the Weibull distribution for various values of λ and κ. We chose the Weibull distribution over all other distributions simply because it is best suited to identifying samples that do not belong to a specific class as reported in [149].

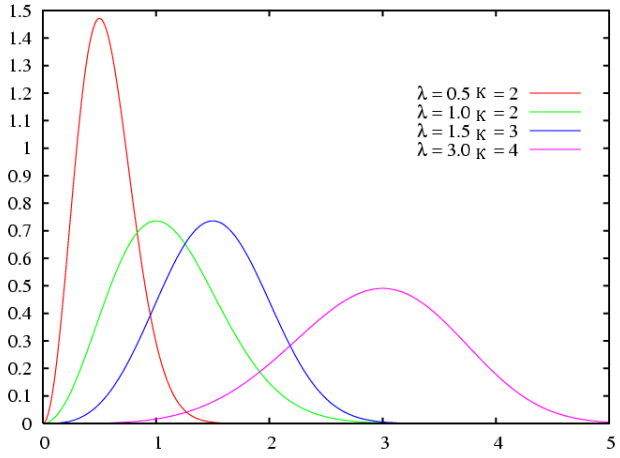

Fig. 4.2 Weibull distribution for various values of λ and κ.

From a general perspective, we compute the distance measure between each training sample feature vector and its class mean. The Weibull-based model is used to determine the probability for all distances acquired, which are then used to obtain the Weibull distribution. A sample close to a particular class has a small distance and, therefore, a high inclusion probability value. Similarly, a sample far from a particular class has a large distance and low inclusion probability value. These probabilities are used to determine whether or not a sample belongs to a known class. When a sample deviates from all classes means, it no longer belongs to these classes and behaves as an unknown.

We describe the steps of mean feature vector computation, the Weibull-based model fitting, and its usage to detect unknown samples as follows:

Computing mean feature vectors. We test the pre-trained recognition model $g_{w^*}(f_{\phi^*}(.))$ on all training samples. From this testing procedure, we select training activity samples that are correctly classified. Then, we map these samples into the feature space $\mathscr{Z}$ of the learned spatio-temporal backbone f_{ϕ^*}. As a result, we have feature vectors of **N** labeled and correctly

classified training activity samples $\{z_i\}_{i=1:\mathbf{N}}$. Finally for each training class, we compute the mean feature vectors z_j^{mean} for $j \in \{1,...,C\}$, with C the number of classes as follows:

$$z_j^{mean} = \frac{1}{\mathbf{n}_j} \sum_{i=1}^{\mathbf{n}_j} z_i^j \tag{4.3}$$

Where $\mathbf{n}_j$ is the number of correctly classified training samples in the j-th class. As a result, we have a set of mean feature vectors $\{z_j^{mean}\}_{j=1:C}$. Thus, each class is represented as a point, which is the mean feature vector z_j^{mean}, that is to say the mean computed over only the correctly classified training samples.

Fitting the Weibull model. To identify unknown samples based on feature vectors, as in [13, 176], we adapt the concept of Nearest Class Mean (NCM) [119] introduced previously in chapter 2, section 2.3.3. The NCM concept is applied per class within the feature vectors. Thus, we compute a Weibull model based on distances once the feature and the mean feature vectors are computed for each training class.

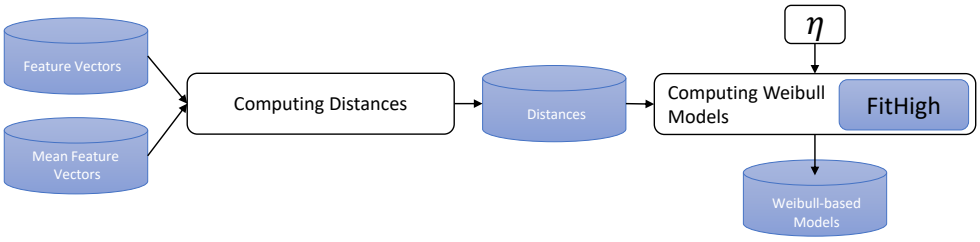

Fig. 4.3 Illustration of Weilbull-based model fitting.

As shown in figure 4.3, we first compute the distances between the feature vector and their associated mean feature vector that we denote by $dist_j$, and we define them as follows:

$$dist_j = \{\rho(z_j^{mean}, z_i^j)\}_{i=1:\mathbf{n}_j} \tag{4.4}$$

Where ρ is a distance measure. As a result of this step, we have a set of distances for each training class $\{dist_j\}_{j=1:C}$.

Then, in the next step, we use the libMR [22] *FitHigh* function to compute a Weibull model for each training class based on computed distances. The *FitHigh* function performs an optimization to find the optimal Weibull distribution parameters. We denote a Weibull-based model for the j-th training class by $Weibull_j$ and we define it as follows:

$$Weibull_j = (\kappa_j, \lambda_j) = FitHigh(dist_j, \eta) \tag{4.5}$$

where λ_j and κ_j are the shape and the scale parameters, which are used to estimate the probability inclusion of an activity input being a known with respect to class j-th. η is the tail parameter that we provide to the *FitHigh* function.

Unknown activity detection using the Weibull-based model. Given a new activity sample x^{new}, we first compute its feature vector $f_{\phi^*}(x^{new}) = z^{new}$. Then, we compute distances between this feature vector and all the C mean feature vectors $\{z_j^{mean}\}_{j=1:C}$. We denote these distances by $\{dist_j^{new}\}_{j=1:C}$ and we define it as follows:

$$dist_j^{new} = \rho(z_j^{mean}, z^{new}) \tag{4.6}$$

Then, we calculate the inclusion probability $P(c_j|x^{new})$ of the test simple x^{new} to each training class c_j using the previously computed Weibull parameters $\{Weibull_j\}_{j=1:C}$ as follows:

$$P(c_j|x^{new}) = 1 - \omega_j(dist_j^{new}) = 1 - \frac{\kappa}{\lambda}\left(\frac{dist_j^{new}}{\lambda}\right)^{k-1} e^{-\left(\frac{dist_j^{new}}{\lambda}\right)^{\kappa}} \tag{4.7}$$

We evaluate the maximum between all inclusions probabilities $\{P(c_j|x^{new})\}_{j=1:C}$. Then, we compare it with a fixed threshold ε:

$$\varepsilon \leqslant MAX(\{P(c_j|x^{new})\}_{j=1:C}) \tag{4.8}$$

If the maximum probability is greater or equal to ε, the sample is classified as belonging to a known class; otherwise, the sample is classified as belonging to an unknown class. We apply the above steps to each test sample. Then, we store the results for the final vote, section 4.2.5.

4.2.3 Isolation Forest (IF)

Unknown samples are often scarce and distinct from known ones. Thus, they are more sensitive to isolation from other samples. *Liu et al.* [104] demonstrated how a tree structure might be efficiently built to isolate every occurrence in a dataset. Outliers points are isolated closer to the tree's root due to their sensitivity to isolation, while regular points are isolated at the tree's deepest end.

The Isolation Forest builds a set of iTrees for a given data set to find outliers (in our case, unknown samples). It splits the data recursively at random partition points with randomly selected features. The heights of the outlier-containing branches are small compared to the heights of the other data points. As a consequence, the branch height is employed as an

outlier score. The final step averages the path lengths of the data points in the Isolation Forest's trees.

High-dimensional data is a real challenge for outlier detection. In a high-dimensional space, every point is equally sparse for distance-based approaches, potentially rendering distance worthless [3]. In contrast, the Isolation Forest method efficiently addresses this problem, which explains our choice of this method. In the following, we briefly introduce the Isolation Forest methodology.

As we did with the Weibull-based model, for each training known class c_j, we build an Isolation Forest model. The initial step is to create an isolation tree root node T that contains all $\mathbf{n}_j$ feature vectors of correctly classified samples in the class c_j. A candidate list Cl of nodes is formed as a single list that contains the root node. Then, until the candidate list Cl is empty, the following operations are repeated in order to construct the isolation tree T:

- Select a node Rl at random from Cl and delete it from the Cl list.
- Choose a random attribute i and divide the data in Rl into two sets, Rl_1 and Rl_2, at a random value v along that attribute. All Rl data points satisfy $z_i \leqslant v$, and all Rl_2 data points satisfy $z_i > v$. The value v is picked at random from the data points in node Rl that have the maximum and minimum values of the i-th attribute.
- The nodes Rl_1 and Rl_2 are children of Rl in T. If Rl_1 and Rl_2 contain more than one point, add it to Cl and repeat the previous step. Otherwise, assign the node as an Isolation tree leaf.

To determine an unknown sample score, the distance from the root to the leaf is averaged and normalized over all trees. A score of 1 indicates a known, while a number close to 0 indicates an unknown one. For more details about the Isolation Forest method, we refer the reader to the original paper [104].

Following the above steps, we fit the Isolation Forest model on the correctly classified training samples; then pass each test sample through the fitted model to determine whether it is known or unknown. The results are stored for the final vote.

4.2.4 Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) is a score generated using an unsupervised density-based approach proposed by *Breunig et al.* [17]. Mainly, it computes the local density deviation of a particular data point in a multidimensional dataset regarding its neighbors. It considers a sample as an outlier if it has a significantly lower density than its neighbors. For each

training class c_j, we build a LOF model using feature vectors of correctly classified training samples. We describe the steps of the LOF method as follows:

k-distance. For each feature vector z, we compute its k-th nearest neighbor, which we denote by $k\text{-}distance(z)$. The $k\text{-}distance(z)$ offers a measure of the density in the surroundings of the mapped sample z. The area surrounding z is dense when $k\text{-}distance(z)$ is small and sparse when it is large.

k-neighborhood. The mapped samples that are within $k\text{-}distance(z)$ of sample z are referred to as being in its $k\text{-}neighborhood(z)$ set. The $k\text{-}neighborhood(z)$ of z comprises all mapped samples with a distance smaller than the $k\text{-}distance(z)$. More precisely, we compute the greatest distance up to the k-th closest mapped sample. For simplicity, we denote the set of $k\text{-}neighborhood(z)$ as $\mathbf{N}_k(z)$.

Reachability distance. The introduced $k\text{-}distance(.)$ is used to define what is called reachability distance. In other words, the reachability distance of a vector z from another vector z_o is the real distance between the two vectors, but at the very least, the value of $k\text{-}distance(z_o)$. We denote the reachability distance by $reach\text{-}dist_k(.,.)$ and we define it as follows:

$$reach\text{-}dist_k(z,z_o) = max\{k\text{-}distance(z_o), \rho(z,z_o)\} \tag{4.9}$$

where $\rho(.,.)$ is problem-specific distance measure (e.g., the Euclidean distance).

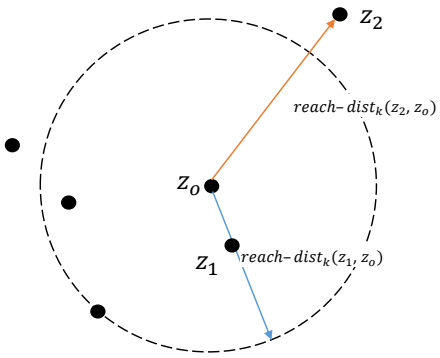

Fig. 4.4 Illustration of the reachability distance $reach\text{-}dist_k$ for $k = 4$. If the feature vector z is far away from z_o (sample z_2), the reachability distance between the two samples is just their actual distance $\rho(z_o,z_2)$. However, if they are near (sample z_1), the actual distance is replaced by the value of $k\text{-}distance(z_0)$. This is done to limit the statistical fluctuations of $\rho(z,z_o)$ for all z's near z_o.

Local reachability distance. We denote for a mapped sample z by $lrd_k(z)$. It is the inverse of the average reachability distance from the $\mathbf{N}_k(z)$:

$$lrd_k(z) = \left[\frac{\sum_{z_i \in \mathbf{N}_k(z)} reach\text{-}dist_k(z, z_i)}{|\mathbf{N}_k(z)|} \right]^{-1} \tag{4.10}$$

The local reachability density of the vector z is calculated by examining the *k-distance* of feature vectors in $\mathbf{N}_k(z)$. Thus, the LOF score can be computed based on the local reachability density.

Local Outlier Factor score. Finally, the Local Outlier Factor *LOF* is the ratio that indicates if a feature vector z of a particular activity sample is an outlier in respect to its surroundings. $LOF(z)$ is the average of the ratios of z's local reachability distance and the set $\mathbf{N}_k(z)$.

$$LOF_k(z) = \frac{\sum_{z_i \in \mathbf{N}_k(z)} \frac{lrd_k(z_i)}{lrd_k(z)}}{|\mathbf{N}_k(z)|} \tag{4.11}$$

If the LOF score for a particular feature vector is close to 1, it mean that the vector is in a relatively dense region with its neighbors, whereas if it is close to zero, the samples is considered as an outlier. For more details about the LOF method, we refer the reader to the original paper [17].

Following the above steps, we fit the LOF model on the correctly classified training samples; then pass each test sample through the fitted model to determine whether it is known or unknown (outlier). The results are stored for the final vote, section 4.2.5.

4.2.5 Consensus-based unknown activity detection

We utilize a voting mechanism to get the final decision after receiving unknown scores from all three introduced outlier detection methods.

4.2.6 Open-set hand activity recognition

We used the methods introduced above to detect and reject unknown activities. However, open-set recognition aims at rejecting unknown activities while classifying the known ones [52].

In our preliminary experiments, we used scores provided by the three previously introduced outlier detection methods to classify known samples. However, it turned out that these methods are very good at rejecting unknown samples, but they classify very poorly known ones into their associated classes. To this end, in contrast to *Prakhya et al.* [149], we propose to use the classification layer g_{w^*}, which is used to train the recognition model to classify known activities. Figure 4.5 illustrates our open-set hand activity recognition method.

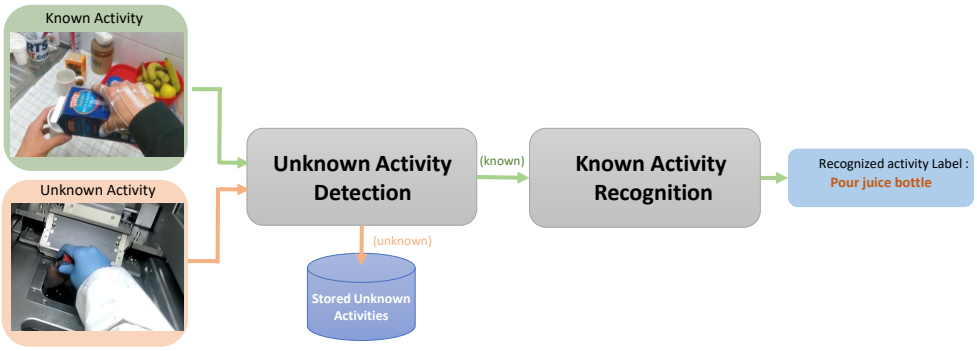

Fig. 4.5 Illustration of our open-set classification strategy. We use the unknown activity detection only for classifying known or unknown samples (not for activity recognition), and we use the recognition model classifier to determine its associated class if a test sample is identified as known.

4.2.7 Experiments

Dataset

To validate our adopted approach for unknown activity detection and open-set activity recognition, we used **FPHA dataset** [49]. As introduced in chapter 2 section 2.2.4, it is a publicly available dataset for first-person hand activity recognition, providing 3D hand joint and image sequences. It perfectly meets our evaluation and analysis needs since it contains 45 different activity categories distributed in three different scenarios: *"kitchen"* (25), *"office"* (12), and *"social"* (8), which can be partitioned as known and unknown classes. Table 4.1 shows the two data configurations that we have adopted for the experiments and which we denote by *Config1* and *Config2* respectively. We tested our approach on both RGB images and 3D hand skeleton sequences.

Table 4.1 The adopted data configurations based on the FPHA dataset [49].

	Config1			***Config2***		
	Scenarios	**Samples**	**Classes**	**Scenarios**	**Samples**	**Classes**
Known	kitchen+office	964 divided into: Train: 496 Test: 468	37	kitchen+social	843 divided into: Train: 431 Test: 412	32
Unknown	social	211	8	office	332	13

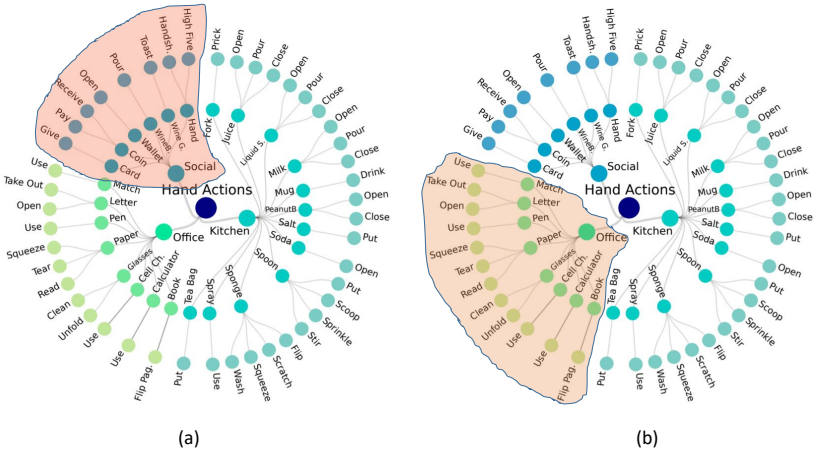

Fig. 4.6 The adopted FPHA dataset [49] partitioning. (a) Configuration 1: the colored region is the classes of the *"social"* scenario which are used as unknown classes, while these of *"kitchen"* and the *"office"* scenarios are used as known classes (b) Configuration 2: the colored region is the classes of *"office"* scenario which are used as unknown classes, while these of *"kitchen"* and the *"social"* scenarios are used as known classes.

We also tested our proposed approach on our **SLS France** dataset. We used all the classes of the FPHA dataset as known classes and all the SLS France dataset classes as unknown. We denote this third configuration by *Config3*. In this way, we test the approach on the generalization over datasets.

Evaluation protocol

The closed-set assessment evaluates a learned classifier using previously unseen samples from seen classes during training. The classes provided during testing are the same classes on which the model was trained, although with new samples. However, in open-set evaluation, the classifier is evaluated with samples from classes not unseen during training as well as samples from the seen classes. As a result of being used to make decisions on new unseen samples during the testing procedure, we claim the classifier might gain incomplete information during the training procedure.

To this end, we train the recognition model in a limited number of classes throughout the training procedure. All classes in the (*"kitchen"* + *"office"*) scenarios are used to train the model following *Config1*; and all classes in the (*"kitchen"*+*"social"*) scenarios are used to train the model following *Config2*. During the testing procedure, we generate a test set by combining test samples of known and unknown classes. Thus, we shuffle testing samples from (*"kitchen"* + *"office"*+*"social"*) scenarios for both configurations *Config1* and *Config2* (see table 4.1).

The classifier's performance is tested by correctly detecting samples from unknown classes and classifying known ones. Thus, we combine all unknown classes into a single undifferentiated unknown class and add them to known classes. Using the *openness* measure described in chapter 2 section 2.3.1, with the equation 2.2, we may estimate the classifier's open-set range. This measure only considers the number of additional classes available, not the open space itself [176]. The $openness = 5\%$ for *Config1*, 8.83% for *Config2* and 5% for *Config3* .

The accuracy measure, commonly employed in closed-set classification tasks, is extended to the open-set setting by merging all unknown classes into a single unified unknown class, as explained above.

Implementation details

- **Hand activity recognition model.** For the hand activity recognition neural network, we used the RGB- and 3D skeleton-based architectures that we proposed in chapter 3 section 3.3 and 3.2. We kept the same implementation details for the training and the testing procedures.

- **Weibull model.** We used feature vectors of correctly classified training samples to build a Weibull model for each training class, as described in section 4.2.2. We followed the suggestions provided in [13] to choose the η and the ε parameters. We tested with the Euclidean and the Cosine similarity distances for the distance measure ρ. We have excluded the choice of using the Mahalanobis distance, as the datasets we use do not have much data.

- **Isolation Forest.** We used feature vectors of correctly classified training samples to fit the Isolation Forest model described in section 4.2.3. We apply the Isolation Forest fitted model to test samples' feature vectors during testing. The Isolation Forest method returns min-max normalized scores. The lower the test sample's score, the more it is suspected to be unknown. We used the Isolation Forest implementation proposed in the Scikit-learn Framework [142]. We kept the default parameters, which follow the suggestions of the original paper [104].

- **Local Outlier Factor.** We used feature vectors of correctly classified training samples to fit the Local Outlier Factor model described in section 4.2.4. The model generates a learned boundary that delimits the distribution contour based on the training data. Any test sample that falls inside the learned boundary is presumed to be of the same class. Otherwise, it belongs to an unknown class if it sits outside the border. The LOF

model gives an outlier score for each test sample based on a density concept. We used the implementation proposed in the Scikit-learn Framework [142] with the default parameters.

- **OpenMax.** We implemented the *OpenMax* [13] open-set recognition method introduced in chapter 2, section 2.3.3 to enhance the state-of-the-art comparison. We kept the exact implementation details as in the original paper. For the number of "top" classes to revise based on the *Softmax* predictions, we tested with 5, 10, and 15, then we selected the one that gave the best result.

Results and discussion

We separately compared the three outlier detection methods and combined them with the *OpenMax*. Table 4.2 shows the unknown activity detection results on the three data configurations. The compared methods must detect all unknown samples of the unknown classes, samples of the *"social"* scenario for the *"Config1"*, samples the *"office"* scenario for the *"Config2"* and samples the *"kitchen"*+ *"social"*+*"office"* scenarios for the *"Config3"*.

The IF and LOF reject unknown samples efficiently, but they struggle to identify known samples, which decreases the overall performance. This is typical behavior that we expect from outlier detection methods. In contrast, the *OpenMax* and the Weibull-based method perform better against the specialized outliers detection methods IF and LOF. This is because our proposed Weibull-based and *OpenMax* decisions are based on probabilities, which has more tolerance in rejecting unknowns. Combining the Cosine similarity-based Weibull, the LOF, and IF methods provides good results. We also observe that the RGB-based method achieves the best results compared to the 3D hand skeleton-based method.

Table 4.2 Unknown hand activity detection results comparison on the FPHA [49] and SLS France datasets following our three data partitioning configurations. The results only concern unknown activity detection accuracy and not the open-set activity recognition. Our Consensus-based method performs better than the state-of-the-art *OpenMax* method.

		OpenMax	**LOF**	**IF**	**Weiblull (Euclidean)**	**Weiblull (Cosine)**	**LOF+IF+ Weibull (cosine)**
Config1	**RGB**	96.241	91.239	91.666	94.308	95.210	**96.682**
	3D	95.256	90.811	91.346	93.858	94.866	**95.734**
Config2	**RGB**	95.250	88.106	89.199	94.104	95.225	**96.084**
	3D	94.820	87.621	88.470	92.288	92.983	**95.632**
Config3	**RGB**	95.398	92.260	92.608	94.963	95.137	**97.481**

Table 4.3 shows the open-set activity recognition results on the three data partitioning configurations. Thus, the compared methods must recognize (classify) all the known samples associated to the known classes, samples of the *"kitchen"+"social"* scenarios for *"Config1"* , samples the *"kitchen"+"office"* scenarios for *"Config2"* and samples the *"kitchen"+"office"+"social"* scenarios for *"Config3"*. In the same time, they must also detect all the unknown samples of the unknown classes, samples of the *"social"* scenario for *"Config1"*, samples the *"office"* scenario for *"Config2"* and samples the SLS France dataset *"Config3"*.

We use our proposed method introduced previously in section 4.2.6 to assign samples identified as known to their associated classes. We can see that the results are proportional to those presented in the previous table since unknown sample detection plays an essential role in open-set recognition. Once again, by combining the Cosine similarity-based Weibull, the LOF, and the IF methods, we achieve the best results on the three configurations.

Table 4.3 Open-set hand activity recognition results comparison on the FPHA [49] and SLS France datasets following our three data partitioning configurations. Our Consensus-based method performs better than the state-of-the-art *OpenMax* method.

		OpenMax	**LOF**	**IF**	**Weiblull (Euclidean)**	**Weiblull (Cosine)**	**LOF+IF+ Weibull (cosine)**
Config1	**RGB**	92.693	87.125	87.859	90.637	91.385	**92.970**
	3D	89.980	84.420	85.878	89.432	90.346	**90.822**
Config2	**RGB**	91.223	85.632	86.910	90.021	90.266	**91.777**
	3D	89.890	83.075	84.775	88.756	88.169	**90.528**
Config3	**RGB**	85.440	81.146	81.648	83.192	84.104	**85.575**

4.2.8 Conclusion

In this section, we presented our adopted consensus-based open-set recognition component that gives the targeted framework the ability to perform open-set hand activity recognition. We used a consensus of three outlier detection approaches to increase open-set recognition performance: The Weibull model, the Isolation Forest, and the Local Outlier Factor method. We aggregated results from the three approaches to deciding whether the activity is from a known or unknown class via majority voting. If a test sample is identified as known, we propose to assign it to its associated known class using the final classification layer. The final layer trains the hand activity recognition model to recognize known activities. The adopted approach was evaluated on two real-world datasets and showed promising results.

In the next section, we introduce our proposed approaches to exploit these detected unknown hand activities by clustering and annotating them to be integrated into the recognition and detection models.

4.3 Unlabeled hand activity clustering

In the previous section, we introduced our adopted approach to detecting and identifying unknown activities. Once the number of detected novel activities reaches a specified number, they will be semi-automatically annotated (labeled). First, we categorize (cluster) them based on similarities; then, a human expert assigns a class label for each category. Therefore, in this section, we propose an approach for unlabeled hand activity categorization (clustering), which follows the Unsupervised Domain Adaptation UDA paradigm. First, we pre-train a supervised neural network on labeled samples from the source domain. Next, we try to solve the UDA by using the pre-trained neural network model feature space as a low-dimensional mapping manifold to categorize unlabeled target domain samples based on classical clustering methods.

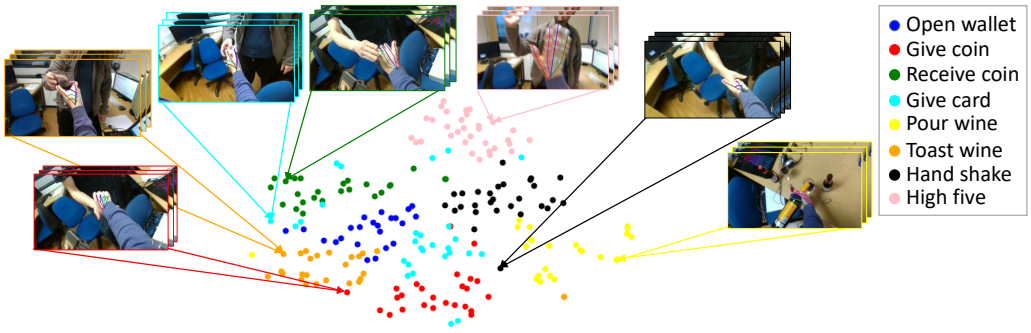

Fig. 4.7 2D features embedding of unlabeled hand activities using our proposed method on the ”social” scenario of the FPHA dataset [49].

Better usage of the prior knowledge from the source domain is crucial for UDA. Therefore, the pre-trained model, which relies on the source domain, must map hand activities to a highly discriminative features space (Figure 4.7). This requirement is highly discussed in the field of open-set face recognition [107, 217, 35]. Thus, we concluded that the desired feature space must satisfy two main objectives of metric learning: (1) maximizing inter-class distances and (2) minimizing intra-class distances for the mapped hand activities. This is one of the main issues we have addressed in this section. On the other hand, we tackled the problem of the high sensitivity of clustering algorithms regarding the feature space provided by the pre-trained model, which we noticed through our experiments.

As we mentioned in chapter 2, section 2.4.3, the development of Deep Learning (DL) allowed many improvements to deep metric learning (DML) methods. This is mainly by employing neural networks as low-dimensional mapping functions. The employed neural

networks are optimized with specialized loss functions to satisfy metric learning objectives on the high-level features space. In this regard, one of the first research directions was contrastive pair-based loss functions [56, 63, 189, 187, 220]. Recently, more practical penalty-based DML loss functions have been proposed [107, 217, 35], which try to map samples and impose discriminative constraints on a hyper-sphere manifold. This is generally done by manually applying penalties to the modified *Softmax* target logit (we refer the reader to chapter 2, section 2.4.3 for more details). However, these loss functions do not explicitly imply constraints to satisfy the first metric learning objective of maximizing the distances between different classes.

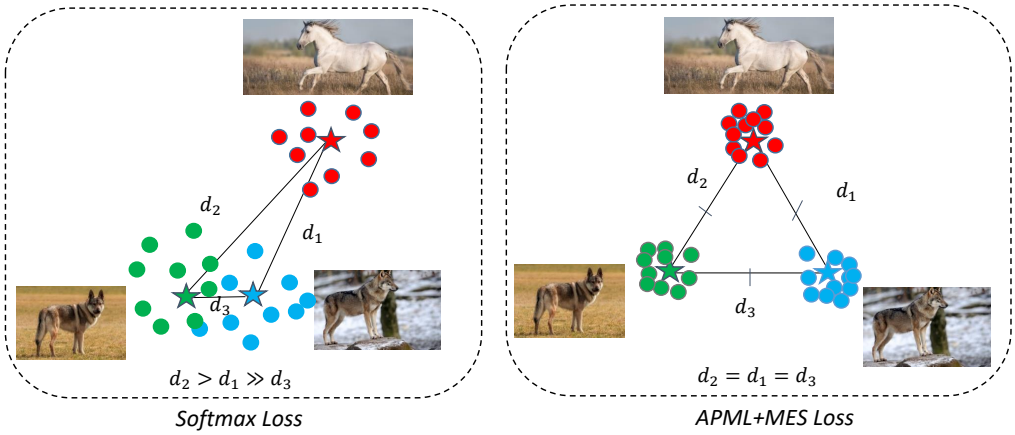

Fig. 4.8 Illustration of feature spaces provided while training a neural network using the *Softmax* and our *APML +MES* loss functions. The *APML* loss function will optimize the Euclidean distance feature vector and their associated weight vector. Thus, the embedded feature vectors will be attracted and move closer to their corresponding centers, enhancing intra-class compactness. As a complement, the *MES* loss function forces the network to ensure the best inter-class separation. The combination of *APML* and *MES* losses results in a highly discriminative feature space.

In this section we propose a novel composite DML loss function to address the above limitations, encouraging Deep Neural Network (DNN) to learn a mapping function that maps samples into a highly discriminative feature space (Figure 4.8). The proposed loss function comprises two complementary losses:

1. An Automated Penalty-based Metric Learning loss function that we denote by (*APML*), which learns similarity measures in the feature space to ensure the best intra-class distance minimization. Unlike state-of-the-art losses that manually imply a margin value to penalize the target logit, *APML* requires no additional hyperparameters.

2. Similarly to the state-of-the-art DML loss functions, APML separates different classes while minimizing intra-class distances. However, this is done in a non-explicit way without imposing any constraints. To this end, we proposed a complementary Mutually Equidistant Separation loss function denoted by (MES), which implies a heavy constraint seeking to achieve an optimal inter-class separation.

We summarize the contributions presented in this section as follows:

- To our humble knowledge, we are the first to propose a comprehensive and generic UDA approach applied to the categorization of unlabeled 3D skeleton-based hand activities. The proposed approach is experimented with and evaluated on a real-world dataset.
- A novel composite (*APML+MES*) loss function. It aims to learn a discriminative feature space while maintaining good recognition accuracy. The proposed loss is experimented and compared with four state-of-the-art losses on hand activity clustering and classification. It is also tested and compared with state-of-the-art methods on clustering and image-retrieval tasks on two commonly used real-world datasets.
- To solve the sensitivity problem of clustering algorithms, which to our knowledge is not addressed in the state-of-the-art, we proposed a Statistical and Consensus Clustering strategy that we denote by SCC.

The remainder of this section is organized as follows. First, in subsection 4.3.1, we give a general overview of the proposed method for unlabeled hand activities categorization. Subsection 4.3.2 presents our adopted discriminative feature space creation method based on a supervised learning paradigm. We present the clustering technique adopted to cluster the mapped unlabeled activity samples in subsection 4.3.3 . Then, in subsection 4.3.4, we introduce our SCC strategy that boosts the clustering performance. Subsection 4.3.5 is devoted to introducing our proposed composite deep metric learning (*APML+MES*) loss function. Then, we show the benefit of the proposed approaches by presenting and discussing the experimental results in subsections 4.3.6 and 4.3.7. Finally, subsection 4.3.8 concludes this section.

4.3.1 Proposed method for unlabeled activity categorization

Following the formulation we gave in chapter 2, section 2.4.1, and the previous section, we overview the method as follows: For a given set of unlabeled hand activity samples $\{x^u\}$ of a target domain, we exploit the knowledge from labeled samples $\{x, y\}$ of the source domain to

find the correspondent unknown set of labels $\mathscr{Y}_u = \{y^u\}$. Where x^u is a particular unlabeled sample and y^u is its correspondent label that we seek to find. As illustrated in Figure 4.9, the method proceeds in three main steps.

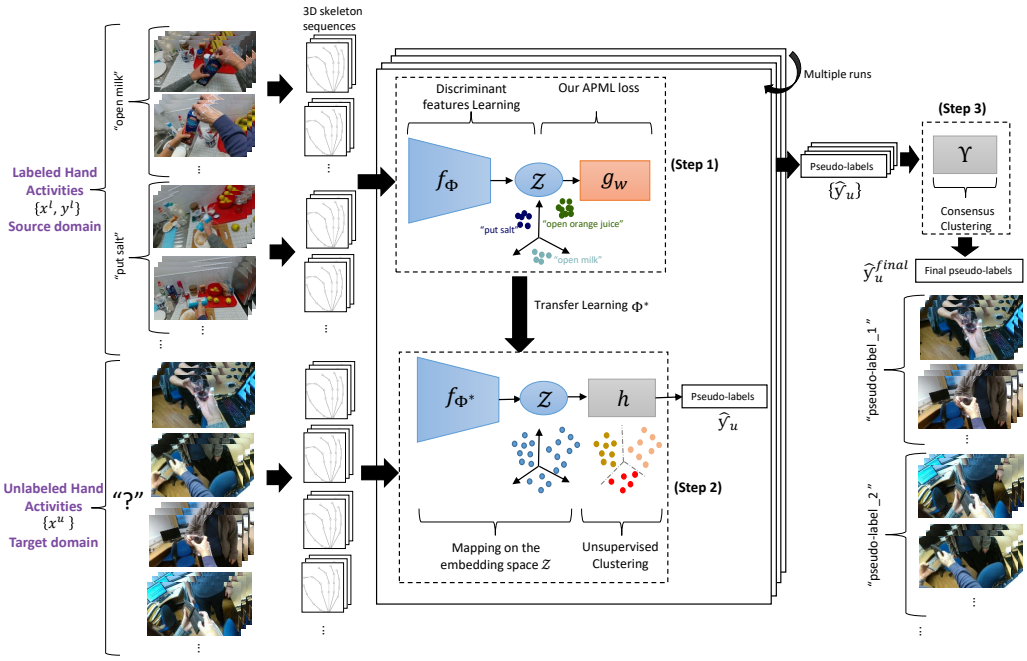

Fig. 4.9 The proposed method. We exploit the knowledge from labeled samples of the source domain $\{x, y\}$ to categorize the unlabeled ones of the target domain $\{x^u\}$. Thanks to our proposed APML+MES loss function, in step1, we learn a discriminative low-dimensional feature space that we use in step 2 to facilitate the clustering. Following our SCC strategy, we repeat steps 1 and 2 multiple times then we perform a consensus clustering on the resulting clustering predictions to generate the final clustering prediction.

4.3.2 Supervised learning for discriminative features space creation

First, we pre-train the hand activity recognition neural network $g_w(f_\phi(.))$ where $g_w(.)$ is the classification layer equipped with our proposed composite DML loss function (*APML+MES*) and $f_\phi(.)$ is the backbone Spatio-temporal feature learner, with w and ϕ learnable parameters. The training is performed on labeled samples of the source domain $\{x, y\}$ where y is the label of the particular sample x. This is done in a classical supervised manner as shown in Figure 4.9 (step 1). For hand activities categorization, we have adopted the neural network architectures that we proposed in the previous chapter as our backbone $f_\phi(.)$.

Thanks to our loss function, this first step results in a pre-trained model $f_{\phi^*}(.)$ with optimized parameters ϕ^*, which maps activities on a discriminative features space $\mathscr{Z} \subset \mathbb{R}_z^d$ where d_z is the embedding dimension.

4.3.3 Clustering of unlabeled activities

In this second step, we map the set of unlabeled samples $\{x^u\}$ into the $\mathscr{Z}$ space. This is done by passing these activities trough the pre-trained neural network $f_{\phi^*}(x^u) = z^u \in \mathscr{Z}$, which results in a set of embedded activities $\{z^u\}$. Next, we apply a classical clustering that we formulate as $h(\{z^u\}) = \hat{\mathscr{Y}}_u$ where $h(.)$ is the clustering function and $\hat{\mathscr{Y}}_u = \{\hat{y}^u\}$ are the predicted pseudo-labels for the unlabeled set of activities $\{x^u\}$. Figure 4.9 (step 2) gives an illustration for this step.

As we mentioned in the introduction, in our preliminary experiments, we observed a high sensitivity of clustering algorithms to the randomness present in the pre-training process of step 1. It is directly related to the random initialization of weights ϕ and w, the dropout layers, and the optimizer. To this end, we proposed a SCC strategy that consists in repeating steps 1 and 2 sequentially several times. It aims at stabilizing the mean and the standard variation [146], which results in a set of predicted pseudo-labels $\{\hat{\mathscr{Y}}_u\}$.

4.3.4 Consensus clustering

Motivated by observations that we have introduced in chapter 2, section 2.4.5, we apply a consensus clustering on the resulted set of predicted pseudo-labels $\Upsilon(\{\hat{\mathscr{Y}}_u\}) = \hat{\mathscr{Y}}_u^{final}$ where $\Upsilon(.)$ is the consensus clustering function and $\hat{\mathscr{Y}}_u^{final}$ is the final predicted pseudo-labels of unlabeled 3D hand activities sequences $\{x^u\}$. We note that consensus clustering is usually applied to predictions resulting from different clustering algorithms or the same algorithm with different hyper-parameters [25]. In our SCC strategy, the clustering algorithms and their hyper-parameters remain the same while only the data distribution changes. This strategy allows for achieving the best clustering results. In subsections 4.3.6 and 4.3.7, we quantitatively show its benefits and give more details about the adopted consensus clustering method.

4.3.5 Our proposed deep metric learning (*APML+MES*) loss function

To introduce our proposed loss function, we used the same mathematical symbols that we used in the previous chapters, primarily subsection 2.4.3, which presents the existing state-of-the-art loss functions. The introduced methods are based on the well-known classification

Softmax loss function. As we illustrate in figure 4.10, they generally exploit the weight vectors of the last classification layer to optimize the angular distance, forcing the network to learn a discriminative feature space that satisfies metric learning objectives.

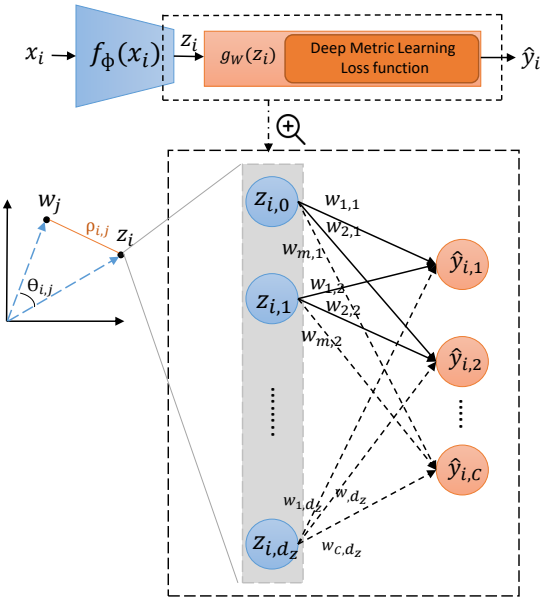

Fig. 4.10 Illustration of the backbone final layer and the fully connected classification layer that relies on state-of-the-art deep metric learning loss functions, which communly optimize the angle $\theta_{i,j}$ between the embedding z_i and the weight vectors w_j.

We recall that w_j is the j-th weight vector of the classification layer, where $j \in \{1,..,C\}$ and C is the number of labeled classes. The dimension of w_j is the same as the dimension of z_i and is equal to d_z. $\theta_{i,j}$ is the angle between the embedding z_i and the weight vectors w_j. Finally, $\rho_{i,j}$ is the Euclidean distance between w_j and z_i.

Automated Penalty-based Metric Learning (*APML*) loss function

Following our presentation and discussion of existing state-of-the-art deep metric learning loss functions (see chapter 2, section 2.4.3), we see that various manually assigned margin penalties, whether added on the angle or cosine space, all enhance intra-class compactness by penalizing the target logit [144]. However, it mainly depends on the manually assigned values of the margin and the re-scaling hyper-parameters. The choice of these values relies on data distribution and the dimension of embedding space of the used neural network backbone. We also note that [107, 35, 217] attributed the margin penalty only to the positive ground-truth classes, while the target logit for false classes is set to $cos(\theta_i)$. This aims to

improve intra-class compactness, but on the other hand, it favors overfitting, which limits the classification performance.

Based on these observations, we propose a fully automated penalty-based loss function that requires no additional hyper-parameters.

Unlike previous methods that focused on the angle between the embedding vectors z_i and the weights w_j, we focused on the Euclidean distance between these vectors. We denote it by $\rho_{i,j}(.,.)$ and we formulate it as follows:

$$\rho_{i,j}(z_i, w_j) = \left\| z_i - w_j \right\|_2 \tag{4.12}$$

Without normalizing w_j and z_i, the Euclidean norm is adaptively learned for minimizing the overall loss. Thus, features are learned in the Euclidean space, a widely recognized state-of-the-art choice.

To imply penalization that enhances the Euclidean margin, we interpret the distance e_j by a Student t-distribution. We fix the degree of freedom to 1, which is the same as a Cauchy heavy-tailed distribution in the low-dimensional map manifold $\mathscr{Z}$. We denote this distance probability by $\mathrm{p}_{i,j}(.)$ and we formulate it as follows:

$$\mathrm{p}_{i,j}(\rho_{i,j}) = (1 + \rho_{i,j})^{-1} \tag{4.13}$$

Following [210], we use the Student t-distribution with a single degree of freedom because it has the particularly nice property of approximating an inverse square law for large pairwise distances $\rho_j(w_j, z_i)$ in the $\mathscr{Z}$ space. This makes the map's representation of joint probabilities almost invariant to changes in the map's scale for embedded samples z_i that are far a par (e.g., noisy samples).

In contrast to [153, 107, 217, 35] who use the *Softmax* function to produce the probabilistic affinity score, we directly normalize the computed probabilities $\mathrm{p}_{i,j}$. This is justified by the fact that our target logit is based on a probability interpretation. It is also justified because of the incompatibility of losses based on the Euclidean margin with the *Softmax* loss as reported in [107, 225]. We formulate the probability normalization as follow:

$$\hat{y}_{i,j} = \frac{\mathrm{p}_{i,j}}{\sum_{j=1}^{C} p_{i,j}} \tag{4.14}$$

This results in a probability prediction score that we denote by $\hat{y}_i = \{\hat{y}_{i,1}, ..., \hat{y}_{i,C}\}$. Finally, our loss function that we denote by $\mathscr{L}_{APLM}$, is formulated in combination with the cross-

entropy loss as follow:

$$\mathscr{L}_{APLM} = -\frac{1}{N}\sum_{i=1}^{N} y_i log(\hat{y}_i) \tag{4.15}$$

Where y_i is the one-hot encoding ground-truth label of the i-th sample x_i.

In this manner, the *APML* loss function will optimize the Euclidean distance. So, the embedded feature vectors z_i will be attracted and move closer to their corresponding centers (weight vectors w_j) in the respect to the ground-truth labels, which results in highly class compactness.

Mutually equidistantly separation (MES) loss function

As a complement to *APML* loss, the *MES* loss function attempts to imply constraints that forces the network to ensure the best inter-class separation in the feature space. This is done by maximizing the separation of the C class centers: $\{w_{j=1:C}\}$. In the 2-sphere $\mathbb{S}^2$ space, this requirement meets the Tammes's Problem [Tammes], also called best-packing problem, which is stated as follows: *What arrangement of C points on the surface of a sphere will maximize the minimum distance between any two points?*

The generalization of Tammes's Problem to d-sphere $\mathbb{S}^d$ space was asked by Böröczky [60], who addressed the problem by proving the following theorem:

Theorem 4.3.5 *For the d-sphere $\mathbb{S}^d$, $d \geq 2$, and $2 \leq C \leq d+2$, best-packing configurations are uniquely given by the vertices of regular $(C-1)$-simplices inscribed in $\mathbb{S}^d$ with centers at the origin.*

According to the Theorem 4.3.5, mutually equidistantly separated C points in d-sphere $\mathbb{S}^d$ space, with $2 \leq C \leq d+2$ and $d \geq 2$, is the unique optimal solution that maximize the separation of these C points. This motivated us to separate the C centers $\{w_{j=1:C}\}$ mutually equidistantly to maximizes the inter-class sepration in the feature space. To this end, we first computed mutually all possible Euclidean distances between the centers that we denote by ρ_l and we define as follows:

$$\rho_l = \left\| w_{j_1} - w_{j_2} \right\|_2 \tag{4.16}$$

Where $1 \leq j_1 \neq j_2 \leq C(C-1)/2$ and $l = 1 : C(C-1)/2$ is the l-th number of possible distances. Once all possible distances $\{\rho_{l=1:C(C-1)/2}\}$ are computed, we calculate the *MES* loss that refers to the variation of these distances, which we denote by $\mathscr{L}_{MES}$ and define as follows:

$$\mathscr{L}_{MES} = \frac{1}{C(C-1)/2} \sum_{l=1}^{C(C-1)/2} (\rho_l - \overline{\rho})^2 \tag{4.17}$$

Where $\overline{\rho}$ is the mean value of all the distances $\{\rho_{l=1:C(C-1)/2}\}$. The minimization of $\mathscr{L}_{MES}$ will leads to the minimization of the variation of the mutual distances, which results in a mutually equidistantly distribution of the centers $\{w_{j=1:C}\}$. This guaranties optimal inter-class separation (see figure 4.11).

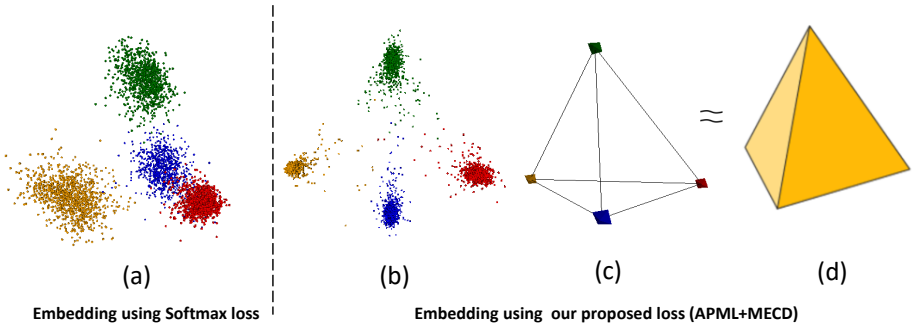

Fig. 4.11 Figures (a) and (b) show the embedding of 4159 test samples into a 3D high-level feature space using learned NNs, where yellow, blue, green, and red colors refer to classes 1, 2, 3, and 4 of the MNIST dataset [36], respectively. Figure (a) when training the model using the *Softmax* loss and figure (b) when using our proposed DML composite loss function (*APML +MES*). The embedding in figure (b) shows the impact of our proposed loss function in enhancing the discriminative power of the neural network. Figure (c) shows the centers of the classes, which refer to the weight vectors of the last neural network layer at the end of the training procedure. The proposed loss forced the neural network to distribute the centers mutually equidistantly in the 3D feature space, resulting in a regular Tetrahedron configuration. Nevertheless, the regular Tetrahedron 3-samplex (figure (d)) is the unique optimal solution to the so called Tammes's Problem [60], which aims at separating the 4 centers to their maximum on a 2-sphere $\mathbb{S}^2$ space [7]. Such a solution perfectly meets the metric learning objective of maximizing the inter-class separation.

In subsection 4.3.7, we show quantitatively the efficiency of the *MES* loss in combination with our proposed *APML* and existing state-of-the-art DML losses.

Convergence constraint of the *MES* loss function

According to Theorem 4.3.5, by putting the embedding dimension $d_z = d+1$, the convergence of $\mathscr{L}_{MES}$ to the optimal solution constraint is that the total number of classes C must respects the following inequality:

$$2 \leq C \leq d_z + 1 \tag{4.18}$$

For $C > d_z + 1$, presently, to our knowledge there is no general solution for any number of $C \in \mathbb{N}$. Exceptionally, when $d_z = 3$, optimal solutions that maximize C points separation in spherical space $\mathbb{S}^2$ are known for C = 5, 6, ..., 14 and for C = 24 as reported in [16].

The composite DML loss function (*APML+MES*)

Finally, to allow tuning between the two introduced losses *APML* and MES, we define the composite DML loss function $\mathscr{L}_{APML+MES}$ as follows:

$$\mathscr{L}_{APML+MES} = \alpha \mathscr{L}_{APLM} + \beta \mathscr{L}_{MES} \tag{4.19}$$

Where α and β are weights parameters for each loss. With the simultaneous optimization of the *APML* and *MES* losses, we minimize intra-class distances and explicitly maximize inter-class separation in the feature space. In subsection 4.3.7, we analyze the tuning between the two losses.

4.3.6 Experiments on hand activity

Dataset

To validate our proposed approaches on a real-world application, we used the **FPHA dataset** [49] and our **SLS France dataset**. We used the same data partitioning configurations (*Config1*, *Config2* and *Config3*) as in the previous section 4.2. The known classes are considered source domain classes, and the unknown ones as the target domain.

Evaluation protocols and metrics

We adopted two protocols (*Protocol 1* and *Protocol 2*) to evaluate the clustering of the unlabeled activities of the two target domains *"social"*, *"office"* and the combination of (*"kitchen"+"social"+"office"*) that belong to the three configurations *Config1*, *Config2* and *Config3*, respectively. We describe the two protocols as follows:

- ***Protocol 1***: we assume that the number of possible categories in the target domains is unknown. Consciously, we performed clustering without giving the ground-truth number of clusters k by using the Affinity Propagation clustering [47], which we evaluated using V-measure (Vm) [160] and Adjusted Mutual Information (AMI) [159] metrics. The V-m metric represents the harmonic mean of the homogeneity and the completeness metrics. We highlight the homogeneity metric (Vm-h) since it provides the rate of meaningful detected clusters.
- ***Protocol 2***: we gave the ground-truth number of clusters, and we used the Agglomerative clustering [247], which we evaluated using the Clustering Accuracy (ACC) [234].

For more details on the used evaluation metrics, we refer the reader to chapter 2, section 2.4.6.

We also evaluated the activity recognition accuracy on the two source domains that belong to the two configurations *Config1* and *Config2*. This is done by training the recognition model using our *APML+MES* loss function on train samples, then testing it based on the *APML* loss outputs on test samples.

Implementation details

- **Supervised learning (step 1).** We pre-trained the neural network architectures proposed in chapter 3 sections 3.2 and 3.3 for 3D skeleton-based and RGB-based hand activity recognition, respectively. We kept the exact implementation and data normalization settings for all the experiments.

- **Clustering of unlabeled hand activities (step 2).** We recover the pre-trained model, discard its final layer, and we use it to project unlabeled samples into the 128D feature space. Next, we perform clustering on these mapped samples adopting the protocols *Protocol 1* and *Protocol 2*. Following our SCC strategy, steps 1 and 2 are repeated 100 times. Therefore, in the end, we have 100 predicted sets of labels.

- **Consensus clustering (step 3).** We adopted the approach proposed by [193] introduced in chapter 2, section 2.4.5. It combines three efficient heuristics to solve the consensus clustering problem. The consensus clustering function takes the 100 predicted sets of labels as an input and outputs the final prediction set of labels that belong to the unlabeled activity samples.

State-of-the-art comparison

Tables 4.4 and 4.5 show unlabeled hand activity clustering evaluation results following the evaluation of *Protocol 1* and *Protocol 2*, respectively. We compared the impact of our *APML+MES* loss function with four state-of-the-art losses. As expected, the *Softmax* loss performs very poorly against metric learning-based losses for both protocols. We can see that our proposed *APML+MES* loss function performs a little better than the-state-of-the-art methods in *Config 1* and *Config 3*, while it still equivalent in *Config 2*.

We notice a performance drop in *Config 2*. This can be explained by the fact that in *Config 2*, the number of labeled samples decreases while the number of unlabeled samples and their classes increases. Nevertheless, in contrast, the UDA requires that the number of labled samples in the source domain be much larger than unlabeled ones in the target domain.

Table 4.4 Clustering results adopting the *Protocol 1*. The clustering is performed on unlabeled activity samples of the three target domains *"social"*, *"office"* and all samples of the SLS France dataset, which belong to *Config 1*, *Config 2* and *Config 3*, respectively. The $\hat{k}$ is the number of detected clusters. The AMI, Vm, and Vm-h metrics are computed on the final clustering result of our SCC strategy. For the FPHA dataset, we tested using RGB and 3D skeleton activities, while for the SLS France dataset we used the provided RGB image activities (the only provided data type).

		Config 1				**Config 2**				**Config 3**			
		AMI	**Vm**	**Vm-h**	**k**	**AMI**	**Vm**	**Vm-h**	**k**	**AMI**	**Vm**	**Vm-h**	k
3D	**Softmax**	62.36	61.55	64.41	**11**	35.16	45.97	43.77	**17**	-	-	-	-
	SphereFace[107]	66.14	69.54	77.00	13	38.62	46.39	48.33	**17**	-	-	-	-
	CosFace [217]	66.33	69.98	78.98	14	40.07	47.99	50.46	18	-	-	-	-
	ArcFace [35]	66.37	69.91	77.83	14	39.63	47.87	**50.59**	19	-	-	-	-
	APML	68.29	71.49	79.56	13	40.16	46.09	49.97	**17**	-	-	-	-
	APML+MES	**68.72**	**72.65**	**79.96**	14	**40.56**	**48.22**	50.38	18	-	-	-	-
RGB	**Softmax**	63.28	62.47	65.70	11	36.61	46.19	44.42	17	32.08	42.14	40.20	14
	SphereFace[107]	67.77	70.42	77.12	**13**	40.03	47.21	49.00	**16**	35.55	43.49	45.49	**14**
	CosFace [217]	67.71	70.39	80.57	14	41.16	48.70	52.23	18	36.82	44.16	46.00	16
	ArcFace [35]	67.69	72.01	81.38	14	40.70	49.15	52.04	19	35.66	44.21	46.01	18
	APML	69.86	72.82	81.48	**13**	42.10	47.73	51.27	17	36.96	44.66	46.43	16
	APML+MES	**70.03**	**72.32**	**81.84**	14	**42.28**	**49.88**	**52.61**	18	**37.22**	**45.14**	**46.75**	18

Table 4.5 ACC results adopting the *Protocol 2*. The min, max and mean values are computed on a population of 100 tries. The SCC refers to the consensus clustering results based on the 100 tries following our SCC strategy. The clustering is performed on unlabeled activity samples of the three target domains *"social"*, *"office"* and all samples of the SLS France dataset, which belong to *Config 1*, *Config 2* and *Config 3*, respectively. For the FPHA dataset, we tested using RGB and 3D skeleton activities, while for the SLS France dataset, we used the provided RGB image activities (the only provided data type).

		Config 1				**Config 2**				**Config 3**			
		min	**max**	**mean**	**SCC**	**min**	**max**	**mean**	**SCC**	**min**	**max**	**mean**	**SCC**
3D	**Softmax**	48.34	79.62	64.22	65.87	29.51	49.69	39.37	45.18	-	-	-	-
	SphereFace[107]	55.45	76.77	67.26	73.45	**34.63**	**49.09**	41.38	**50.60**	-	-	-	-
	CosFace[217]	52.60	79.62	67.37	73.93	32.53	46.08	39.59	47.59	-	-	-	-
	ArcFace [35]	54.02	79.62	67.12	71.09	32.83	46.98	39.32	47.28	-	-	-	-
	APML	54.97	88.62	67.83	75.82	34.03	47.59	40.37	47.39	-	-	-	-
	APML+MES	**56.01**	**79.88**	**68.15**	**75.97**	34.52	48.81	**41.66**	47.44	-	-	-	-
RGB	**Softmax**	50.24	79.51	65.22	65.39	31.83	49.78	42.22	46.60	25.88	45.15	36.32	42.13
	SphereFace[107]	56.82	80.61	69.14	75.64	34.39	49.11	42.49	48.00	31.45	46.81	37.19	47.20
	CosFace[217]	54.11	81.88	68.31	74.97	34.51	47.39	40.54	48.15	29.16	43.61	36.57	45.71
	ArcFace [35]	55.69	81.61	68.71	75.29	34.46	47.32	40.89	48.16	29.70	43.96	36.32	45.66
	APML	55.91	90.09	69.11	76.03	35.43	48.23	43.33	49.50	31.86	46.90	37.25	47.77
	APML+MES	**57.21**	**91.55**	**71.25**	**78.74**	**36.21**	**49.98**	**45.25**	**51.00**	**31.98**	**47.12**	**37.89**	**49.85**

Unlabeled activity clustering analysis

Confusion matrices in figure 4.12 show the highly meaningful quality of clustering provided by our method. Indeed, most of the confused activity classes are meaningfully too close for e.g., *"give coin"* and *"give card"* classes where the performed action *"give"* remain the same, while only the objects *"card"* and *"coin"* change. We also observe a high confusion between "toast wine" and "pour wine" classes where the action changes, but the social activity context is still meaningfully close. Even for the very challenging *Config 2*, the confusions are still significant, e.g., *"read letter"* and *"take letter"*.

When adopting the *Protocol 1* (k not given), we still get a high homogeneity score Vm-h (Table 4.4). This means that regrouped samples in each detected clusters are mostly similar, confirming the meaningful clustering quality.

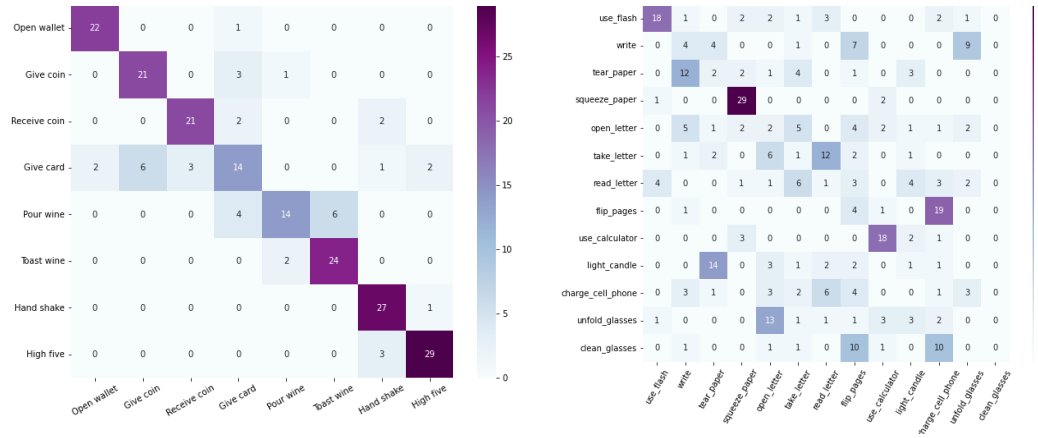

Fig. 4.12 The left and right matrices refer to the confusions of unlabeled activities clustering of the two target domains *"social"* and *"office"* that belong to *Config1* and *Config2* respectively. The confusions are computed based on the protocol *Protocol 2*.

Impact of the Statistical and Consensus Clustering (SCC) strategy

As we explained in subsection 4.3.3, in our preliminary experiments, we observed a high sensitivity of clustering algorithms to the randomness present in the pre-training (step 1). Figure 4.13 (a) shows the clustering accuracy results (ACC) over 100 tries. We can see the sensitivity of clustering algorithms regarding the feature space provided by the pre-trained model equipped with our *APML+MES*. The sensitivity is also observed using different loss functions, as shown in Figure 4.13 (b).

This observation motivated us to propose the SCC strategy, which allows selecting a good clustering prediction among multiple tries concerning the unsupervised learning constraint.

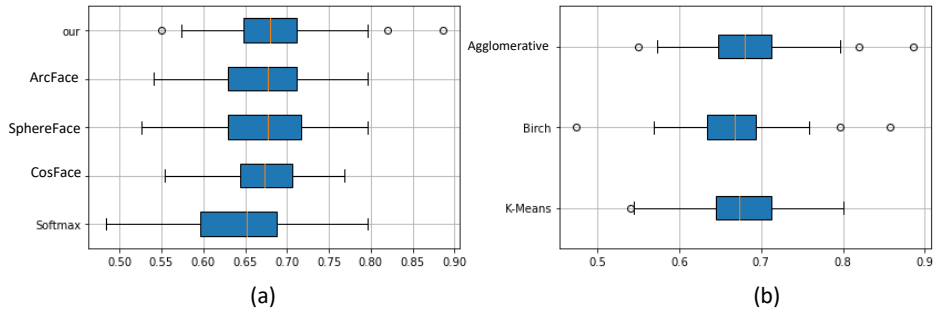

Fig. 4.13 (a) ACC over 100 tries. The clustering algorithms and their hyper-parameters remain the same while only the data distribution changes. (b) ACC over 100 tries. The clustering is performed using the Agglomerative clustering algorithm while testing with different loss functions.

Table 4.4 and 4.5 confirm the advantage of our SCC strategy. For 100 tries, the final result of the consensus clustering is far from the minimum, above the average, and not so far from the maximum ACC. The SCC strategy allows selecting a good clustering prediction among multiple tries with respect to the unsupervised learning constraint.

Performance of the *APML+MES* loss on activity recognition (classification)

Table 4.6 shows that our proposed loss overcomes all the metric learning-based losses in classification accuracy, while it is still not too far from the *Softmax* loss function. These results validate our *APML+MES* loss function's main contribution, which aims to balance learning a clustering-friendly manifold that facilitates the clustering of the embedded unlabeled samples of the target domain and good recognition accuracy for the labeled ones of the source domain.

Table 4.6 The average recognition accuracy results over 100 tries. The recognition train/test is performed on the data of the two source domains that belong to *Config 1* and *Config 2*.

		Softmax	*SphereFace[107]*	*CosFace[217]*	*ArcFace[35]*	*APML*	*APML+MES*
3D	***Config1***	**94.80**	86.67	84.72	82.53	**95.46**	91.20
	Config2	**96.24**	90.30	88.88	86.62	**96.51**	92.42
RGB	***Config1***	**95.23**	88.32	85.62	84.88	**95.88**	92.55
	Config2	**97.52**	91.66	90.19	89.70	**97.41**	95.21

4.3.7 Experiments on clustering and image-retrieval tasks

For fair validation and comparison of our proposed deep metric learning loss function with the state-of-the-art losses, we present experiments on clustering and image retrieval tasks in this subsection.

Datasets

Following the experimental protocol in [189], we evaluate the clustering and k nearest neighbor retrieval on data from previously unseen classes on the following previously introduced (see chapter 2, subsection 2.4.8) datasets:

- **CUB200-2011.** [214] It has 200 classes of birds with 11,788 images. We split the first 100 classes for training (5,864 images) and the rest of the classes for testing (5,924 images).
- **Cars196.** [86] It has 198 classes of cars with 16,185 images. We split the first 98 classes for training (8,054 images) and the other 98 classes for testing (8,131 images).

We also used the (Modified National Institute of Standards and Technology) **MNIST** [36] dataset to evaluate and enhance the analysis of the proposed approach. MNIST is a large collection of handwritten digits. It has a set of 70,000 images.

Evaluation metrics

For the clustering task, we used the *K-means* clustering with the desired number of clusters set equal to the number of classes in the test set. The clustering quality is measured with the standard normalized mutual information NMI metric [170] previously introduced in chapter 2, section 2.4.6. We use the Recall@K [72] metric for the retrieval task. Scikit-learn [143] framework is used for the clustering and NMI evaluation metric implementations. For the MNIST dataset, in addition to *K-means*, we used the previously introduced *BIRCH* and *Agglomerative* clustering algorithms. We refer the reader to chapter 2, subsection 2.4.4 for more details about the used clustering algorithms.

Implementation details

- **Clustering and image retrieval experiment on CUB200-2011 and Cars196 datasets.** For a fair comparison, we use the Resnet50 [59] architecture with batch normalization [70] as the main backbone for all the compared methods. The Resnet50 model is first pre-trained on ILSVRC 2012-CLS dataset [164], then fine-tuned for the evaluation. As

illustrated in figure 4.14, we added an FC layer on top of the final average pooling layer of the Resnet50 backbone network to learn the mapping of varying dimensionality d_z. To stabilize the sensitivity to the random initialization of the FC layer, we added a layer normalization before the last FC classification layer.

All the train images of CUB200-2011 and Cars196 are normalized to 256 by 256. For training data augmentation, all images are randomly cropped at 227 by 227 and randomly mirrored horizontally. During testing, after images are normalized to 256 by 256, they are center cropped to 227 by 227. Following [241], we used a balanced class sampling to construct training mini-batches by sampling 25 samples per class for Cars196 and CUB200. All the methods are trained for 65 epochs using the Adam optimizer with a learning rate of 0.001 and a 1e-4 weight decay. We note that for CUB200-2011 and Cars196, we did not use our SCC strategy because the datasets are large and require significant computation time.

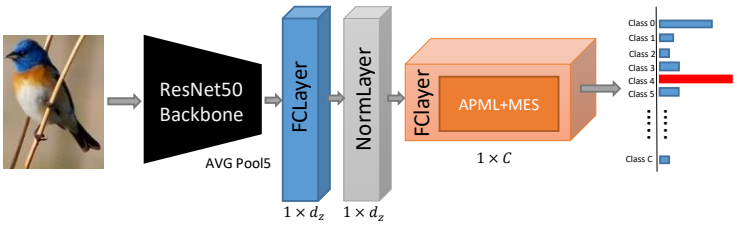

Fig. 4.14 Illustration of the proposed training neural network architecture. Where d_z is the embedding dimension and C is the number of classes.

- **Clustering experiment on the MNIST dataset.** For the MNIST dataset, we used a CNN neural network backbone based on VGG8, where the dimension of the feature space is fixed to $d_z = 64$. The network is trained for ten epochs with a learning rate of 0.001 and a batch size of 32. To simplify the analysis, we split the dataset as follows: {*"2", "3", "4", "5","6","8"*} digits are used as known training classes; while {*"0", "1","9","7"*} digits are used as unseen classes that our method seek to cluster. We have chosen this data split because there is no state-of-the-art partitioning base to rely on. Moreover, the digit classes *"1"* and *"7"* are very similar and this presents a real challenge for the clustering. As illustrated in figure 4.15, we used our SCC strategy with 100 runs.

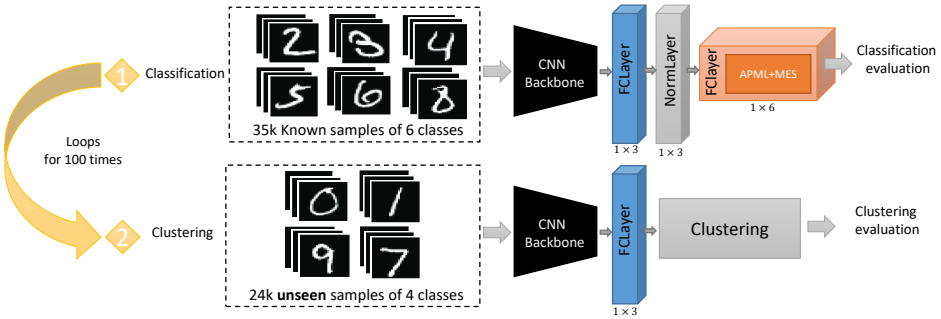

Fig. 4.15 Illustration of the training (supervised classification) and the testing (unsupervised clustering) steps on the MNIST dataset. The two steps are repeated 100 times following our proposed SCC strategy. The network is first pre-trained to classify {*"2", "3", "4", "5","6","8"*} digits classes using our proposed *AMPL+MES* loss function. Then, the pre-trained network is used to map the unlabeled samples of the {*"0", "1","9","7"*} digits classes, which are clustered using classical clustering algorithms. Finally, a consensus clustering is performed on the 100 clustering predictions to provide the final clustering prediction.

- **Clustering experiment using Variational Autoencoder (VAE).** We also compared the effect of our proposed method with a VAE [82]. This comparison aims to determine how efficient our approach is compared to VAEs since they are shown to be exploitable for similar image clustering tasks [150]. A VAE is a neural network architecture that is regularised during an unsupervised training procedure to ensure that its latent space has suitable properties that allow the generation of new approximative samples. Moreover, these properties allow performing clustering of unlabeled samples [150]. We implemented a simplified CNN VAE that we trained on the {*"0", "1","9","7"*} digit classes, and we tested for clustering the {*"0", "1","9","7"*} digit classes, as shown in figure 4.16. The Encoder is composed of 4 convolutional layers, and the weights of the reconstruction and kullback-leibler losses are fixed to 1 and 0.001, respectively.

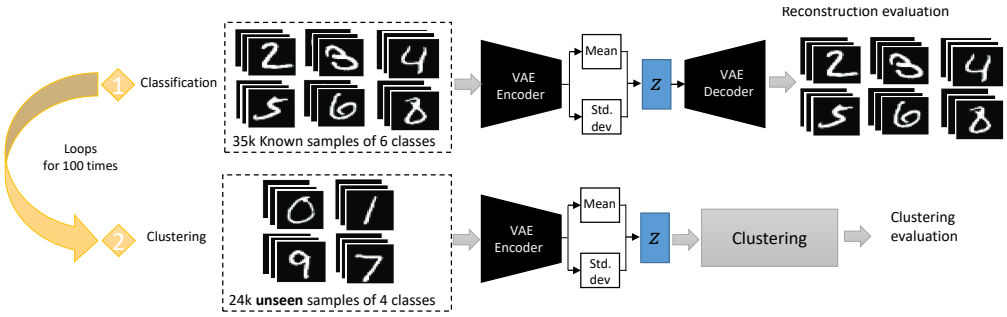

Fig. 4.16 Illustration of the training (unsupervised reconstruction) and the testing (unsupervised clustering) steps on the MNIST dataset. The two steps are repeated 100 times following our proposed SCC strategy. The pre-trained Encoder part of the VAE is used to map the unlabeled samples of the {*"0", "1","9","7"*} digits classes, which are clustered using classical clustering algorithms. Finally, a consensus clustering is performed on the 100 clustering predictions to provide the final clustering prediction.

We used Tensorflow [1] package for our implementation for all the methods.

State of the art comparison

Comparaison on CUB200-2011 and CARS-196 datasets. Finally, we compared our composite *APML+MES* loss and the baselines [107, 217, 35] that we boosted with the proposed MES loss against state-of-the-art metric learning approaches on CUB200-2011 and CARS-196 datasets. For the reproduced boosted baselines, we trained two variants: one with their recommended network architecture of 512-dimensional embeddings and one with our modifications of 2048-dimensional embeddings, which showed more improvement. For a fair comparison, we compared the approaches that use Resnet50 as a neural network backbone for low-level feature learning and the same evaluation metrics.

Table 4.7 Comparison of clustering and retrieval performance of our approach and modified baselines against the state-of-the-art metric learning approaches on CUB200-2011 and CARS-196 datasets. All methods are trained using ResNet50 neural network backbone. The Top-2 results are marked in bold.

		CUB-200-2011					**CARS196**				
Method	ED	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**
Margin Loss [117]	128	69.0	63.6	74.4	83.1	90.0	69.1	79.6	86.5	91.9	95.1
Normalized *Softmax* [241]	512	69.7	61.3	73.9	83.5	90.0	74.0	84.2	90.4	94.04	96.9
Divide and Conquer [169]	1028	69.6	65.9	76.6	84.4	90.6	70.3	84.6	90.7	94.1	96.5
Proxy NCA++ [204]	512	71.3	66.3	77.8	**87.7**	91.3	71.5	84.9	90.6	94.9	97.2
DiVA [120]	512	71.4	69.2	**79.3**	-	-	72.2	87.6	**92.9**	-	-
PADS [162]	128	69.9	**67.3**	78.0	85.9	-	68.8	83.5	89.7	93.8	-
Proxy Anchor [81]	512	-	**69.7**	**80.0**	**87.0**	**92.4**	-	**87.7**	**92.9**	**95.8**	**97.9**
SphereFace [107]+MES	512	70.96	64.46	74.57	83.37	90.47	71.14	83.53	91.1	93.97	95.75
CoseFace [217]+MES	512	66.94	59.75	70.57	80.36	87.4	69.99	84.35	89.7	91.24	94.6
ArcFace [35]+MES	512	72.75	65.42	74.66	84.0	90.68	71.6	86.45	91.93	93.75	96.32
APML+MES	512	72.82	65.31	74.12	84.1	90.72	71.21	86.73	91.43	93.69	96.43
SphereFace [107]+MES	2048	73.24	66.69	75.54	84.23	90.53	74.44	86.25	92.13	94.0	96.63
CoseFace [217]+MES	2048	72.47	67.03	75.67	84.75	**91.50**	71.39	85.68	91.89	93.40	96.74
ArcFace [35]+MES	2048	**73.35**	66.74	75.50	83.69	89.73	**74.80**	**87.89**	92.24	**95.92**	97.54
APML+MES	2048	**73.26**	66.98	75.49	84.45	91.22	**74.82**	87.69	92.28	95.64	**97.62**

Table 4.7 shows that our proposed composite loss *APML+MES* and the boosted baselines outperform the the-state-of-the-art methods on the clustering task by more than 1.95% and 0.8% of NMI for the CUB200-2011 and CARS-196 datasets respectively. Nerveless, the performance of the proposed methods is not very far from image retrieval dedicated methods such as Proxy Anchor [81]. We can see that the combination of *ArcFace* [35]+MES showed a good performance on the CARS196 dataset for both clustering and image retrieval tasks. Hence, our principal composite loss *APML+MES* gives an equivalent performance without requiring any additional hyper-parameter, such as the margin and re-scaling values.

Comparaison on MNIST dataset. Table 4.8 shows the clustering results on the MNIST dataset. We compared the impact of our *APML+MES* loss function with four state-of-the-art losses and the VAE architecture. The *Softmax* loss function and the VAE perform poorly against metric learning-based losses. Our proposed loss is equivalent to the state-of-the-art or even better by achieving 98.49% of ACC. Figure 4.17 shows the mapping of test unlabeled samples of the {*"0", "1", "9", "7"*} digits classes. Unlabeled samples are mapped using the neural network pre-trained with our *APML+MES* loss function. We can see that the classes are well separated, although the classes *"1"* and *"7"* are usually very close. This proves the ability of our method to learn a friendly clustering feature space.

The results in table 4.8 also confirm the effectiveness of our SCC strategy again. For 100 tries, the final result of the consensus clustering is far from the minimum, above the average, and not so far from the maximum ACC.

Table 4.8 ACC results. The min, max and mean values are computed on a population of 100 tries. The SCC refers to the consensus clustering results based on the 100 tries following our SCC strategy. The clustering is performed on unlabeled samples of the {*"0", "1","9","7"*} digits classes of the MNIST Dataset.

	K-means				***Birch***				***Agglomerative***			
	min	***max***	***mean***	***SCC***	***min***	***max***	***mean***	***SCC***	***min***	***max***	***mean***	***SCC***
Softmax	68.25	91.72	86.89	90.74	58.78	94.08	83.26	92.03	58.45	95.22	86.78	92.03
SphereFace [107]	84.51	95.22	92.88	94.22	85.11	98.16	94.93	98.13	85.69	98.52	95.18	98.10
CosFace [217]	87.77	95.07	92.74	94.48	87.59	98.39	93.61	97.37	87.64	97.41	93.80	96.87
ArcFace [35]	91.04	95.22	93.27	94.55	87.47	98.24	93.24	96.41	88.49	98.26	94.44	96.98
VAE	60.47	89.93	85.53	90.82	62.01	90.59	87.38	91.13	62.01	90.59	87.38	92.22
APML+MES	**93.24**	**96.38**	**94.82**	**95.62**	**89.11**	**98.63**	**96.13**	**98.47**	**89.15**	**98.89**	**96.22**	**98.49**

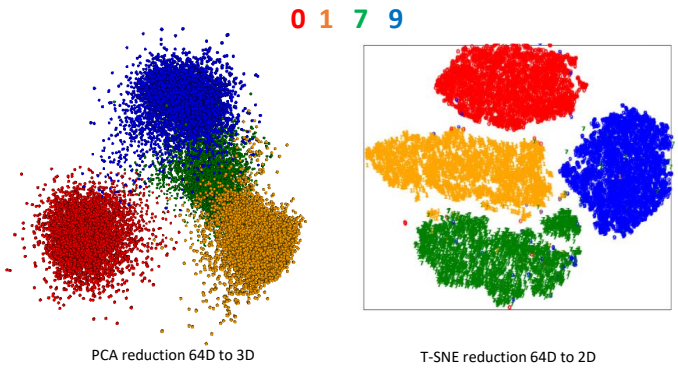

Fig. 4.17 The mapping (embedding) of unlabeled samples of the {*"0", "1","9","7"*} digits classes. The mapping network is trained to classify labeled {*"2","3","4","5","6","8"*} digits classes equiped with our APML+MES loss function. We used Principal Component Analysis (PCA) [181] to reduce the dimension from 64D to 3D, while T-SNE [210] is used to reduce the dimension from 64D to 2D

Effects of the *MES* loss function.

To analyse the impact of our proposed *MES* loss function, we evaluated our proposed *APML* loss and existent losses [107, 217, 35] with and without our complementary *MES* loss. Table 4.9 shows the results of clustering and image retrieval tasks on the CUB200-2011 dataset.

Indeed, by adding the *MES* loss, the Recall@1 and the NMI scores has been improved by an average of 2.1% and 2.7% respectively.

Table 4.9 Clustering and retrieval performance with and without using our *MES* loss in combination with our *APML* loss and existent DML losses. The test was done on the 100 unseen classes of CUB200-2011 dataset. Results show a clear improvement of DML loss functions when combined we the *MES* loss.

Methods	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**
SphereFace [107]	70.87	64.97	74.24	79.89	84.72
CosFace [217]	70.25	64.58	74.83	80.77	85.26
ArcFace [35]	70.17	65.07	74.74	80.24	84.60
APML	70.10	64.18	74.56	80.21	88.31
SphereFace[107]+MES	73.24	66.69	75.54	84.23	90.53
CosFace [217]+MES	72.47	67.03	**75.67**	**84.75**	**91.50**
ArcFace[35]+MES	**73.35**	66.74	75.50	83.69	89.73
APML+MES	73.26	**66.98**	75.30	84.45	91.22

In addition, we provide more visualizations that support these findings. So, in figure 4.18 we visualize the difference between the embeddings of CUB200-2011 unseen test samples of a given batch after only one epoch of training: (a) while using only the *APML* loss and (b) while using *APML+MES*. We can observe that training with *MES* favors the intra-class compactness and inter-class separation, which facilitates the clustering and retrieval tasks. This confirms the effectiveness of the optimal separation of class centers in the training procedure, which is imposed by our *MES* loss.

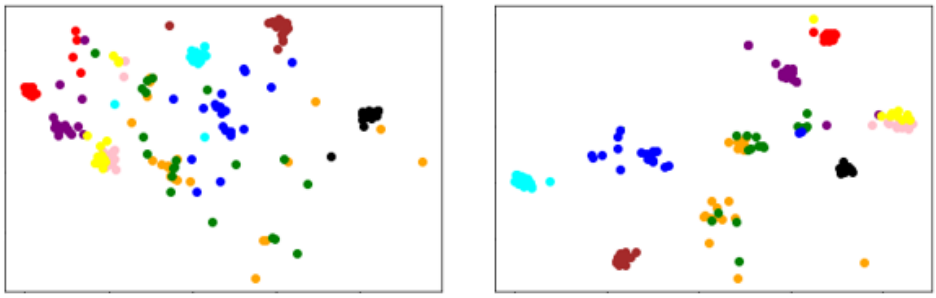

Fig. 4.18 The figure shows the embeddings of CUB200-2011 unseen test samples of a given batch after only one epoch of training. For clarity, we selected only the first 10 classes of 100. (a) When training only with the *APML* loss. (b) when training with both *APML+MES*. The t-SNE is used for the 2048D to 2D mapping. As can be seen in (b), the mapped samples after training with the MES are much more clustered than when training without MES.

Tuning between *APML* and *MES* loss functions.

During our experiments on the CUB200-2011 and Car196 datasets, we observed that the MES loss value is relatively small in the training procedure. The main reason is that the number of training classes C is very small for both datasets compared to the chosen embedding dimension of 2048. So selecting a relatively large β could help achieving best optimization. To this end, we varied β from $\{1, 50, 100, 200, 300, 400, 500, 1000\}$ while keeping $\alpha = 1$ to seek an optimal parameter. Figure 4.19, shows the corresponding Recall@1 and NMI scores. We can observe that that the Recall@1 and NMI scores reaches the maximum when they are at $\beta = 500$ and $\beta = 400$ for the CUB200-2011 and Car196 datasets, respectively.

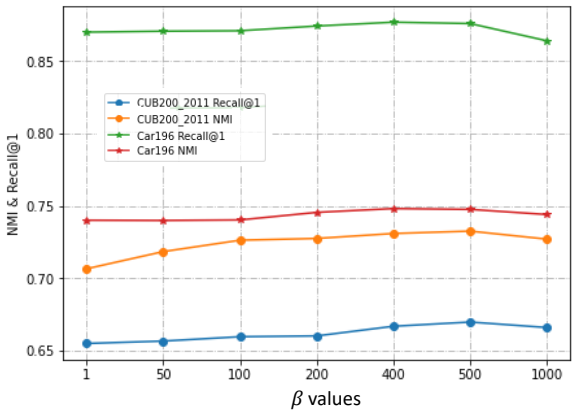

Fig. 4.19 Clustering and retrieval performance on CUB200-2011 and Car196 datasets. The results are given while varying the weight value β of MES loss, with the weight value of the *APML* loss kept to $\alpha = 1$.

Embedding dimensionality.

We study the effects of dimensionality on our method by varying only the embedding dimension d_z (number of neurons of the FC layer before the normalization layer) from $\{64, 128, 256, 1024, 2048, 4056\}$ while keeping all other optimization hyper-parameters fixed. Figure 4.20 shows the obtained Recall@k1 and NMI scores on the CUB200-2011 dataset. We can consistently observe that dimensionality is directly related to retrieval and clustering performance, which can be explained by the fact that the larger the dimension, the richer the feature vector is of relevant information. We note that for $d_z = 64$, the convergence to an optimal separation condition of the MES loss (Section 4.3.5) is not verified, since in this case $d_z + 1 < C$ with the number of classes $C = 100$. We observe that the clustering performance drops at dimension 4056. This is due to the incompatibility of k-means clustering with a high dimensionality clustering, and the Euclidean distance divergence [3]. We do not go more than the dimension of 2048, which can be computationally costly for retrieval and clustering tasks.

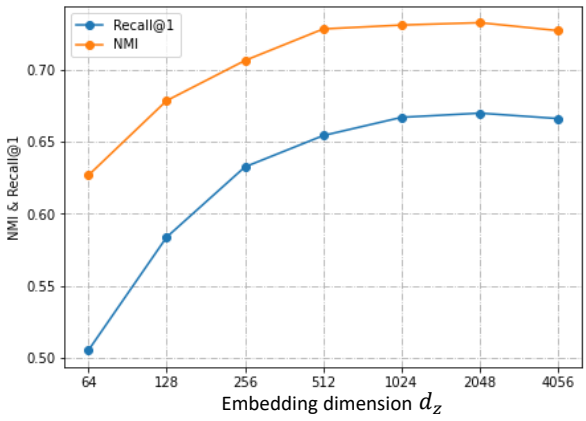

Fig. 4.20 Clustering and retrieval performance on CUB200-2011 while varying embedding dimensions. Our *APML+MES* based embeddings improve performance when increasing dimensionality.

MES loss convergence analysis

To analyze the behavior of our *APML+MES* loss function when the convergence to a mutually equidistant distribution condition is not satisfied, we fixed the feature space dimension to $d_z = 3$, and we variated the number of classes $C \in \{4, 6, 8, 12\}$. Figure 4.21 presents the results. For $C = 4$, the *APML+MES* loss forced the neural network to ensure a mutually equidistantly separation of the four classes in the 3D feature space, resulting in a regular Tetrahedron [7]. For $C \in \{6, 8, 12\}$, the condition is not satisfied since $C > d_z + 1$. Nevertheless, the *APML+MES* loss forced the neural network to find the optimal solutions that maximize C points separation in $\mathbb{S}^2$ spherical space [16], resulting in Platonic solid configurations. In addition to the optimal separation, we can see the high compactness of the embedded samples, where similar samples are very close to each other. This proves the effectiveness of our proposed loss function in encouraging the neural network to learn a highly discriminative feature space that satisfies the metric learning objectives.

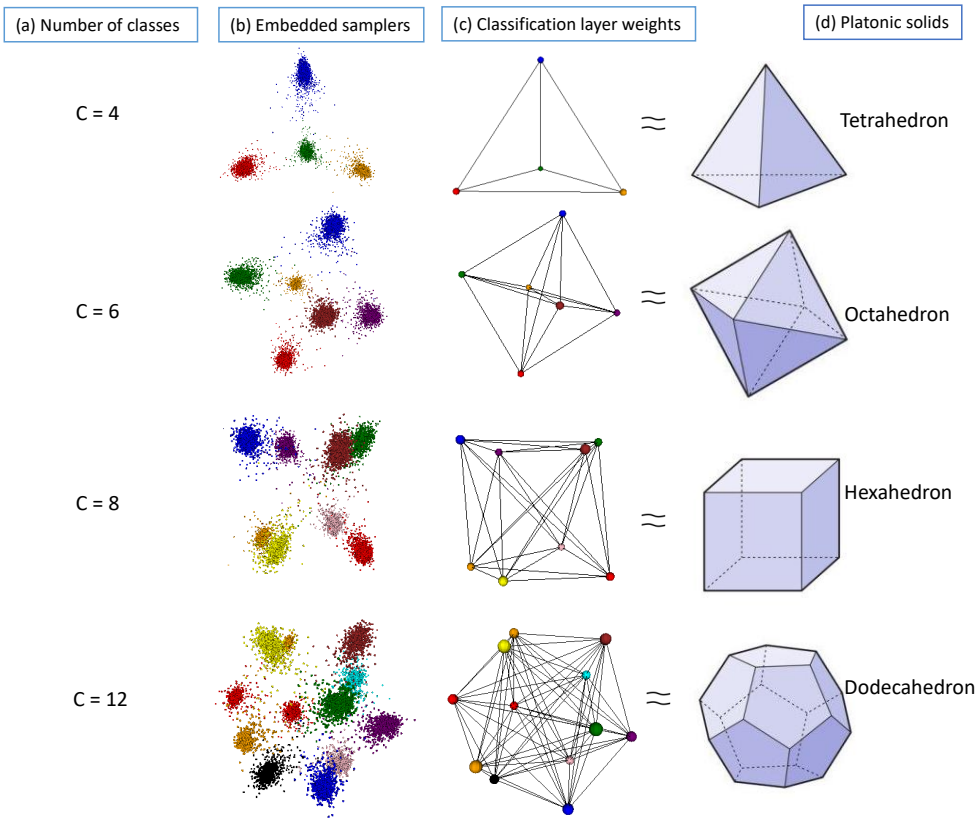

Fig. 4.21 Column(a) presents the number of classes used to learn classification using a neural network equipped with our *APML+MES* loss function. Column (b) shows the embedding of samples into a 3D high-level feature space using the learned neural network. The training and the testing samples belong to the MNIST dataset [36], while for $C = 12$, we added two classes from the Fashion-MNIST dataset [229]. The embedding in column (b) shows our loss function's impact on enhancing the neural network's discriminative power. Column (c) shows the centers of the classes, which refer to the weight vectors of the last neural network layer at the end of the training procedure. The proposed loss forced the neural network to distribute the centers optimally in the 3D feature space, resulting in Platonic solid configurations (column (d)). Nevertheless, the Platonic solid configurations represent optimal solutions that maximize C points separation in $\mathbb{S}^2$ spherical space [16].

4.3.8 Conclusion

We presented in this section our proposed method for unlabeled hand activity clustering. The experiments based on real-world datasets show that the feature space learned using our *APML+MES* loss function allows meaningful clustering. We also confirmed that, in contrast to the state-of-art deep metric learning-based losses, our *APML* loss preserves a good recognition accuracy. We solved the clustering sensitivity problem using our SCC

strategy, allowing the best clustering result selection. For a fair comparison, we tested and compared our *APML+MES* with state-of-the-art methods on clustering and image-retrieval tasks on two commonly used real-world datasets. The proposed *APML+MES* loss has shown very effective in encouraging neural networks to learn a highly discriminative representation. Furthermore, experiments show that MES loss can improve performance when combined with existing metric learning loss functions.

In the next section, we exploit the resulting clustered activities by integrating them into the initial recognition and the unknown activity detection models. Thus, the desired user activity recognition framework incrementally learns and extends its multi-class classifier, making each new class "known" to the models.

4.4 Incremental hand activity recognition

The final component we propose aims to make the user activity recognition framework adaptable and expansible for possible future applications. To this end, the resulting clustered and annotated activities from the previous component are integrated into the initial recognition and the unknown activity detection models. In other words, this final component allows the framework to learn incrementally and extends its multi-class classifier, making each new class "known" to the models. Figure 4.22 highlights this model extension component of our desired user activity recognition framework.

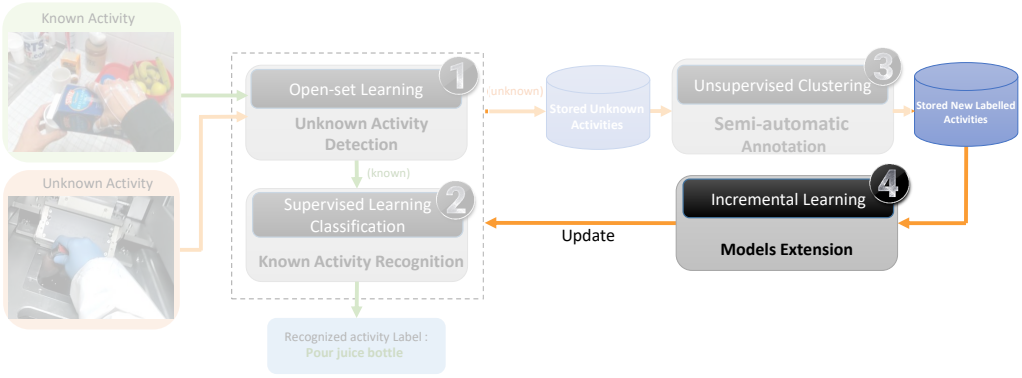

Fig. 4.22 Illustration of the targeted hand activity recognition Framework. We highlight the model extension step. This step aims to update the recognition and unknown activity detection models, allowing the Framework to incrementally recognize hand activities.

Based on the state-of-art study presented in section 2.5, in this section, we present our adopted method for incremental hand activity recognition. The proposed method is based on fine-tuning with replay memory. More precisely, we based our approach on iCaRL [156] method that we briefly introduced in chapter 2, subsection 2.5.1. As we concluded in chapter 2, section 2.5, the memory replay methods are the most accurate ones. Nevertheless, they require the storage of information about the old training samples, which can be very costly in terms of memory. On the other hand, pseudo-replay-based methods generate synthetic data to represent past classes in the current incremental state. However, in our case, generating synthetic hand activities is very difficult and requires a lot of data.

Attempting an incremental hand activity recognition solution that balances good recognition accuracy and reducing the memory cost, we highlight the contributions of our proposed method as follows:

- Unlike [156], we propose an efficient memory replay solution that aims at storing all training samples in a reduced size while keeping a discriminative representation of the original samples.
- We used our proposed *APML+MES* loss function instead of the *Softmax* loss, which showed more improvements.

The remainder of this section is organized as follows. In subsection 4.4.1, we present the initialization of the proposed incremental hand activity recognition. Subsection 4.4.2 details the progress of the incremental learning iterations. In subsection 4.4.3, we present the evaluation of the proposed approach. Thus, we detail our experiments and discuss the obtained results. Subsection 4.4.4, concludes the section.

4.4.1 Initialization of the incremental learning

Initialization

We first train a neural network to recognize hand activities to initialize incremental hand activity recognition. Following the mathematic formulation used in the previous sections, let denote by $g_w(f_\phi(.))$ the neural recognition network where $g_w(.)$ is the classification layer equipped with our proposed composite DML loss function (*APML+MES*) and $f_\phi(.)$ is the backbone spatial-temporal feature learner, with w and ϕ learnable parameters. The training procedure results in a pre-trained hand activity neural network model f_{ϕ^*} with ϕ^* learned parameter, which will be used next for fine-tuning-based incremental hand activity recognition. We used the RGB- and 3D skeleton-based neural network architectures proposed in chapter 3.

Our proposed memory-reply strategy

To prevent the *catastrophic forgetting* problem, we use the memory of the past. In contrast to iCaRL [156], which uses a selection of training samples (prototypes), we store all training samples with their associated ground-truth labels. Consequently, we use the spatial feature extractor blocks as low-dimensional manifolds to reduce activity sequences size, lowering the memory cost. Let us formally take the RGB-based hand activity recognition architecture (introduced in chapter 3, section 3.3) as our primary training network. Instead of storing activity videos of all training samples, we only store their associated Ψ_l and Ψ_r (please refer to Eq. 3.16 for more details). Similarly, when we use the 3D skeleton-based hand activity recognition architecture (introduced in chapter 3, section 3.2) as our primary training network, for each training sample, we only store its corresponding Ψ_{SoCJ}, Ψ_{IIFRD} and

Ψ_{GRT} introduced in subsection 3.2.1. Thus, we use only this reduced and highly pertinent hand activity representation in each incremental learning iteration. Figure 4.23 shows the illustration of the proposed representation of training data.

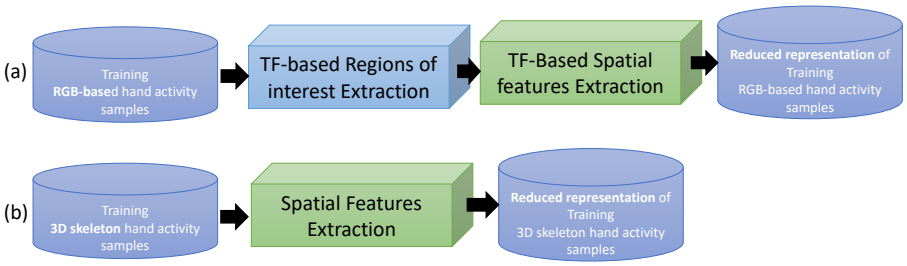

Fig. 4.23 Reduced and discriminative fixed representation of training data for fine-tuning incremental hand activity recognition. (a) when using the RGB-based hand activity method proposed in section 3.3. (b) when using the 3D skeleton-based hand activity method proposed in section 3.2.

4.4.2 Incremental learning progress

The adopted fine-tuning strategy

When a set of new annotated hand activities arrives, similarly to the training samples, we compute the spatial features for each newly arrived sample.

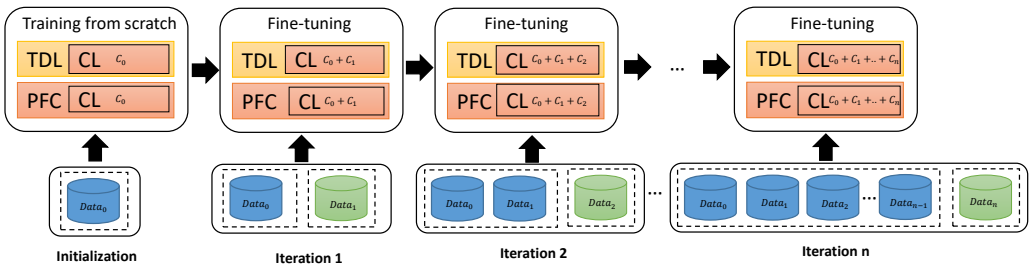

Fig. 4.24 Illustration of the incremental learning procedure. First, the procedure is initialized by training the neural network from scratch. Then, at each incremental learning iteration, the neural network architecture classification layers are modified by adding new neurons outputs according to the new classes. Finally, the network is fine-tuned using the combination of old and new data. Where C_i is the number of classes at the i-th iteration.

As illustrated in figure 4.24, to learn the newly arrived classes incrementally, we modify all the classification layers (CLs), these of the temporal dependencies learning (TDL) and

the one of the post-fusion classification (PFC). This is by adding the new classes to allow the classification of the old and new classes.

Next, we fine-tune the recognition networks using our *APML+MES* loss function and the distillation loss. The network parameters are updated by minimizing the *APML+MES* loss. For each set of new samples, the *APML+MES* encourages the network to output the correct class indicator for new classes (classification loss) and for old classes to reproduce the exact predictions as in the previous iteration (distillation loss).

$$\mathscr{L} = \mathscr{L}_{APLM+MES} + \mathscr{L}_{Distillation} \tag{4.20}$$

The fine-tuning involves training the LSTMs of the temporal dependencies learning and the MLP of the post-fusion classifier. Thus, to feed the LSTM networks, we use the reduced representation of samples from old classes with their predicted labels in combination with samples from new classes with their ground-truth labels. To overcome the *catastrophic forgting* problem, the fine-tuning we adopted for incremental learning consists of two enhancements:

- The training data comprises newly arrived and previously stored training samples. This ensures information about data distribution from all prior learned classes joins the training process. As we explained previously, old and new training samples are saved in low-dimensional representation generated with fixed spatial feature extraction functions, which are not affected by the change of weight parameters.

- The loss function is also enhanced. It includes the distillation loss, which ensures that the discriminative information learned previously is not lost during the new learning iteration, in addition to our *APML+MES* loss, which encourages improvements in feature representation that allow the classification of the newly arrived samples.

Classification of new hand activity samples

Inspired by [119], the authors of iCaRL[156] use a nearest-class-mean classifier (NCM) to classify samples. They first compute the mean vectors z_j^{mean} for each old training class based on selected mapped samples in feature space. This is exactly as we did in subsection 4.2.2 E.q 4.3. Then, they compute the feature vector of the sample that should be classified and assigns the class label with the most similar class mean vector. Let us take x^{new} as a new sample to be classified, its feature vector is computed by $f_\phi(x^{new}) = z^{new}$, then, its class assignment $\hat{y}_*$ is computed as follows:

$$\hat{y}_* = \underset{j=1:C}{\operatorname{argmin}} \left\| z^{new} - z_j^{mean} \right\| \tag{4.21}$$

As we briefly presented iCaRL[156] in chapter 2, subsection 2.5.1, their classification method aims to reduce the prediction bias caused by the imbalance between old and new classes. However, the classification decision can not be very efficient since it uses a selection of training prototypes and does not cover all training samples. In contrast, we used a size-reduced version of all training samples to enhance the classification decision. Instead of NCM, we use the output of our *APML* loss to assign the class label for a given new activity sample as follows:

$$\hat{y}_* = \underset{j=1:C}{\operatorname{argmin}} \left\{ \frac{(1 + \left\| z^{new} - w_j \right\|)^{-1}}{\sum_{j=1}^{C} (1 + \left\| z^{new} - w_j \right\|)^{-1}} \right\} \tag{4.22}$$

We justify this by the fact that when using our *APML+MES* loss during the training, the classification weight vectors w_j are not decoupled from the mapping function f_ϕ. Moreover, our *APML* -based classification decision is very similar to NCM in the way that the new sample will be assigned to the nearest class. This is done by measuring the distance between the feature vector associated with the new example and the classification weights (Eq 4.22). Thanks to our *MES* loss, the classification weights w_j they are considered the mean feature vector of each training class z_j^{mean}.

4.4.3 Evaluation of the adopted incremental learning method

Dataset

We used the **FPHA** [49] and the **SLS France** datasets to evaluate our approaches. First, we randomly partitioned the FPHA dataset into five partitions. The first four partitions contain ten classes, and the last one contains five classes. Each class contains an average of 26 samples, divided into 13 for the train and 13 for the test.

For a more realistic scenario, we adopted the three data configurations presented in subsection 4.2.7, table 4.1. For the two configurations (*Config1* and *Config2*), we used the partition of known classes for the initialization and the unknown classes for the first incremental learning iteration. For *Config3*, we used all classes of the FPHA dataset for the initialization and the SLS France classes as newly arrived data for the first incremental learning iteration. Thus, our adopted approach aims to robustly learn the old classes of the FPHA dataset and the new ones of the SLS France dataset.

Implementation details

We implemented three incremental hand activity recognition methods. We used our two proposed hand activity recognition neural network architectures as backbones for the three methods. We kept the exact implementation details for the 3D skeleton- and 3D-based architectures we presented in chapter 3 sections 3.2 and 3.3, respectively. We define the implementations as follows:

- **APML+MES.** For the first training step, we kept the same hyper-parameters for the two 3D skeleton- and 3D-based architectures as in chapter 3 sections 3.3 and 3.2, respectively. Except for the number of training epochs, which is fixed at 200. The networks are trained using our *APML+MES* loss function instead of the *Softmax*.

  At each incremental learning iteration, we fine-tune the network using previously learned weights in the previous iteration. We fine-tune the network with a small learning rate value to prevent the network from drastically changing the weights, causing the *catastrophic forgetting problem*.

  To classify new hand activity samples, we used the *APML* loss output.

- **APML+MES+NCM.** We use the same implementation. Except for the classification of new hand activity samples, we use the NCM as in iCaRL[156] instead of the *APML* outputs.

- **iCaRL.** We used the same implementation details and recommendations as in [156]. For the number of prototypes, we fixed it at 5 for the FPHA dataset since the average number of training samples per class is 13.

Results and discussion

We compared two variants of our proposed method for incremental learning hand activity recognition (*APML+MES* and *APML+MES+NCM*) with the state-of-the-art baseline iCaRL [156]. The comparison is done on the FPHA dataset, streaming data by sets of 10 classes at each incremental learning iteration and five classes for the final iteration. We performed eight random selections of the ten classes of streaming sets. Thus the entire incremental learning process is repeated eight times, and then we computed the average accuracy result.

Figures 4.25 (a) and (b) show that our method outperforms all the methods when using the NCM with all training samples for classification. The NCM is shown to be better than the *APML* for classification. This can be explained by the fact that the mean feature vectors computed by the NCM are more precise than classification weight centers optimized during

the training procedure using our *APML+MES* loss function. The iCaRL [156] method performs very poorly. Their strategy of using a given number of selected prototypes may be very effective in the case of using large datasets, as reported in their paper [156]. However, it performs poorly in our case since the dataset does not contain many training samples. Moreover, based on their strategy of keeping the original form to store selected training samples without any dimensionality reduction, storing all training samples may be very expensive in terms of memory.

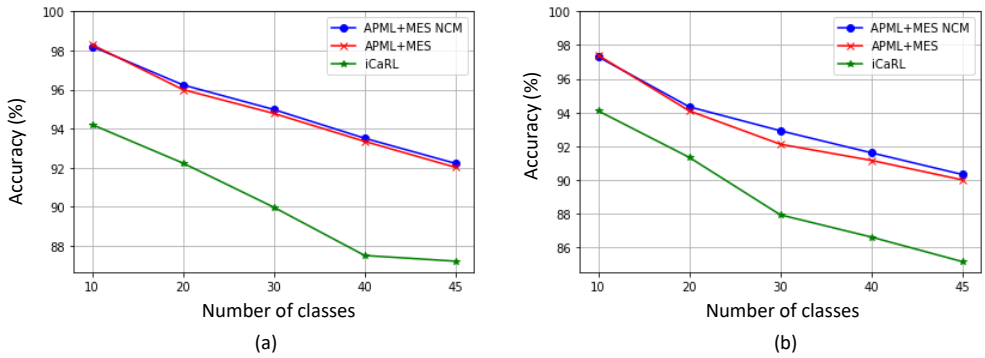

Fig. 4.25 Incremental hand activity recognition accuracy results. (a) Using the RGB-based spatial-temporal backbone. (b) Using the 3D-based spatial-temporal backbone.

For a more realistic experiment, we tested our method on the FPHA dataset based on the two configurations (*Config1* and *Config2*). We initialized the incremental leaning with data of the known classes, (*"kitchen" +"office"*) for the *Config1* and (*"kitchen"+"social"*) for the *Config2*. Once the incremental learning is initialized, we use the data of the unknown classes as newly arrived classes of the first incremental learning iteration. The classes of the *"social"* scenario for *Config1*, and the classes of the *"office"* scenario for *Config2*, respectively. We note that for this experiment, we used only RGB image sequences.

Thus, in the first iteration, the network is fine-tuned with known and unknown classes. Table 4.10 shows the results using our method based on the *APML+MES* loss for training and fine-tuning, while the NCM is used for classifying new samples. We can see that the accuracy between the initialization and the first iteration is very close. A difference of 0.66% in the *Config1* and 0.43% in the *Config2*. Moreover, we achieve good accuracy results for the entire FPHA dataset classes (*"kitchen" +"office"+ "social"*), 95% and 95.70% for *Config1* and *Config2*, respectively. We note that these results are too close to the results presented in chapter 3, section 3.3, where the network is trained from scratch, directly on the entire FPHA dataset classes (*"kitchen" +"office"+ "social"*).

Table 4.10 Accuracy results for the initialization and the first incremental learning iteration.

	Config 1			Config 2		
	kitchen+office acc (%)	**social acc (%)**	**kitchen+office +social acc (%)**	**kitchen+social acc (%)**	**office acc (%)**	**kitchen+social +office acc (%)**
Initialization	92.55	-	-	95.21	-	-
First iteration	91.89	98.12	95.00	94.78	96.63	95.705

Similarly, in another experiment, we used all the classes of the FPHA data set for the initialization and all the classes of the SLS France dataset as newly arrived data for the first incremental learning iteration. Here again, from table 4.11, we observe that the accuracy between the classification in the initialization step and the first iteration is very close. This confirms that our method handles the *catastrophic forgetting* problem.

Table 4.11 Accuracy results for the initialization and the first incremental learning iteration. The classes of the FPHA dataset are used for the initialization. Then, the SLS France dataset classes are used for the first incremental learning iteration.

	FPHA dataset acc (%)	**SLS France dataset acc (%)**	**FPHA+SLS France datasets acc (%)**
Initialization	96.45	-	-
First iteration	94.02	62.74	78.38

4.4.4 Conclusion

In this section, we presented our adopted approach addressing the last proposed component of our desired user activity recognition framework. This last component aims at making hand activity recognition models extendable and continuous. The presented approach uses the memory of the past to overcome the problem of *catastrophic forgetting*. Thus, we proposed an efficient solution to store learned samples as a memory of the past, which balances the memory cost and the highly discriminative data representation. The experiments performed on two real-world hand activity datasets confirmed that our method handles the *catastrophic forgetting* problem very well. Particularly when using our proposed *APML+MES* loss function for fine-tuning the recognition model.

Chapter 5

Conclusions and Perspectives

Contents

5.1 Conclusions

This thesis explores first-person hand activity recognition, proposing four main components for a comprehensive framework that understands the user's hand activity to assist them in an augmented reality context. The proposed components allow the framework to robustly recognizes hand activities from known classes observed during the training procedure based on the first-person viewpoint. Moreover, these proposed components allow the detection of activities from unknown classes unseen during the training procedure, a fundamental requirement in teaching and user assistance use cases. The user activity recognition framework learns in an open-set setting to identify unknown activities. In this realistic and challenging open-set scenario, incomplete knowledge of the world exists at training time, and unknown activities can be seen during testing. These detected unknown activities are collected, automatically annotated, and then incorporated into the models. In this way, the proposed framework learns and gradually expands by making each new activity known to the learned models. Much of the work was also devoted to minimizing computation and data acquisition costs, resulting in cost-effective and easily adaptable components.

In this regard, we began by reviewing state-of-the-art existing first-person hand activity recognition methods, which we divided into three distinct categories. We provided a description, advantages, and limitations for each category; and presented the available state-of-the-art datasets for hand activity recognition evaluation. We also gave an overview of the state-of-the-art methods related to our contributions. Thus, we explored the following research fields:

- **Open-set recognition.** To detect unknown hand activities while correctly classifying the known ones.
- **Unsupervised domain adaptation and metric learning.** To automatically cluster, then manually annotate detected unknown hand activities.
- **Incremental learning.** To integrate annotated unknown activities into the recognition and unknown activity detection models.

Motivated by the state-of-the-art observations, we presented in this thesis our two approaches for first-person hand activity recognition: First, we detailed our proposed method for 3D skeleton-based hand activity recognition. Then, we introduced our method for RGB-based and the combination of RGB- and 3D skeleton-based method for hand activity recognition. We gave extensive experiments that evaluated and validated each proposed method by discussing and comparing the obtained results with existing state-of-the-art methods. The experiments have proven the effectiveness of our proposed methods on real-world

datasets. They also showed the advantages and disadvantages of the RGB images and 3D hand skeleton modalities on hand activity recognition. We also experimented with combining the two modalities for hand activity recognition, which significantly improved the recognition accuracy.

To give the user activity recognition framework the ability to perform open-set recognition, we presented our adopted consensus-based open-set recognition component that groups three approaches to deciding whether an activity is from a known or unknown class. The method is compared with state-of-the-art methods on two real-world datasets. We showed its benefit for efficient open-set hand activity recognition by presenting and discussing the experimental results.

In order to exploit the detected unknown hand activities, we introduced our proposed unlabeled hand activity clustering method. The experiments based on real-world datasets show that the feature space learned using our *APML+MES* composite loss function allows meaningful clustering. We also confirmed that, in contrast to the state-of-art deep metric learning-based losses, our *APML* loss preserves a good recognition accuracy on labeled activities. We solved the clustering sensitivity problem using our statistical and consensus clustering strategy, allowing the best clustering result selection. For a fair comparison, we tested and compared our *APML+MES* loss with state-of-the-art losses on clustering and image-retrieval tasks on two commonly used real-world datasets. The proposed *APML+MES* loss has shown very effective in encouraging neural networks to lean a highly discriminative representation. Furthermore, experiments show that *MES* loss can improve performance when combined with existing state-of-the-art metric learning loss functions.

Finally, we presented our adopted approach for incremental hand activity recognition component to make the user activity recognition framework expansible and flexible. The goal is to make the recognition models extendable and continuously learn newly arrived hand activities. The presented approach uses the memory of the past to overcome the problem of *catastrophic forgetting*. Thus, we proposed an efficient solution to store learned samples as a memory of the past, which balances the memory cost and the highly discriminative data representation. The experiments performed on two real-world hand activity datasets confirmed that our method handles the *catastrophic forgetting* problem. Particularly when using our proposed *APML+MES* loss function for fine-tuning the recognition models.

5.2 Limitations and future works

In this section, we briefly describe all the approaches proposed in this thesis, followed by the main limitations and possible future improvements.

5.2.1 Hand activity recognition

3D skeleton-based hand activity recognition. It is a novel learning pipeline for first-person hand activity recognition. The proposed pipeline is composed of three blocks. The first block is a new combination of hand-crafted feature extraction methods. The second block is our multi-stream temporal dependencies learning strategy. In the last block, we introduced our proposed post-fusion strategy, which has proven to be more efficient than other traditional fusion methods. During our experiments, we observed the following main limitations:

- Inter-class confusion occurs when different classes use objects of the same shape but different colors, e.g., *"open juice bottle"* and *"open milk bottle"*. The usage of RGB images can surpass this limitation since they provide color information for manipulated objects.

- We used 3D hand-crafted spatial-feature extraction methods to overcome the data scarcity problem. However, the hand-crafted proposed methods are problem-specific. Thus, they perform poorly for other tasks, e.g., hand gesture recognition. This limitation can be solved by using a learnable and adaptable 3D hand skeleton spatial-feature extractor. It must provide discriminative spatial-features while learning on a limited amount of data. This may be the best solution that can replace our hand-crafted spatial-feature methods.

RGB-based hand activity recognition. A novel learning pipeline for first-person hand activity recognition. The proposed pipeline is composed of four stages. In the first stage, we presented our transfer learning-based regions of interest extraction, the left and right hands regions, which have proven effective. The second stage is the transfer learning-based deep spatial feature extraction method that exploits the regions of interest in visual information. To manage the temporal dimension, in the third stage, we trained the temporal neural networks in a multi-stream manner. Then, in the last stage, we applied the post-fusion strategy to classify activities. In the following, we highlight the limitations we observed during our experiments. We also provide possible future improvements:

- Inter-class confusion that can occur while using RGB-based methods is when the handled object in different classes has the same color. However, the shape is different.

The 3D hand skeleton data usage can help overcome this limitation since the handled object's shape can be driven from the 3D hand skeleton coordinates.

- We assign the hands as left or right based on their position in the image. However, the hands are not always in a suitable position, which results in bad assignments.

  This can be surpassed by using a learned model that can distinguish between the user's right and left hands.

- The extracted hands regions contain information about the background, which may confuse the neural network.

  This can be solved by using a learned model that segments only the hands and the handled object based on their contour.

The proposed methods perform hand activity recognition in offline mode. However, the targeted augmented reality use case may require an online activity recognition solution to identify the start and end of the activity.

5.2.2 Open-set hand activity recognition

We introduced our adopted consensus-based open-set hand activity recognition. It groups three approaches to deciding whether a test activity sample is from a known or unknown class. We employed a consensus of three outlier detection approaches to increase open-set recognition performance: The Weibull model, the Isolation Forest, and the Local Outlier Factor methods. We aggregated results from the three approaches to deciding whether the activity is from a known or unknown class via majority voting. In the following, we introduce the limitations we observed during our experiments:

- One of the main limitations is related to the Weibull-based approach since it requires two hyper-parameters, which are essential to build the model and make decisions. Mainly for the fixed ε threshold, since a small value of the ε, the model will reject samples even if they had a small chance of being known. A possible solution is to design a model that automatically learns and predicts these parameters from the training data.

5.2.3 Unlabeled hand activity clustering

We proposed a novel approach for unlabeled hand activity clustering. The proposed approach follows the UDA paradigm. First, we pre-train a supervised neural network on labeled

samples from the source domain. Next, we try to solve the UDA by using the pre-trained neural network model feature space as a mapping manifold to cluster unlabeled target domain samples based on classical clustering methods. The pre-trained model, driven from the source domain, must map hand activities to a highly discriminative feature space. Thus, we proposed a composite metric learning loss function (*APML+MES*) that forces the neural network to learn a highly discriminative feature space. In the following, we introduce the limitations we observed during our experiments. We also provide possible future improvements:

- In our experiments, we observed a high sensitivity of clustering algorithms to the randomness present in the pre-training procedure. It is directly related to the random initialization of weights, the dropout layers, and the optimizer. To this end, we proposed a statistical and consensus clustering (SCC) strategy. It consists in repeating the pre-training and the clustering steps sequentially several times. Thus, in the end, a consensus clustering is prefomed on resulting clustering predictions to provide the final one. However, performing a multiple-run of the pre-training and clustering steps can be computationally highly expensive, especially for large amounts of data and/or deep neural network architectures. As future improvements, we plan to design a neural network architecture that provides a clustering-friendly feature space while stabilizing the clustering sensitivity.
- Our *MES* loss function has proven its effectiveness in boosting the discriminative power of neural networks. This is when it is combined with our *APML* loss function or other existing state-of-the-art metric learning loss functions. However, our actual design of the *MES* loss requires the computation of all mutually possible Euclidean distances between centers of classes (weights vectors of the final classification layer). Unfortunately, this can be very costly in computational time for datasets with a large number of classes. An optimized solution that enforces the mutually equidistant separation of class centers is necessary.

5.2.4 Incremental hand activity recognition

We presented our adopted method for incremental hand activity recognition. The proposed method is based on fine-tuning with replay memory. Thus, we proposed a solution to store learned samples as a memory of the past, which balances the memory cost and the highly discriminative data representation. The fine-tuning procedure is performed using our proposed *APML+MES* loss function, which showed a significant improvement. In the following, we highlight the limitations we observed during our experiments. We also provide possible future improvements:

- To prevent the *catastrophic forgetting* problem, we used the memory of the past. We stored all training samples with their associated ground-truth labels. Instead of storing activity videos of all training samples, we only store their associated extracted spatial-temporal features. This feature extraction process reduces the size of the data a little bit, but not enough to allow long-term incremental learning.

- In each fine-tuning iteration, we relearn all network weights, but we keep the network architecture unchanged. However, an additional number of neurons may be required with the increasing number of classes and associated training samples. To solve this problem, we could rely on the concept of dynamic networks introduced in chapter 2, section 2.5.3.

Résumé en français

Introduction

Le développement de l'Internet des objets a permis d'utiliser les technologies numériques aussi bien dans des contextes quotidiens que dans l'univers d'industrie 4.0. Avec la sortie de dispositifs de visualisation de la réalité augmentée (RA) plus compacts, plus puissants et plus légers, les entreprises industrielles réinvestissent dans la RA et explorent son adéquation au contexte industriel. Pour réduire les coûts et les temps de production, les fabricants cherchent des solutions aux problèmes liés à l'interruption des machines de production, par exemple pour des raisons de maintenance ou de spécifications de certains produits. La formation des opérateurs est une partie du processus de production où la RA peut être intégré pour gagner du temps et de l'argent. Une formation appropriée et adéquate des opérateurs industriels peut être moins coûteuse qu'une reconfiguration de l'ensemble du processus de production. En raison de leur flexibilité, les opérateurs peuvent répondre assez facilement et rapidement aux besoins particuliers du fabricant. Leur formation nécessite souvent la présence d'un autre opérateur enseignant suffisamment expérimenté pour transmettre les compétences requises. Dans le cas où une telle personne n'est pas disponible, des manuels d'instruction peuvent être utilisés pour former les opérateurs. Cependant, ils peuvent parfois être plus lourds, dépassés et insuffisamment informatifs. Ces manuels peuvent ne pas répondre aux spécificités d'un apprentissage ni au niveau technique des apprentis. Pour être efficace, complète et durable, la procédure de formation dans un contexte industriel fait appel à nos différents sens, notamment la vue, l'ouïe et le toucher. La meilleure façon d'apprendre de manière appropriée et d'acquérir efficacement de nouvelles connaissances est de visualiser la tâche à accomplir et d'en faire l'expérience. La RA peut répondre à ces exigences en affichant des informations pertinentes au bon moment et au bon endroit.

D'autre part, l'assemblage est crucial pour l'ensemble du processus de fabrication. Le coût total d'un produit, le temps nécessaire à sa fabrication et sa qualité dépendent de l'efficacité et de la précision des différentes étapes de l'assemblage. Les opérations d'assemblage sont souvent complexes et nécessitent des ajustements fins pour obtenir un

résultat acceptable. La séquence d'assemblage peut être longue, avec de nombreuses pièces à assembler dans un ordre précis pour assurer le bon fonctionnement du produit. Pour ces raisons, les travailleurs assembleurs doivent être qualifiés et formés pour le faire dans le temps de cycle imposé par la cadence de production. Le produit final peut dépendre de la variante à assembler et peut nécessiter la consultation de manuel papier. Ces tableaux de référence peuvent entraîner des pertes de temps, des distractions et des problèmes de sécurité. Dans l'assemblage manuel, les tâches sont effectuées par des opérateurs humains assistés d'outils ou de machines semi-automatiques. Cependant, l'erreur humaine est également un problème fondamental dans les chaînes de montage. Elles peuvent entraîner une augmentation des déchets de production ou du temps et des coûts de traitement, ainsi qu'une dégradation de la qualité des produits en raison de défauts de fabrication. Diverses méthodes, telles que des sessions de formation intensive ou des dispositifs de détection, sont utilisées pour surmonter ce problème. Ces approches sont souvent coûteuses et, dans de nombreux cas, ne donnent pas l'assurance complète d'éviter ces désagréments. La probabilité d'une erreur humaine pendant l'assemblage doit être réduit afin d'éviter tout dommage potentiel à l'ensemble du système de production. À cette fin, la nécessité de combiner l'assemblage manuel traditionnel avec un outil capable d'améliorer l'efficacité et l'efficience du processus, tel qu'un système de Réalité Augmentée (RA), devient évidente. L'opérateur doit être soutenu et guidé dans ses activités, par exemple en effectuant les opérations d'assemblage tout en étant équipé d'un système assistant intelligent de RA. Un tel système de RA peut empêcher l'opérateur de détourner son attention et de ne pas être distrait du processus, ce qui contribue à l'efficacité des tâches.

Un système de RA intelligent peut être particulièrement bien adapté pour résoudre ces problèmes. À cet égard, la compréhension des activités de l'opérateur d'un point de vue à la première personne est fondamentale dans un tel système de RA. La compréhension des activités réalisées dans un contexte de RA permet au travailleur d'interagir avec l'environnement naturel et les informations virtuelles tout en étant guidé et en recevant un retour d'information. Cela peut donner la possibilité de remplacer l'enseignement traditionnel par des instructions interactives basées sur la RA et les manuels papier par une assistance basée sur la RA. Motivée par toutes ces observations, cette thèse se concentre sur la compréhension de l'activité de la main à la première personne et les défis qui y sont liés. La section suivante donne plus de détails sur les objectifs de la thèse.

Objectifs

Pour résoudre les problèmes abordés dans la section précédente, ce travail de thèse a été consacré à la conception des principaux composants d'un système complet qui reconnaît les

activités des utilisateurs de RA afin de les assister dans leurs activités complexes. Le système de reconnaissance des activités de l'utilisateur souhaité doit reconnaître de manière robuste les activités habituelles basées sur le point de vue de la première personne. En outre, il doit détecter les activités inhabituelles afin d'empêcher l'utilisateur d'effectuer de mauvaises manœuvres, une exigence fondamentale dans les cas d'utilisation de l'enseignement et de l'assistance aux utilisateurs.

À cette fin, nous avons basé nos recherches sur des approches d'apprentissage automatique et de vision par ordinateur, un choix recommandés de l'état de l'art. Une grande partie du travail a été consacrée à la minimisation des coûts de calcul et d'acquisition de données, ce qui a permis de créer des composants peu coûteux et facilement adaptables. Le système de reconnaissance de l'activité de l'utilisateur devrait permettre d'être appris sur une quantité limitée de données, par exemple, il peut d'abord être entraîné sur un ensemble de données disponibles publiquement, puis rapidement adapté à un autre cas d'usage privé, comme une application industrielle. Cela permet de réduire considérablement le coût d'acquisition des données annotées et d'étendre la marge d'applications pour couvrir différents domaines industriels.

Le système souhaité est constitué de quatre composants principaux. Dans la première composante, l'activité est vérifiée si elle est connue ou inconnue. Ainsi, si l'activité est identifiée comme connue, l'étiquette de classe associée sera donnée par le deuxième composant qui classifie les activités connues. Sinon, si l'activité est inconnue, elle sera stockée. Une fois qu'un certain nombre d'activités inconnues détectées est atteint, elles seront annotées de manière semi-automatique dans le troisième composant. Enfin, les activités inconnues annotées seront intégrées au modèle de reconnaissance dans le dernier composant. Grâce à ce dernier composant, les activités inconnues annotées seront traitées comme des activités connues et classées dans le futur.

État de l'Art

L'objectif principal de cette thèse est de proposer les composants essentiels d'un système de reconnaissance d'activité de la main, qui peut être utilisé pour assister les utilisateurs de RA. Le premier composant vise à détecter les activités inconnues. Le deuxième composant vise à reconnaître les activités de la main du point de vue de la première personne. Le troisième composant regroupe et annote les activités inconnues détectées. Enfin, le dernier composant intègre ces activités annotées dans les modèles de reconnaissance et de détection.

Chacune des quatre composants visés par nos objectifs peut être considérée comme une étape de reconnaissance en monde ouvert (open-world recognition) [12], représentant

un domaine de recherche complexe et étendu : apprentissage en milieu ouvert (open-set learning), classification d'activité de la main par apprentissage supervisé, regroupement non supervisé et apprentissage incrémental. C'est pourquoi, nous décrivons l'idée générale de chaque composant et présentons les approches de pointe existantes qui s'y rapportent.

Contributions

Dans ce travail, nous avons essayé de surmonter tous les défis mentionnés dans la section précédente. Cela nous a conduit à une proposition dont les principales contributions sont résumées ci-dessous :

- **Solutions de reconnaissance d'ctivités de la main.** Pour relever les défis de la reconnaissance de l'activité de la main, l'un des composants les plus importants du système ciblé, nous avons proposé deux solutions pour la reconnaissance de l'activité de la main à la première personne. En plus de la robustesse de la reconnaissance, nous avons également pris en compte la manque des données et le coût de calcul. Les solutions proposées sont basées sur des données squelettiques 3D et des images RGB, respectivement :

  1. Un nouveau pipeline d'apprentissage hybride pour la reconnaissance de l'activité de la main basée sur le squelette 3D de la main, qui se compose de trois blocs. Tout d'abord, les caractéristiques spatiales pour une séquence donnée de positions d'articulations de la main sont rapidement extraites en utilisant une combinaison spécifique de nos méthodes d'extraction de caractéristiques spatiales locales et globale qu'on propose. Ensuite, les dépendances temporelles sont apprises en utilisant une stratégie d'apprentissage multi-flux. Enfin, un classificateur de séquence d'activité manuelle est appris en utilisant notre stratégie de post-fusion et appliqué aux dépendances temporelles précédemment apprises. Ce pipeline d'apprentissage en plusieurs étapes permet de s'entraîner avec un nombre limité d'échantillons d'entraînement tout en assurant une bonne précision, ce qui répond au problème du manque des données. Les expériences évaluées sur deux ensembles de données du monde réel montrent que notre approche est plus performante que l'état de l'art. Pour une étude d'ablation, nous avons comparé notre stratégie de post-fusion avec trois méthodes classiques de fusion traditionnelles et avons montré une amélioration de la précision.
  2. Un nouveau pipeline d'apprentissage par transfert à plusieurs étapes pour la reconnaissance d'activité de la main de la première personne basées sur des

séquence d'image RGB, qui aborde le problème de la rareté des données. La première étape extrait les régions d'intérêt pour une séquence d'activité d'image RVB donnée en utilisant un réseau neuronal pré-entraîné. Contrairement aux méthodes existantes qui utilisent l'attention visuelle par le biais du Deep Learning et nécessitent une grande quantité de données, nous proposons d'utiliser directement les mains droite et gauche comme régions d'intérêt pertinentes qui fournissent des informations sur les objets manipulés et les actions effectuées. Ces régions d'intérêt sont extraites en utilisant une technique d'apprentissage par transfert. Nos expériences ont montré que ces informations sont essentielles à la reconnaissance de l'activité des mains à la première personne. Nous proposons une procédure d'augmentation des données adaptée à ces régions d'intérêt pour renforcer le modèle de reconnaissance. Ensuite, des caractéristiques spatiales de haut niveau sont extraites dans la deuxième étape à l'aide d'un réseau neuronal profond pré-entraîné. Dans la troisième étape, les dépendances temporelles sont apprises. Enfin, un classificateur de séquence d'activité manuelle est appris dans la dernière étape en appliquant une stratégie de post-fusion aux dépendances temporelles précédemment apprises.

L'adaptation de l'apprentissage par transfert permet d'apprendre avec un nombre limité d'exemples d'entrainement tout en offrant une bonne précision. Il réduit également le coût d'apprentissage puisque le réseau neuronal transféré est déjà pré-entrainé.

Les expériences évaluées sur deux ensembles de données du monde réel montrent que notre pipeline atteint des performances de pointe. De plus, le pipeline proposé obtient de bons résultats même avec une quantité de données limitées.

Nous avons également expérimenté la combinaison des deux modalités, le squelette 3D des articulations de la main et les pipelines basés sur les images RGB, ce qui améliore considérablement la précision de la reconnaissance de l'activité de la main.

- **Détection d'activité de la main inconnue.** Les solutions présentées ci-dessus pour la reconnaissance de l'activité des mains sont basées sur le paradigme classique de la reconnaissance en ensemble fermé (closed-set learning), où les échantillons d'activité de d'entrainement et de teste sont supposés faire partie des classes connues. Cependant, l'un des composants de notre système de reconnaissance de l'activité de l'utilisateur vise à effectuer la reconnaissance dans un ensemble ouvert (open-set learning). Ainsi, il reconnaît les activités des classes connues tout en détectant et en rejetant les activités inconnues des classes inconnues qui n'ont pas été vues pendant

la procédure d'apprentissage. À cet égard, nous avons présenté dans cette thèse une méthode de reconnaissance d'activité de la main dans un ensemble ouvert, basée sur un consensus qui regroupe trois approches pour décider si un échantillon d'activité de teste est d'une classe connue ou inconnue. Afin d'augmenter la performance globale de la reconnaissance dans un ensemble ouvert, nous utilisons un consensus de trois approches de détection des aberrations et agrégeons leurs décisions par le biais d'un vote.

- **Regroupement des activités de la main non étiquetées.** La troisième composante de notre système souhaité vise à regrouper (catégoriser) les activités inconnues détectées afin de les annoter. À cet égard, nous avons proposé une approche nouvelle et originale qui se rapproche de l'adaptation non supervisée du domaine pour regrouper les activités de la main non étiquetées. Elle utilise les connaissances obtenues à partir d'échantillons étiquetés du domaine source (les activités connues) pour catégoriser les échantillons non étiquetés du domaine cible (les activités inconnues détectées). Ainsi, nous avons introduit une fonction de perte composée, que nous notons *APML+MES*. C'est une fonction de perte nouvelle et originale basée sur l'apprentissage par métrique pour apprendre une représentation hautement discriminative tout en maintenant une bonne précision de reconnaissance des activités dans le domaine source. La représentation apprise est utilisée comme un espace de dimension réduite pour regrouper les échantillons d'activités non étiquetées. Pour obtenir les meilleurs résultats de regroupement, nous avons également proposé une stratégie de regroupement statistique et consensuelle. Pour une comparaison équitable, nous avons testé et comparé notre perte *APML+MES* avec des méthodes de pointe pour des tâches de regroupement et de recherche d'images sur deux bases de données réels couramment utilisés. La fonction perte *APML+MES* proposée s'est avérée très efficace pour encourager les réseaux neuronaux à apprendre une représentation hautement discriminante. En outre, les expériences montrent que la perte MES peut améliorer les performances lorsqu'elle est combinée avec les fonctions de perte d'apprentissage métrique existantes.

- **Reconnaissance incrémentale de l'activité de la main.** La dernière composante du système de reconnaissance d'activité souhaité vise à rendre les modèles appris extensibles et adaptables à de futures applications. À cette fin, les activités regroupées et annotées résultant du composant précédent sont intégrées dans les modèles initiaux de la reconnaissance de détection des activités inconnues. Ainsi, avec ce composant, le système apprend et étend de manière incrémentielle son classificatrice multi-classe, en faisant en sorte que chaque nouvelle classe soit "connue". À cette fin, nous présentons

dans cette thèse la méthode que nous avons adoptée pour la reconnaissance incrémentale de l'activité des mains. La méthode proposée est basée sur un réglage fin avec une méthode efficace de stockage des échantillons d'entrainement initiaux pour éviter le problème de l'*oubli catastrophique*.

Conclusion

Cette thèse explore la reconnaissance de l'activité de la main à la première personne, en proposant quatre composants principaux pour un système complet qui comprend l'activité de la main de l'utilisateur pour l'assister dans un contexte de réalité augmentée. Les composants proposés permettent au système de reconnaître de manière robuste les activités de la main à partir des classes connues observées pendant la procédure d'entraînement basée sur le point de vue de la première personne. De plus, ces composants proposés permettent la détection d'activités de classes inconnues non observées pendant la procédure d'apprentissage, une exigence fondamentale dans les cas d'utilisation d'enseignement et d'assistance aux utilisateurs. Le système de reconnaissance des activités de l'utilisateur apprend à identifier des activités inconnues dans un environnement ouvert. Dans ce scénario réaliste, la connaissance du monde est incomplète au moment de l'apprentissage, et des activités inconnues peuvent être observées pendant le teste. Ces activités inconnues détectées sont collectées, annotées automatiquement, puis incorporées dans les modèles. De cette façon, le cadre proposé apprend et s'étend progressivement en faisant connaître chaque nouvelle activité aux modèles appris. Une grande partie du travail a également été consacrée à la minimisation des coûts de calcul et d'acquisition de données, ce qui permet d'obtenir des composants efficace et facilement adaptables.

À cet égard, nous avons commencé par examiner l'état de l'art des méthodes existantes de reconnaissance d'activité de la main à la première personne, que nous avons divisées en trois catégories distinctes. Nous avons fourni une description, les avantages et les limites de chaque catégorie et présenté les bases de données disponibles pour l'évaluation de la reconnaissance de l'activité de la main. Nous avons également donné un aperçu de l'état de l'art des méthodes liées à nos contributions. Ainsi, nous avons exploré les domaines de recherche suivants :

- **Apprentissage dans un ensemble ouvert (open-set learning).** Détecter les activités inconnues de la main tout en classant correctement les activités connues.

- **Regroupement non supervisé du domaine et apprentissage par métrique.** Pour regrouper automatiquement, puis annoter manuellement les activités manuelles inconnues détectées.

- **Apprentissage incrémental.** Intégration des activités inconnues annotées dans les modèles de reconnaissance et de détection des activités inconnues.

Motivés par les observations de l'état de l'art, dans cette thèse, nous avons commencé par l'introduction de nos deux approches pour la reconnaissance d'activité de la main à la première personne : Tout d'abord, nous avons détaillé notre méthode proposée pour la reconnaissance de l'activité de la main basée sur le squelette 3D. Ensuite, nous avons présenté notre méthode de reconnaissance de l'activité de la main basée sur des images RGB et la combinaison de la méthode basée sur des images RGB et le squelette 3D. Nous avons présenté des expériences approfondies qui ont évalué et validé chaque méthode proposée en discutant et en comparant les résultats obtenus avec les méthodes existantes de l'état de l'art. Les expériences ont prouvé l'efficacité de nos méthodes proposées sur des bases de données du monde réel. Elles ont également montré les avantages et les inconvénients des images RGB et des modalités du squelette 3D de la main pour la reconnaissance de l'activité de la main. Nous avons également expérimenté la combinaison des deux modalités pour la reconnaissance de l'activité de la main, ce qui a permis d'améliorer considérablement la précision de la reconnaissance.

Pour donner au système de reconnaissance de l'activité de l'utilisateur la capacité d'effectuer la reconnaissance d'ensembles ouverts, nous avons présenté le composant de reconnaissance d'ensembles ouverts basé sur le consensus que nous avons adopté et qui regroupe trois approches pour décider si une activité appartient à une classe connue ou inconnue. La méthode est comparée à des méthodes de l'état de l'art sur deux bases de données du monde réel. Nous avons démontré son avantage pour une reconnaissance efficace de l'activité de la main dans un ensemble ouvert en présentant et en discutant les résultats expérimentaux.

Afin d'exploiter les activités de la main inconnues détectées, nous avons présenté notre méthode de regroupement des activités de la main non étiquetées. Les expériences basées sur des ensembles de données du monde réel montrent que l'espace des caractéristiques appris en utilisant notre fonction de perte composite *APML+MES* permet un regroupement significatif. Nous avons également confirmé que, contrairement aux fonctions de pertes basées sur l'apprentissage par métrique profond de l'état de l'art, notre fonction de perte *APML* préserve une bonne précision de reconnaissance sur les activités étiquetées. Nous avons résolu le problème de la sensibilité du regroupement en utilisant notre stratégie de regroupement

statistique et consensuel, permettant la sélection du meilleur résultat de regroupement. Pour une comparaison équitable, nous avons testé et comparé notre fonction de perte *APML+MES* avec les les fonctions de pertes de l'état de l'art sur des tâches de regroupement et de recherche d'images sur deux bases de données réelles couramment utilisées. La fonction de perte *APML+MES* proposée s'est avérée très efficace pour encourager les réseaux neuronaux à s'appuyer sur une représentation hautement discriminante. En outre, les expériences montrent que la perte *MES* peut améliorer les performances lorsqu'elle est combinée avec les fonctions de perte d'apprentissage métrique de existantes.

Enfin, nous avons présenté l'approche que nous avons adoptée pour le composant d'apprentissage incrémental de reconnaissance de l'activité des mains afin de rendre le système de reconnaissance de l'activité des utilisateurs extensible et flexible. L'objectif est de rendre les modèles de reconnaissance extensibles et d'apprendre continuellement des nouvelles activités de la main. L'approche présentée utilise la mémoire du passé pour surmonter le problème de l'*oubli catastrophique*. Nous avons donc proposé une solution efficace pour stocker les échantillons appris en tant que mémoire du passé, ce qui permet d'équilibrer le coût de la mémoire et la représentation hautement discriminante des données. Les expériences réalisées sur deux ensembles de données réelles sur l'activité des mains ont confirmé que notre méthode permet de résoudre le problème de l'(*oubli catastrophique*). En particulier lorsque nous utilisons la fonction de perte *APML+MES* que nous proposons pour affiner les modèles de reconnaissance.

Publications

International Journal Papers

- **Boutaleb, Y.**; Soladie, C.; Kacete, A.; Duong, N.; Royan, J. and Seguier, R., *MES loss: Mutually Equidistant Separation Metric Learning Loss Function.* Submitted to Pattern Recognition Letters journal.

- **Boutaleb, Y.**; Soladie, C.; Royan, J. and Seguier, R., *Principal Component for a Hand Activity Recognition Framework.* To be submitted.

International Conference Papers

- **Boutaleb, Y.**; Soladie, C.; Duong, N.; Kacete, A.; Royan, J. and Seguier, R., *Efficient Multi-stream Temporal Learning and Post-fusion Strategy for 3D Skeleton-based Hand Activity Recognition.* In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) - 2021 (oral).

- **Boutaleb, Y.**; Soladie, C.; Duong, N.; Royan, J. and Seguier, R., *Multi-stage RGB-based Transfer Learning Pipeline for Hand Activity Recognition.* In Proceedings of 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) - 2022 (oral).

- **Boutaleb, Y.**; Soladie, C.; Kacete, A.; Duong, N.; Royan, J. and Seguier, R. *Metric Learning-based Unsupervised Domain Adaptation for 3D Skeleton Hand Activities Categorization*. In Image Analysis and Processing (ICIAP) - 2022 (oral).

National Communications

- **Boutaleb, Y.**; Soladie, Kacete, A.; C.; Duong, N.; Royan, J. and Seguier, R. *Regroupement d'Activités de la Main Non-étiquetées*. GRETSI, Nancy, France, 2022.

Patents

- **Boutaleb, Y.**; Soladie, C.; Duong, N.; *Procédé d'apprentissage, procédé de reconnaissance associé, dispositifs correspondants*. Patent ID2020-002PCT (extended).

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv*, abs/1603.04467.

[2] Abebe, G., Cavallaro, A., and Parra, X. (2016). Robust multi-dimensional motion features for first-person vision activity recognition. *Comput. Vis. Image Underst.*, 149:229–248.

[3] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*.

[4] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: ordering points to identify the clustering structure. In *SIGMOD '99*.

[5] Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., and Massaroni, C. (2019). Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia*, 21:234–245.

[6] Babu, A. R., Zakizadeh, M., Brady, J., Calderon, D., and Makedon, F. (2019). An intelligent action recognition system to assess cognitive behavior for executive function disorder. *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 164–169.

[7] Bagchi, B. (1997). How to stay away from each other in a spherical universe. *Resonance*, 2:18–26.

[8] Bambach, S. (2015). A survey on recent advances of computer vision algorithms for egocentric video. *ArXiv*, abs/1501.02825.

[9] Bambach, S., Lee, S., Crandall, D. J., and Yu, C. (2015). Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1949–1957.

[10] Bao, W., Yu, Q., and Kong, Y. (2021). Evidential deep learning for open set action recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13329–13338.

[11] Bao, W., Yu, Q., and Kong, Y. (2022). Opental: Towards open set temporal action localization. *ArXiv*, abs/2203.05114.

[12] Bendale, A. and Boult, T. E. (2015). Towards open world recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902.

[13] Bendale, A. and Boult, T. E. (2016). Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572.

[14] Bhatnagar, B. L., Singh, S., Arora, C., and Jawahar, C. V. (2017). Unsupervised learning of deep feature representation for clustering egocentric actions. In *IJCAI*.

[15] Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:257–267.

[16] Borodachov, S. V., Hardin, D. P., and Saff, E. B. (2019). Discrete energy on rectifiable sets. *Springer Monographs in Mathematics*.

[17] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *SIGMOD '00*.

[18] Bullock, I. M., Feix, T., and Dollar, A. M. (2015). The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments. *The International Journal of Robotics Research*, 34:251 – 255.

[19] Caetano, C., Brémond, F., and Schwartz, W. R. (2019a). Skeleton image representation for 3d action recognition based on tree structure and reference joints. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 16–23.

[20] Caetano, C., de Souza, J. S., Brémond, F., dos Santos, J. A., and Schwartz, W. R. (2019b). Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

[21] Cai, M., Kitani, K., and Sato, Y. (2015). A scalable approach for understanding the visual structures of hand grasps. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1360–1366.

[22] Cai, M., Kitani, K. M., and Sato, Y. (2016). Understanding hand-object manipulation with grasp types and object attributes. In *Robotics: Science and Systems*.

[23] Castro, F. M., Marín-Jiménez, M. J., Mata, N. G., Schmid, C., and Karteek, A. (2018). End-to-end incremental learning. *ArXiv*, abs/1807.09536.

[24] Cevikalp, H., Triggs, B., and Franc, V. (2013). Face and landmark detection by using cascade of classifiers. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7.

[25] Chalamalla, A. (2010). A survey on consensus clustering techniques.

[26] Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. *ArXiv*, abs/1801.10112.

[27] Chaves, E., Gonçalves, C. B., Albertini, M., Lee, S., Jeon, G., and Fernandes, H. (2020). Evaluation of transfer learning of pre-trained cnns applied to breast cancer detection on infrared images. *Applied optics*, 59 17:E23–E28.

[28] Chen, W., Chen, X., Zhang, J., and Huang, K. (2017a). Beyond triplet loss: A deep quadruplet network for person re-identification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1320–1329.

[29] Chen, X., Guo, H., Wang, G., and Zhang, L. (2017b). Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2881–2885.

[30] Chen, Z. and Liu, B. (2016). Lifelong machine learning. *Transfer Learning*.

[31] Cho, K., Merrienboer, B. V., Çaglar Gülçehre, Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. *ArXiv*, abs/1406.1078.

[32] Chopra, S., Balakrishnan, S., and Gopalan, R. (2013). Dlid: Deep learning for domain adaptation by interpolating between domains.

[33] Crawford, S. L. (1989). Extensions to the cart algorithm. *Int. J. Man Mach. Stud.*, 31:197–217.

[34] Damen, D., Doughty, H., Farinella, G., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. *ArXiv*, abs/1804.02748.

[35] Deng, J., Guo, J., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694.

[36] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

[37] Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., and Bimbo, A. D. (2015). 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Transactions on Cybernetics*, 45:1340–1352.

[38] Devineau, G., Moutarde, F., Xi, W., and Yang, J. (2018). Deep learning for hand gesture recognition on skeletal data. *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 106–113.

[39] Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. (2019). Learning without memorizing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5133–5141.

[40] Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118.

[41] Dudoit, S. and Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3:research0036.1 – research0036.21.

[42] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*.

[43] Evangelidis, G. D., Singh, G., and Horaud, R. (2014). Skeletal quads: Human action recognition using joint quadruples. *2014 22nd International Conference on Pattern Recognition*, pages 4513–4518.

[44] fang Hu, J., Zheng, W.-S., Lai, J.-H., and Zhang, J. (2015). Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*.

[45] Fathi, A., Ren, X., and Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. *CVPR 2011*, pages 3281–3288.

[46] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941.

[47] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315:972 – 976.

[48] Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018a). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations.

[49] Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018b). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 409–419.

[50] Ge, W., Huang, W., Dong, D., and Scott, M. R. (2018). Deep metric learning with hierarchical triplet loss. In *ECCV*.

[51] Ge, Z., Demyanov, S., Chen, Z., and Garnavi, R. (2017). Generative openmax for multi-class open set classification. *ArXiv*, abs/1707.07418.

[52] Geng, C., Huang, S.-J., and Chen, S. (2021). Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3614–3631.

[53] Goodfellow, I. J., Mirza, M., Da, X., Courville, A. C., and Bengio, Y. (2014). An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, abs/1312.6211.

[54] Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P. N., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. (2017). The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851.

[55] Guan, D., Yuan, W., Lee, Y.-K., Gavrilov, A., and Lee, S. (2007). Activity recognition based on semi-supervised learning. *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*, pages 469–475.

[56] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742.

[57] Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. (2020). Remind your neural network to prevent catastrophic forgetting. In *ECCV*.

[58] Hayes, T. L. and Kanan, C. (2020). Lifelong machine learning with deep streaming linear discriminant analysis. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 887–896.

[59] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[60] Henk, M. (2006). Finite packing and covering (cambridge tracts in mathematics 154). *Bulletin of The London Mathematical Society*, 38:868–869.

[61] Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

[62] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

[63] Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *SIMBAD*.

[64] Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. (2018). Lifelong learning via progressive distillation and retrospection. In *ECCV*.

[65] Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839.

[66] Huang, Z. and Gool, L. V. (2016). A riemannian network for spd matrix learning. *ArXiv*, abs/1608.04233.

[67] Huang, Z., Wu, J., and Gool, L. V. (2016). Building deep networks on grassmann manifolds. In *AAAI*.

[68] Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

[69] Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*.

[70] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167.

[71] Irani, M. and Anandan, P. (1999). About direct methods. In *Workshop on Vision Algorithms*.

[72] Jégou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:117–128.

[73] Júnior, P. R. M., de Souza, R. M., de Oliveira Werneck, R., Stein, B. V., Pazinato, D. V., de Almeida, W. R., Penatti, O. A. B., da Silva Torres, R., and Rocha, A. (2016). Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106:359–386.

[74] Kacem, A., Daoudi, M., Amor, B. B., and Paiva, J. C. Á. (2017). A novel space-time representation on the positive semidefinite cone for facial expression recognition. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3199–3208.

[75] Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. (1997). Multilevel hypergraph partitioning: Application in vlsi domain. *Proceedings of the 34th Design Automation Conference*, pages 526–529.

[76] Karypis, G. and Kumar, V. (1998). Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distributed Comput.*, 48:96–129.

[77] Kay, E. (1974). Methods for statistical analysis of reliability and life data.

[78] Kaya, M. and Bilge, H. S. (2019). Deep metric learning: A survey. *Symmetry*, 11:1066.

[79] Kemker, R. and Kanan, C. (2018). Fearnet: Brain-inspired model for incremental learning. *ArXiv*, abs/1711.10563.

[80] Khan, A. U. and Borji, A. (2018). Analysis of hand segmentation in the wild. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4710–4719.

[81] Kim, S., Kim, D., Cho, M., and Kwak, S. (2020). Proxy anchor loss for deep metric learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3235–3244.

[82] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

[83] Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

[84] Kläser, A., Marszalek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

[85] Kotz, S. and Nadarajah, S. (2000). Extreme value distributions: Theory and applications.

[86] Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia.

[87] Krishnan, R., Subedar, M., and Tickoo, O. (2018). Bar: Bayesian activity recognition using variational inference. *ArXiv*, abs/1811.03305.

[88] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90.

[89] Lange, M. D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G. G., and Tuytelaars, T. (2019). Continual learning: A comparative study on how to defy forgetting in classification tasks. *ArXiv*, abs/1909.08383.

[90] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[91] Laradji, I. H. and Babanezhad, R. (2020). M-adda: Unsupervised domain adaptation with deep metric learning. *ArXiv*, abs/1807.02552.

[92] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324.

[93] Lee, J., Hong, H. G., Joo, D., and Kim, J. (2020). Continual learning with extended kronecker-factored approximate curvature. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8998–9007.

[94] Lee, K., Lee, K., Shin, J., and Lee, H. (2019). Incremental learning with unlabeled data in the wild. In *CVPR Workshops*.

[95] Li, C., Li, S., Gao, Y., Zhang, X., and Li, W. (2021a). A two-stream neural network for pose-based hand gesture recognition. *ArXiv*, abs/2101.08926.

[96] Li, C., Zhong, Q., Xie, D., and Pu, S. (2017). Skeleton-based action recognition with convolutional neural networks. *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 597–600.

[97] Li, F. and Wechsler, H. (2005). Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1686–1697.

[98] Li, L., Wang, M., Ni, B., Wang, H., Yang, J., and Zhang, W. (2021b). 3d human action representation learning via cross-view consistency pursuit. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4739–4748.

[99] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*.

[100] Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., and Li, W. (2021c). Trear: Transformer-based rgb-d egocentric action recognition. *ArXiv*, abs/2101.03904.

[101] Li, Y., Liu, M., and Rehg, J. M. (2018). In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*.

[102] Li, Z. and Hoiem, D. (2018). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947.

[103] Lin, J., Gan, C., and Han, S. (2018). Temporal shift module for efficient video understanding. *ArXiv*, abs/1811.08383.

[104] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422.

[105] Liu, J., Shahroudy, A., Xu, D., and Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*.

[106] Liu, S. and Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734.

[107] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746.

[108] Liu, Y., Jiang, X., Sun, T., and Xu, K. (2019). 3d gait recognition based on a cnn-lstm network with the fusion of skegei and da features. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

[109] Lohit, S., Wang, Q., and Turaga, P. K. (2019). Temporal transformer networks: Joint learning of invariant and discriminative time warping. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12418–12427.

[110] Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. *ArXiv*, abs/1502.02791.

[111] Luvizon, D. C., Picard, D., and Tabia, H. (2018). 2d/3d pose estimation and action recognition using multitask deep learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5137–5146.

[112] Ma, J., Tao, X., Ma, J., Hong, X., and Gong, Y. (2021). Class incremental learning for video action classification. *2021 IEEE International Conference on Image Processing (ICIP)*, pages 504–508.

[113] Ma, M., Fan, H., and Kitani, K. M. (2016). Going deeper into first-person activity recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903.

[114] Maghoumi, M. and LaViola, J. J. (2018). Deepgru: Deep gesture recognition utility. In *ISVC*.

[115] Mallya, A., Davis, D., and Lazebnik, S. (2018). Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*.

[116] Mallya, A. and Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773.

[117] Manmatha, R., Wu, C., Smola, A., and Krähenbühl, P. (2017). Sampling matters in deep embedding learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2859–2867.

[118] Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. (2020). Class-incremental learning: survey and performance evaluation on image classification.

[119] Mensink, T., Verbeek, J. J., Perronnin, F., and Csurka, G. (2013). Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2624–2637.

[120] Milbich, T., Roth, K., Bharadhwaj, H., Sinha, S., Bengio, Y., Ommer, B., and Cohen, J. P. (2020). Diva: Diverse visual feature aggregation for deep metric learning. In *ECCV*.

[121] Min, K. and Corso, J. J. (2020). Integrating human gaze into attention for egocentric activity recognition. *ArXiv*, abs/2011.03920.

[122] Moghimi, M., Azagra, P., Montesano, L., Murillo, A. C., and Belongie, S. J. (2014). Experiments on an rgb-d wearable vision system for egocentric activity recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 611–617.

[123] Mur-Artal, R., Montiel, J., and Tardós, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31:1147–1163.

[124] Newling, J. and Fleuret, F. (2016). Nested mini-batch k-means. In *NIPS*.

[125] Nguyen, N. and Caruana, R. (2007). Consensus clusterings. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 607–612.

[126] Nguyen, X. S., Brun, L., Lézoray, O., and Bougleux, S. (2019). A neural network based on spd manifold learning for skeleton-based hand gesture recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12028–12037.

[127] Ni, J., Liu, J., Zhang, C., Ye, D., and Ma, Z. (2017). Fine-grained patient similarity measuring using deep metric learning. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

[128] Oberweger, M. and Lepetit, V. (2017). Deepprior++: Improving fast and accurate 3d hand pose estimation. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 585–594.

[129] Oh, S. H., Yu, X., Stefanie, J., and Silvio, S. (2016). Deep metric learning via lifted structured feature embedding.

[130] Ohn-Bar, E. and Trivedi, M. M. (2013). Joint angles similarities and hog2 for action recognition. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

[131] Ohn-Bar, E. and Trivedi, M. M. (2014a). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15:2368–2377.

[132] Ohn-Bar, E. and Trivedi, M. M. (2014b). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15:2368–2377.

[133] Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723.

[134] Ostapenko, O., Puscas, M. M., Klein, T., Jähnichen, P., and Nabi, M. (2019). Learning to remember: A synaptic plasticity driven framework for continual learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11313–11321.

[135] Oza, P. and Patel, V. M. (2019). C2ae: Class conditioned auto-encoder for open-set recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2302–2311.

[136] Pan, S. J., Tsang, I. W.-H., Kwok, J. T., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22:199–210.

[137] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359.

[138] Pang, S., Ozawa, S., and Kasabov, N. K. (2005). Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35:905–914.

[139] Papadimitriou, C. H. and Steiglitz, K. (1981). Combinatorial optimization: Algorithms and complexity.

[140] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks : the official journal of the International Neural Network Society*, 113:54–71.

[141] Park, J., Kang, M., and Han, B. (2021). Class-incremental learning for action recognition in videos. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13678–13687.

[142] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[143] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011b). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[144] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., and Hinton, G. E. (2017). Regularizing neural networks by penalizing confident output distributions. *ArXiv*, abs/1701.06548.

[145] Phillips, P. J., Moon, H., Rizvi, S. A., and Rauss, P. J. (1997). The feret evaluation methodology for face-recognition algorithms. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 137–143.

[146] ping Hu, L., lei Bao, X., and Wang, Q. (2011). The repetition principle in scientific research. *Zhong xi yi jie he xue bao = Journal of Chinese integrative medicine*, 9 9:937–40.

[147] Pinheiro, P. H. O. (2018). Unsupervised domain adaptation with similarity learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8004–8013.

[148] Pirsiavash, H. and Ramanan, D. (2012). Detecting activities of daily living in first-person camera views. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2847–2854.

[149] Prakhya, S., Venkataram, V., and Kalita, J. K. (2017). Open set text classification using convolutional neural networks.

[150] Prasad, V., Das, D., and Bhowmick, B. (2020). Variational clustering: Leveraging variational autoencoders for image clustering. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10.

[151] Rahmani, H. and Mian, A. S. (2016). 3d action recognition from novel viewpoints. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1506–1515.

[152] Ramirez-Amaro, K., Beetz, M., and Cheng, G. (2017). Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artif. Intell.*, 247:95–118.

[153] Ranjan, R., Castillo, C. D., and Chellappa, R. (2017). L2-constrained softmax loss for discriminative face verification. *ArXiv*, abs/1703.09507.

[154] Rao, H., Xu, S., Hu, X., Cheng, J., and Hu, B. (2021). Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Inf. Sci.*, 569:90–109.

[155] Rastgoo, R., Kiani, K., and Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton.

[156] Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542.

[157] Rippel, O., Paluri, M., Dollár, P., and Bourdev, L. D. (2016). Metric learning with adaptive density discrimination. *CoRR*, abs/1511.05939.

[158] Rogez, G., Supancic, J. S., and Ramanan, D. (2015). Understanding everyday hands in action from rgb-d images. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3889–3897.

[159] Romano, S., Bailey, J., Nguyen, X. V., and Verspoor, K. M. (2014). Standardized mutual information for clustering comparisons: One step further in adjustment for chance. In *ICML*.

[160] Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP*.

[161] Rosenfeld, A. and Tsotsos, J. K. (2020). Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:651–663.

[162] Roth, K., Milbich, T., and Ommer, B. (2020). Pads: Policy-adapted sampling for visual similarity learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6567–6576.

[163] Roy, D., Panda, P., and Roy, K. (2020). Tree-cnn: A hierarchical deep convolutional neural network for incremental learning. *Neural networks : the official journal of the International Neural Network Society*, 121:148–160.

[164] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252.

[165] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *ArXiv*, abs/1606.04671.

[166] Ryoo, M., Rothrock, B., and Matthies, L. (2015). Pooled motion features for first-person videos. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–904.

[167] Ryoo, M. S., Joung, J. H., Choi, S., and Yu, W. (2010). Incremental learning of novel activity categories from videos. *2010 16th International Conference on Virtual Systems and Multimedia*, pages 21–26.

[168] Saito, K., Ushiku, Y., Harada, T., and Saenko, K. (2019). Strong-weak distribution alignment for adaptive object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6949–6958.

[169] Sanakoyeu, A., Tschernezki, V., Büchler, U., and Ommer, B. (2019). Divide and conquer the embedding space for metric learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 471–480.

[170] Sanderson, M. (2010). Christopher d. manning, prabhakar raghavan, hinrich schütze, introduction to information retrieval, cambridge university press 2008. isbn-13 978-0-521-86571-5, xxi + 482 pages. *Natural Language Engineering*, 16:100 – 103.

[171] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[172] Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S.-N., and Chellappa, R. (2018). Learning from synthetic data: Addressing domain shift for semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3752–3761.

[173] Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2013a). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772.

[174] Scheirer, W. J., Jain, L. P., and Boult, T. E. (2014). Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:2317–2324.

[175] Scheirer, W. J., Rocha, A., Micheals, R. J., and Boult, T. E. (2011). Meta-recognition: The theory and practice of recognition score analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1689–1695.

[176] Scheirer, W. J., Rocha, A., Sapkota, A., and Boult, T. E. (2013b). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1757–1772.

[177] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

[178] Schuster, M. and Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681.

[179] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. *Proceedings of the 15th ACM international conference on Multimedia*.

[180] Sculley, D. (2010). Web-scale k-means clustering. In *WWW '10*.

[181] Shen, H. T. (2009). Principal component analysis. In *Encyclopedia of Database Systems*.

[182] Shi, Y., Wang, Y., Zou, Y., Yuan, Q., Tian, Y., and Shu, Y. (2018). Odn: Opening the deep network for open-set action recognition. *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[183] Shu, L., Xu, H., and Liu, B. (2018). Unseen class discovery in open-world classification. *ArXiv*, abs/1801.05609.

[184] Sigurdsson, G. A., Gupta, A., Schmid, C., Farhadi, A., and Karteek, A. (2018). Charades-ego: A large-scale dataset of paired third and first person videos. *ArXiv*.

[185] Singh, S., Arora, C., and Jawahar, C. V. (2016). First person action recognition using deep learned descriptors. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2620–2628.

[186] Smedt, Q. D., Wannous, H., and Vandeborre, J.-P. (2016). Skeleton-based dynamic hand gesture recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1206–1214.

[187] Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*.

[188] Song, H. O., Jegelka, S., Rathod, V., and Murphy, K. P. (2017). Deep metric learning via facility location. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2206–2214.

[189] Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012.

[190] Sridhar, S., Feit, A. M., Theobalt, C., and Oulasvirta, A. (2015). Investigating the dexterity of multi-finger input for mid-air text entry. In *CHI '15*.

[191] Steinbach, M. S., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques.

[192] Strehl, A. and Ghosh, J. (2002a). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

[193] Strehl, A. and Ghosh, J. (2002b). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

[194] Subedar, M., Krishnan, R., López-Meyer, P., Tickoo, O., and Huang, J. (2019). Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6300–6309.

[195] Sudhakaran, S., Escalera, S., and Lanz, O. (2019). Lsta: Long short-term attention for egocentric action recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9946–9955.

[196] Sudhakaran, S. and Lanz, O. (2017). Convolutional long short-term memory networks for recognizing first person interactions. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2339–2346.

[197] Sudhakaran, S. and Lanz, O. (2018). Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. *ArXiv*, abs/1807.11794.

[198] Surie, D., Pederson, T., Lagriffoul, F., Janlert, L.-E., and Sjölie, D. (2007). Activity recognition using an egocentric perspective of everyday objects. In *UIC*.

[199] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*.

[200] Tadesse, G. and Cavallaro, A. (2018). Visual features for ego-centric activity recognition: a survey. *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*.

[Tammes] Tammes, P. M. L. On the origin of number and arrangement of the places of exit on the surface of pollen-grains.

[202] Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International journal of scientific and research publications*, 9:9420.

[203] Tang, Y., Tian, Y., Lu, J., Feng, J., and Zhou, J. (2017). Action recognition in rgb-d egocentric videos. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3410–3414.

[204] Teh, E. W., Devries, T., and Taylor, G. W. (2020). Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. *ArXiv*, abs/2004.01113.

[205] Tekin, B., Bogo, F., and Pollefeys, M. (2019). H+o: Unified egocentric recognition of 3d hand-object poses and interactions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4506–4515.

[206] Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22 19:2405–12.

[207] Tschiatschek, S., Iyer, R. K., Wei, H., and Bilmes, J. A. (2014). Learning mixtures of submodular functions for image collection summarization. In *NIPS*.

[208] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971.

[209] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *ArXiv*, abs/1412.3474.

[210] van der Maaten, L. and Hinton, G. E. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

[211] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *ArXiv*, abs/1706.03762.

[212] Vaudaux-Ruth, G., Chan-Hon-Tong, A., and Achard, C. (2021). Actionspotter: Deep reinforcement learning framework for temporal action spotting in videos. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 631–638.

[213] Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595.

[214] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. J. (2011). The caltech-ucsd birds-200-2011 dataset.

[215] Wang, H., Kläser, A., Schmid, C., and Liu, C. (2012). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79.

[216] Wang, H. and Wang, L. (2017). Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3633–3642.

[217] Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., and Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

[218] Wang, J., Zhou, F., Wen, S., Liu, X., and Lin, Y. (2017). Deep metric learning with angular loss. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2612–2620.

[219] Wang, L., Huynh, D. Q., and Koniusz, P. (2019a). A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29:15–28.

[220] Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. (2019b). Multi-similarity loss with general pair weighting for deep metric learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5017–5025.

[221] Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., and Robertson, N. M. (2019c). Ranked list loss for deep metric learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5202–5211.

[222] Wang, Z., She, Q., Chalasani, T., and Smolic, A. (2020). Catnet: Class incremental 3d convnets for lifelong egocentric gesture recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 935–944.

[223] Weinberger, K. Q. and Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

[224] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016a). A discriminative feature learning approach for deep face recognition. In *ECCV*.

[225] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016b). A discriminative feature learning approach for deep face recognition. In *ECCV*.

[226] Willems, G., Tuytelaars, T., and Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*.

[227] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. R. (2019). Large scale incremental learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382.

[228] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., and Fu, Y. R. (2018). Incremental classifier learning with generative adversarial networks. *ArXiv*, abs/1802.00853.

[229] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747.

[230] Xiao, H., Sun, F., and Liang, Y. (2010). A fast incremental learning algorithm for svm based on k nearest neighbors. *2010 International Conference on Artificial Intelligence and Computational Intelligence*, 2:413–416.

[231] Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995.

[232] Xu, D. and jie Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193.

[233] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

[234] Yang, Y., Xu, D., Nie, F., Yan, S., and Zhuang, Y. (2010). Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19:2761–2773.

[235] Yeung, K. Y. and Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms supplement to the paper " an empirical study on principal component analysis for clustering gene expression data " (to appear in bioinformatics).

[236] Ying, X. (2019). An overview of overfitting and its solutions.

[237] Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., and Naemura, T. (2019). Classification-reconstruction learning for open-set recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4011–4020.

[238] Yu, B. and Tao, D. (2019). Deep metric learning with tuplet margin loss. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6489–6498.

[239] Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and van de Weijer, J. (2020). Semantic drift compensation for class-incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6980–6989.

[240] Zanfir, M., Leordeanu, M., and Sminchisescu, C. (2013). The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. *2013 IEEE International Conference on Computer Vision*, pages 2752–2759.

[241] Zhai, A. and Wu, H.-Y. (2019). Classification is a strong baseline for deep metric learning. In *BMVC*.

[242] Zhang, C., Yang, X., and Tian, Y. (2013a). Histogram of 3d facets: A characteristic descriptor for hand gesture recognition. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8.

[243] Zhang, H., Li, L., Jia, W., Fernstrom, J., Sclabassi, R., and Sun, M. (2010). Recognizing physical activity from ego-motion of a camera. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 5569–5572.

[244] Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., and Kuo, C.-C. J. (2020a). Class-incremental learning via deep model consolidation. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1120–1129.

[245] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *SIGMOD '96*.

[246] Zhang, W., Lin, Z., Cheng, J., Ma, C., Deng, X., and Wang, H. (2020b). Sta-gcn: two-stream graph convolutional network with spatial–temporal attention for hand gesture recognition. *The Visual Computer*, 36:2433–2444.

[247] Zhang, W., Zhao, D., and Wang, X. (2013b). Agglomerative clustering via maximum incremental path integral. *Pattern Recognit.*, 46:3056–3065.

[248] Zhang, X., Qin, S., Xu, Y., and Xu, H. (2019). Quaternion product units for deep learning on 3d rotation groups. *ArXiv*, abs/1912.07791.

[249] Zhang, X., Wang, Y., Gou, M., Sznaier, M., and Camps, O. (2016). Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[250] Zhao, B., Xiao, X., Gan, G., Zhang, B., and Xia, S. (2020). Maintaining discrimination and fairness in class incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13205–13214.

[251] Zheng, J., Furao, S., Fan, H., and Zhao, J. (2011). An online incremental learning support vector machine for large-scale data. *Neural Computing and Applications*, 22:1023–1035.

[252] Zhou, P., Mai, L., Zhang, J., Xu, N., Wu, Z., and Davis, L. S. (2019). M2kd: Multi-model and multi-level knowledge distillation for incremental learning. *ArXiv*, abs/1904.01769.

[253] Zhu, Q., Chen, Z., and Soh, Y. C. (2019). A novel semisupervised deep learning method for human activity recognition. *IEEE Transactions on Industrial Informatics*, 15:3821–3830.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

CentraleSupélec

Titre : Les principaux composants d'un système de reconnaissance d'activité de la main, exploitable pour l'assistance aux utilisateurs en réalité augmentée.

Mots clés : Reconnaissance d'activité de la main de la première personne, apprentissage par métrique, apprentissage d'ensembles ouverts

Résumé : Les êtres humains utilisent leurs mains pour diverses tâches dans la vie quotidienne et professionnelle, ce qui fait que la recherche dans ce domaine a récemment suscitée un grand intérêt. De plus, l'analyse et l'interprétation du comportement humain à l'aide de signaux visuels est l'un des domaines les plus actifs et les plus explorés de la vision par ordinateur. Avec l'arrivée des nouvelles technologies de réalité augmentée, les chercheurs s'intéressent de plus en plus à la compréhension de l'activité de la main d'un point de vue de la première personne, en explorant la pertinence de son utilisation pour le guidage et l'assistance humaine. L'objectif principal de cette thèse est de proposer un système de reconnaissance de l'activité de l'utilisateur incluant quatre composants essentiels, qui peut être utilisé pour assister les utilisateurs lors d'activités orientées vers des objectifs spécifiques : industrie 4.0 (par exemple, assemblage assisté, maintenance) et enseignement. Ainsi, le système observe les mains de l'utilisateur et les objets manipulés depuis le point de vue de l'utilisateur afin de reconnaître et comprendre ses activités manuelles réalisées. Le système de réalité augmenté souhaité doit reconnaître de manière robuste les activités habituelles de l'utilisateur. Néanmoins, il doit détecter les activités inhabituelles afin d'informer l'utilisateur et l'empêcher d'effectuer de mauvaises manœuvres, une exigence fondamentale pour l'assistance à l'utilisateur. Cette thèse combine donc des techniques issues des domaines de recherche de la vision par ordinateur et de l'apprentissage automatique afin de proposer des composants de reconnaissance de l'activité de l'utilisateur nécessaires à un outil d'assistance complet.

Title : The main components of a hand activity recognition framework, exploitable for augmented reality user assistance.

Keywords : First-person hand activity recognition, Metric Learning, open-set recognition

Abstract : Humans use their hands for various tasks in daily life and industry, making research in this area a recent focus of significant interest. Moreover, analyzing and interpreting human behavior using visual signals is one of the most animated and explored areas of computer vision. With the advent of new augmented reality technologies, researchers are increasingly interested in hand activity understanding from a first-person perspective exploring its suitability for human guidance and assistance. Our work is based on machine learning technology to contribute to this research area. Recently, deep neural networks have proven their outstanding effectiveness in many research areas, allowing researchers to jump significantly in efficiency and robustness. This thesis's main objective is to propose a user's activity recognition framework including four key components, which can be used to assist users during their activities oriented towards specific objectives: industry 4.0 (e.g., assisted assembly, maintenance) and teaching. Thus, the system observes the user's hands and the manipulated objects from the user's viewpoint to recognize his performed hand activity. The desired framework must robustly recognize the user's usual activities. Nevertheless, it must detect unusual ones to feedback and prevent him from performing wrong maneuvers, a fundamental requirement for user assistance. This thesis, therefore, combines techniques from the research fields of computer vision and machine learning to propose comprehensive hand activity recognition components essential for a complete assistance tool.