Università di Genova

DIPARTIMENTO DI
INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Cross View Action Recognition

Gaurvi Goyal

Theses Series

DIBRIS-TH-2020-42

Università di **Genova**

Dipartimento di Informatica, Bioingegneria,

Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in
Computer Science and Systems Engineering
Computer Science Curriculum

Cross View Action Recognition

by

Gaurvi Goyal

March, 2020

Dottorato di Ricerca in Informatica ed Ingegneria dei Sistemi
Indirizzo Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova

DIBRIS, Univ. di Genova
Via Dodecaneso, 35
I-16145 Genova, Italy
https://www.dibris.unige.it/

Ph.D. Thesis in
Computer Science and Systems Engineering
Computer Science Curriculum
(S.S.D. INF/01)

Submitted by Gaurvi Goyal
DIBRIS, Univ. di Genova
gaurvi.goyal@dibris.unige.it

Date of submission: March 2020

Title: Cross View Action Recognition

Advisor: Francesca Odone
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
francesca.odone@unige.it

Ext. Reviewers:
Alessandra Sciutti
Center for Human Technologies
Italian Institute of Technology
alessandra.sciutti@iit.it

Dima Damen
Department of Computer Science
University of Bristol
dima.damen@bristol.ac.uk

Paolo Napoletano
DISCo (Department of Informatics, Systems and Communication)
University of Milan
paolo.napoletano@unimib.it

Ph.D. Thesis in Computer Science and Systems Engineering (S.S.D. INF/01)
Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi
Università di Genova

Candidate
Gaurvi Goyal
gaurvi.goyal@dibris.unige.it

Title
Cross View Action Recognition

Advisor
Francesca Odone
DIBRIS, Università di Genova
francesca.odone@unige.it

Location
DIBRIS, Univ. di Genova
Via Dodecaneso, 35
I-16145 Genova, Italy

Submitted On
March 2020

To my mother
who always believes in me.

Abstract

Cross View Action Recognition (CVAR) appraises a system's ability to recognise actions from viewpoints that are unfamiliar to the system. The state of the art methods that train on large amounts of training data rely on variation in the training data itself to increase their ability to tackle viewpoints changes. Therefore, these methods not only require a large scale dataset of appropriate classes for the application every time they train, but also correspondingly large amount of computation power for the training process leading to high costs, in terms of time, effort, funds and electrical energy. In this thesis, we propose a methodological pipeline that tackles change in viewpoint, training on small datasets and employing sustainable amounts of resources. Our method uses the optical flow input with a stream of a pre-trained model as-is to obtain a feature. Thereafter, this feature is used to train a custom designed classifier that promotes view-invariant properties. Our method only uses video information as input, in contrast to another set of methods that approach CVAR by using depth or pose input at the expense of increased sensor costs. We present a number of comparative analysis that aided the design of the pipelines, farther assessing the power of each component in the pipeline. The technique can also be adopted to existing, trained classifiers, with minimal fine-tuning, as this work demonstrates by comparing classifiers including shallow classifiers, deep pre-trained classifiers and our proposed classifier trained from scratch. Additionally, we present a set of qualitative results that promote our understanding of the relationship between viewpoints in the feature-space.

Acknowledgements

Francesca Odone has been a better PhD advisor and mentor to me than anyone can ever hope to have, and I would like to convey my deepest gratitude for her time, effort, guidance and patience and most importantly, for always keeping her door open for her students. I would also like to thank Nicoletta Noceti for her guidance and our collaborations during the PhD study. I would like to thank University of Genova, DIBRIS, and previous and current colleagues for all the help and plenty of constructive conversations throughout the PhD term with special mention to Damiano Malafronte.

My family has been remarkably supportive and encouraging. I would like to thank my mother, Sangeeta Goyal, who calls me her strength but in reality she is mine, my sister, Yamini Goyal, to whom I owe, in big part, my sanity and my father, Subhash Goyal. I could not have done this without them.

I would like to thank Chiara Bassano, Vanessa D'Amario, Luca Demetrio, Elena Nicora and Matteo Moro for being both colleagues and friends and making my time at the University all the more memorable. I also thank my friends, Ashish Nanda, Ishu Goel, Vyshakh Palli Thazha, Emily-Jane Rolley-Parnell, Ana Tanevska for the beautiful people that they are and for being there every time I needed advice or to talk. And I sincerely thank Andrea Corradi and Beatrice Vizzi for making Genova feel more like home.

Contents

List of Figures

List of Tables

1

Introduction

Human Action recognition (HAR) has applications in a variety of fields, human-robot interaction, medical surveillance, service industry, security surveillance to name a few. Thus, its increased popularity in research in the recent years is conceivably a precursor to widespread use in the near future. Existing sensors, like camera phones, webcams, mounted cameras for surveillance, can largely facilitate the early adoption. Widespread use would involve increasingly uncontrolled settings, where systems would need to function in the wild, incorporating the ability to tackle variance in the data. Actions are inherently three dimensional motions, thus, they can look substantially different when seen from diverse viewpoints. A camera acquires rich two-dimensional projections of complex three-dimensional scenes. Therefore, camera sensor data capture involves loss of depth information pertinent for learning the 3D motion of an action, thereby triggering the challenge of recognizing actions from arbitrary or unseen viewpoints. Using methods with the right design considerations, it may be possible to use the richness of the data to compensate for this loss of depth information, thus achieving improved ability to recognize actions from viewpoints not included in the training data of a system.

The property of observing and processing an object or a motion in a way that the result is independent of the viewpoint from which the object or motion is viewed, is called *View Invariance*. Testing a system's ability to be view-invariant would theoretically require an infinite number of testing viewpoints, or practically a large number of them. In HAR applications, with the multi-view datasets available today (listed and detailed in Table 3), videos from a maximum of 8 viewpoints are available, making it practically impossible for the property to be proven. This makes any claim of calling a video-based method *View Invariant* a relatively shallow one. Instead, *Cross-View Action Recognition* (CVAR) is the task of learning actions from one or limited viewpoints and recognising actions from viewpoints not present in the training data. This task tests the system's ability to adapt to unseen viewpoints and can also be used to study relationships between these viewpoints. CVAR can be considered a step towards the generalization of a system towards view-invariance. For these reasons, we choose to tackle the task of cross-view action recognition in our work. It is

necessary to note that CVAR is different from Multi-View Action Recognition (MVAR). While both scenarios use multi-view datasets, MVAR tests on views that are already involved in training while in CVAR, testing data is sourced from viewpoints that are not used in training. CVAR is a closer match to the action representation ability of humans and human skill is often considered a benchmark for HAR.

Generally, HAR is considered to have two primary components: action detection and action classification. Action detection is the task of temporally localizing or segmenting an action clip within a longer video sequence. Action classification takes temporally trimmed inputs containing a single action each and classifies the actions. Our work is focused on action recognition, which we formalize as a multi-class, supervised, classification problem. In this document, we use the terms *action recognition* and *action classification* interchangeably.

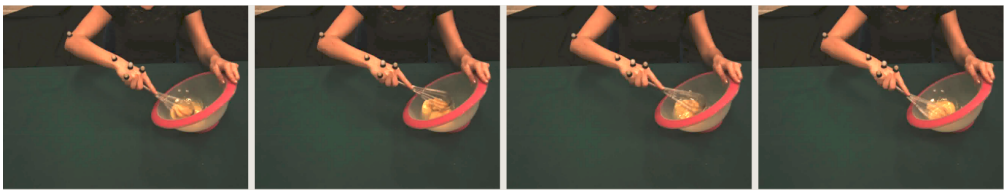
Action recognition is affected by viewpoints and change of these viewpoint, in two ways. The first is the dynamics of the motion. The height, distance and viewing angle can substantially change the shape in which the motion is perceived by the viewer. In Figure 1 note that the motion of the same action is perceived as considerably different from the three viewpoints.

The second factor is self-occlusion, part of the action being covered by person executing the action. For example, refer to Figure 2 where the action of scratching the head is demonstrated. In this example, the viewpoint from the front which is generally be considered optimal in terms of capturing most information of the action, would lose the pertinent information: the dynamics of the scratching motion behind the head. From that point of view, it would be impossible to decipher if the person is scratching her head, or rubbing/ pressing it, or combing her hair down with her fingers. This was a simpler example but this problem can be amplified substantially with increase in percentage of action occluded with respect to the viewpoint. Loss of information is inherent to the nature of this problem.

Most recently, many works addressing CVAR have employed Motion Capture (MoCap) systems and RGBD depth sensors either exclusively or both together [88]. MoCap provides a sparse but precise, 3D location of body limbs or joints while depth sensors provides a richer and still precise 3D location of the limbs, though limited by the line of sight like a regular camera. Thus, MoCap and RGBD sensors are relatively more apt at addressing the problem of dynamic motion due to viewpoint change, but the issue of self occlusion remains a major concern in all cases. Additionally, the sparse information provided by MoCap may be detrimental when a more granular set of classes are concerned, where movements of the major limbs is quite similar, like the example above. Therefore, considering the availability, cost and nature of the different sensors, we choose to use only video information as our input.

(a) View 0

(b) View 1



(c) View 2

Figure 1: Samples of actions from the Kinetics dataset [114]

The primary goal of this thesis is to explore view invariance of human action, using the cross-view action recognition applications, by probing the limits of video input and capability of representations built using video data, and doing so with minimal resources, in terms of training data and computation requirement.

1.1 Motivation

Action recognition is quickly becoming an interesting application. With 70% of internet traffic being videos in 2016 [113] with estimates to grow, automated processing of these videos is becoming a general requirement the same way as images are processed today by the variety of tools on-line and off-line in our daily lives. The first video-based action recognition methods appeared decades ago [111, 127] and over time, the expertise has improved as with any research field. Additionally, with the release of large scale datasets in recent years enabling improved data-driven machine learning approaches, a number of substantially effective action recognition methods have been released recently. Largely based on Deep Learning, these methods involve end-to-end training of deep models using large amounts of data in order to train effectively [16, 151]. Even within CVAR scenario, similar techniques have shown proficiency [88, 175]. But the cost of these methods, in terms of data require-

(a) Cam1 (b) Cam2

Figure 2: Frame samples of the *scratch head* action from the IXMAS dataset [187]

ments and computations is creating a gap between the effective methods and what is feasible by widespread users. We have taken the initiative towards reducing this gap.

During the progression of this work, we made a number of choices. These choices were directly influenced by 3 factors:

i *Wide applicability*: The action recognition scenario is far from simple to address. This is largely due to the complexity of the task, and the nature and amounts of data required by most methods available today. Our objective involves building methods that move HAR to a more approachable scenario, opening doors to more light-weight applications. Ideally, they should

- Use easily available or existing infrastructure, in terms of sensors
- Require short time to capture and annotate training data
- Be economically feasible in terms of computational systems involved in training or deployment.

ii *Scientific Curiosity*: The loss of information in videos makes CVAR a very challenging problem yet many classical methods had in fact resulted in relatively good results [50, 87, 90, 208]. Thereafter, with depth and skeleton based methods clearly outperforming them with a margin, video based methods were practically abandoned for half a decade in between. At the beginning of our work in 2016, it was an open question if it is possible to achieve results competitive to depth and skeleton based methods.

iii *Sustainability*: With the exponential increase of computational requirements closely related to the machine learning explosion in the last decade, and the influx of data centres catering to this market, the concern regarding the carbon emissions based on this has finally appeared. Al-

> though the initial concern pertained to data storage, statistics with respect to the computation, especially the newest machine learning methods have gained attention lately. Most data centres are using a combination of carbon based, renewable and nuclear sources of energy to fuel their electricity and cooling requirements. Therefore, if one is using a cloud based computing solution, it is very likely that one is contributing to some amount of carbon emission. For these reasons, we have made a commitment towards development of sustainable computing practices, by focusing on minimizing the energy required by the systems to train, such that state of the art results can still be achieved with a fraction of the cost of the environment in terms of carbon emissions, or more directly, a fraction of computation time.

The first and the third point are partially intertwined. A very recent work by Strubell, Ganesh and McCallum (2019), has quantified the carbon footprint of training a neural Network, Transformer, for a Natural Language Processing task with architecture search, a method that progressively changes the architecture of the model for optimal results, with a number of generalizations and assumptions. They derived the value to be equivalent of the total lifetime emissions of 5 cars including their manufacturing process. In monetary terms, the training time is derived to be approximately USD 942,973 – USD 3,201,722 in cloud computing cost. With these being the baselines costs and any substantial research requiring multiple training cycles, the values can increase considerably above these numbers. Although cloud computing can be financially expensive, it can be argued that large amounts of computations do not necessarily elicit high carbon footprint. Renewable sources of electricity can substantially reduce emissions. But for now, none of the larger data centers use fully renewable sources of energy, with anywhere between 29% to 54% of the total energy used being sourced from carbon based sources, as stated by the same work. Data centres are constantly investing efforts in making the electrical usage more efficient but any gain by increasing efficiency is outdone by the scale at which these computation requirements is increasing[1].

Considering another example of resource usage, in 2018, Sony broke the record of the minimum time taken to train a deep neural network, a ResNet-50, on the image dataset ImageNet, bring the training time down to 224 seconds from 6.6 minutes, using *2,176* GPUs parallelly [110]. The massive amounts of resources involved are jarring for most private and public research institutes. This demonstrates the private industry's investment in systems that can train faster but also reinforces the need to develop computationally efficient machine learning methods and pipelines that can be employed with limited resources.

1 https://www.theguardian.com/environment/2015/sep/25/server-data-centre-emissions-air-travel-web-google-facebook-greenhouse-gas

Table 1: Constraints

Sensors to Obtain the Data	Cameras only
Amount of Training Data	Order of (lower) 10s per class
Variation in Training Data	No/Minimal Change in Viewpoint
Computational Power for Training	One 16 GB GPU
Computation Time for Training	<10 min per label

These motivations together led to a list of constraints that we worked towards incorporating into our system, that are listed in Table 1. Within this scenario we set out to explore 3 primary questions that are the backbone to this entire work:

- Is it possible to build a system that performs Cross View Action Recognition with acceptable results, solely from videos and within the other imposed constraints?
- How well can this system perform with only information from a single viewpoint to train on?
- Can the system combine the information from multiple viewpoints in order to obtain a quasi-view-invariant representation, i.e. a representation robust among viewpoints without significant self-occlusion?

1.2 Challenges

The first challenge that we faced during the course of this work has been to find appropriate datasets. A number of action recognition datasets with multi-view video data are publicly available (see Table 2). Some factors made the choice of dataset a challenge. In order to study the relationship between viewpoints, we needed of same fixed viewpoints for all actions. For example, the *Breakfast actions* dataset has a large number of videos samples and a variety of action classes. It has 18 different kitchens, and 52 different subjects. On the other hand, to test a model's ability to learn from, say, a single view point, we would need samples of each action from the same one view point, which is not the case in the dataset. The cameras are fixed for a specific kitchen (the number of cameras and their viewpoint vary per kitchen), but the orientation of an actor while performing any given action, can vary. While this brings the dataset much closer to a real life scenario, it makes a structured study substantially more complicated. On the other hand, more defined datasets in

terms of viewpoint, tend to be smaller, with lower level of naturalness present in the actions.

Another factor concerning the datasets is the specific viewpoints that each dataset provides. Different multi-view datasets have captured a variety of different viewpoints but typically, they only contain allocentric (second or third person) viewpoint. Egocentric viewpoints are still largely missing from the multi-view analysis. However, a large fraction of video data being uploaded to the internet today is egocentric, and with the increase in augmented reality and spectacle related technologies, it may be crucial to leverage the allocentric-egocentric relationship. The allocentric data can provide a head start to egocentric action recognition today, and in the future, with ego vision data overtaking the allocentric data availability, egocentric data can return the favor. This relationship can also be beneficial for humanoid robotics applications. This is the primary reason that we chose to incorporate the *Multimodal Cooking Actions* dataset [114], that has been acquired in-house, and contains 2 allocentric and one pseudo-egocentric viewpoint which is captured by a camera placed close to the head of the actor (sample shown in Figure 1). Overall, we used 4 different datasets in order to facilitate the study of a variety of aspects like number of viewpoints, types of actions, training samples per class, relationships between different viewpoints etc.

The second challenge is related to the approach towards computer vision and the interest that the field has garnered in the recent times. During the course of this work, a large number of other works both in the field of action recognition and cross-view action recognition have been published. The open question, *can videos provide sufficient information to perform CVAR*, has been answered in the affirmative by works contemporary to ours albeit using substantially more resources [10, 88, 175]. Although these methods are different from our own, they reinforce our accomplishment. They provide a validation of the results, just as our work provides the same to them, proving that videos do provide sufficient information for effective CVAR. The pace of research today with the explosion of works in computer vision and machine learning have bolstered growth in all areas and CVAR has received the merited interest.

The third and most challenging aspect of the work has been to minimize the ecological impact of the research. A primary objective has been to ensure that the final method should require minimal computation throughout the pipeline, despite the fact that the task of action recognition itself is a very complex task and requires considerable learning by a system. We have dedicated our efforts in both creating an *ad hoc* computed representation as well using the most appropriate learnt representation available in the literature. Knowledge transfer from pre-existing sources has been key in minimizing resource requirements. Throughout the work we avoid end-to-end learning, altogether.

Table 2: Datasets with multiview visual data. FB: Full Body. UB: Upper Body. VV: Varying view capture

Dataset	Classes	Views	Subjects	Total Clips	FB/UB	Year
IXMAS [187]	13	5	10	1650	FB	2006
I3DPost [45]	12	8	8	832	FB	2009
MuHaVi [153]	17	8	14	3808	FB	2010
Breakfast actions [75]	43	3-5	52	11267	~	2014
MHAD [116]	11	4	12	2640	FB	2014
N-UCLA [180]	10	3	10	1500	FB	2014
UWA 3D multiview activity II [134]	30	4	9	1080	FB	2014
NTU RGB+D [147]	60	5	40	56880	FB	2016
UESTC [66]	40	8 + VV	118	25,600	FB	2018
MoCA [114]	20	3	1	1500	UB	2017

1.3 Contributions

Minimal resource in terms of training data requires a representation that is robust to view changes by virtue of its property. Thus the first step has been to investigate the existence of a representation, or a process to create one, that is largely robust to view changes out-of-the-box. First we tested the use of a hand crafted representation based on the Shearlet Transform. The representation shows promise not only for action recognition applications but especially for its cross-view recognition properties. In terms of efficiency, it was possible to bring down the computational requirements with algorithm designs. On the other hand, a custom designed classifier was needed to tap into the potential of the representation.

At the same time, it was becoming clear that the rest of the research in computer vision was largely taken over by deep learning methods, be it partially or end-to-end. Their success was not only impossible to ignore, but as researchers it also sparked our scientific curiosity, wondering if our questions could be answered using these methods. Therefore, we investigated a number of possibilities to finally arrive at using a learnt representation extracted from the temporal stream of the dual stream Inception3D model [16]. The representation is effective for the a regular action recognition task, but it also contains view-specific information that can be detrimental to simple classifiers when working towards the CVAR task. In order to promote the ability of the classifiers to recognize unseen viewpoints based on this representation, we treat the viewpoints as domains and adopt Domain Adaptation methods, specifically

using a variation of Batch Normalization in the classifiers. During the course of the work we conducted 3 sets of comparative analyses, on 3 different extraction points for the Inception3D feature extractor, between features extracted from the different streams of the two-stream deep network, and the third comparison between a variety of different classifiers including our proposed classifier which demonstrated superior robustness to viewpoint change.

The contributions of our work are manifold:

- A quantitative gauge of adaptive capabilities, of a representation extracted from a pre-trained network, with respect to viewpoint change and unseen views.
- A classifier with a proven view-invariance capability.
- Evidenced that the incorporation of the simple technique of modified batch normalization can help classifiers and architectures adapt to new viewpoints.
- The first analysis of the relationship between egocentric and allocentric viewpoints in HAR scenario, to the best of our knowledge.

1.4 Thesis Overview

The thesis is divided into 2 parts: Part I, subdivided into two chapters, lays down the building blocks of this research work, both in terms of background information as well as the state-of-the-art of the relevant areas. Chapter 2 details the classical approaches pertaining to the action recognition scenario and our early efforts towards tackling the cross-view problem. Chapter 3 focuses on background information from the Deep Learning aspect and summarizes works in the area of Action recognition in general as well as cross-view in particular.

Part II comprises of Chapter 4 where we describes the methodology involved in the work, proposing the pipeline and a number of significant comparative analyses, and Chapter 5 which reports the experimental details and a comprehensive analysis of the results.

The document concludes with Chapter 6 providing a discussion on the advantages and disadvantages of the studies and the methods covered in this thesis.

PART I

Theoretical Background and State-of-the-art

This part of the document will cover a detailed study of the state of the art of topics and fields pertaining to our problem statement as well as the relevant theoretical background information that would be required in order to fully grasp the contribution of the thesis. Here, we also describe our initial approaches and the reasoning behind them.

2

Classical Approaches to Action recognition

Understanding human motion and its regularities is a key research goal of Human-Machine Interaction, with a potential to unlock more refined abilities – such as the anticipation of action goals – and thus the design of intelligent machines able to proficiently and effectively collaborate with humans [38, 165].

Human action recognition (HAR) is a subset of the broader field of motion analysis, and specifically biological motion analysis. Hence it is only natural that a number of components classically are derived from these areas. Initial works in Action Recognition were built on top of existing work in the domain of motion and video analysis. Over years, there have also been many works based on single images based action recognition [48]. We mostly limit our discussion to video based methods, giving a overview of the major aspects of important classical approaches to HAR, with more detailed discussion on topics relevant to the work presented in this document. There have been a number of surveys and reviews on the area of Human Action Recognition using video input [13, 14, 30, 111, 127, 174, 204, 207].

Section 2.1 will cover the overview, taxonomy and categorization of the classical approaches to Action Recognition. Section 2.2 will comprise specifically the feature extraction approaches and the different types of hand crafted features used in action recognition. Section 2.3 covers approaches dealing with cross view and view invariant action recognition. And finally, in Section 2.4 we discuss an approach that we investigated in the beginning of this work, followed by concluding remarks in Section 2.5.

2.1 Introduction

We first introduce the main reference problems in the field of Human Activity Recognition or HAR. In the literature, there are usually three distinct terms,

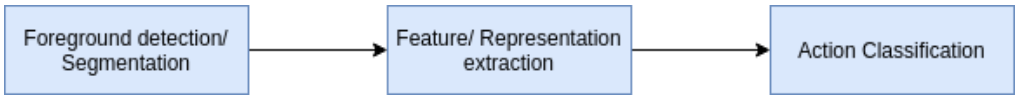


Figure 3: Pipeline followed by most classical action recognition methods.

action primitives, *actions*, and *activities* that form a hierarchical structure. Action primitives are atomic motions at the limb level which constitute an action, either by being placed both in parallel and in sequence. *e.g.* 'move left leg forward' and 'lift left arm' are action primitives and 'throwing a ball' is an action. Some works distinguish between actions and activities such that multiple actions make up an activity, *e.g.* 'running', 'dribbling' and 'shooting a hoop' are actions that make up the activity of 'playing basketball'. Other works use the terms 'actions' and 'activities' interchangeably. We will be using the terms interchangeably in this work unless specified otherwise.

The principle paradigm followed by most methods for HAR applications is shown in Figure 3. The first step is segmentation or separating the foreground and background of the input. The second step, feature extraction and building a representation followed by the final step, classification, which usually employs machine learning algorithms. Note that in this paradigm, usually the term 'feature' is referred to lower level descriptors and the term 'representation' is used for higher level information. The sequential nature of the methodology requires effective execution of each step of the pipeline for a system to function well. For the complex HAR task, the representations are needed to be accurate and discriminative for the machine learning classification algorithms to be effective.

SEGMENTATION Segmentation methods separate the salient region of interest (ROI) or foreground from the remaining image or background. These methods can be divided farther into two categories, background construction-based and foreground extraction-based methods. Background construction-based methods [155] construct a background from initial frames and employ background subtraction to obtain the ROI in the successive frames.These methods are effective when the camera is static and the ROI is fast-moving *e.g.* in a surveillance scenario. Foreground-extraction methods are effective when the camera itself is in motion, *e.g.* in a moving vehicle or Unmanned Aerial Vehicle. In this case temporal, spatial, or spatio-temporal information is employed to obtain the initial ROI from video and in the successive frames, ROIs are determined using change and motion information. A relatively recent survey on action recognition that has a detailed section on these methods is written by Bux, Angelov and Habib [14]. An older, though still relevant review is available in [188].

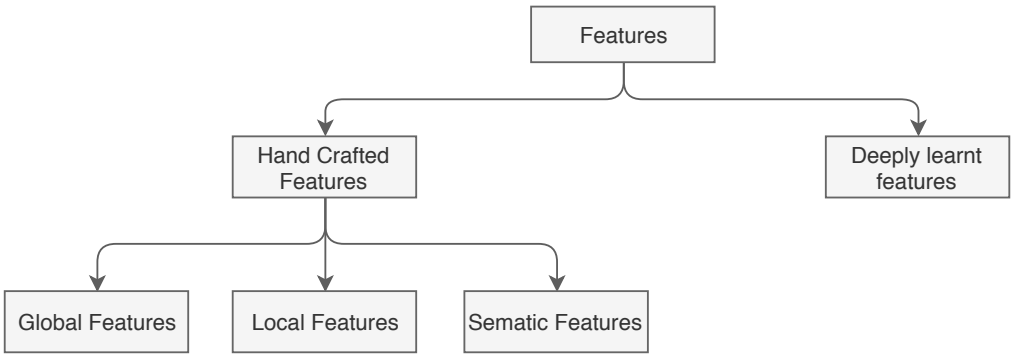


Figure 4: Categorization of the features based on methods of extraction.

BUILDING REPRESENTATION Representations can vary widely in their sizes, computation time, information they provide, depending on the methods involved in building them. Classically, prior knowledge of the field and the data has been used to build these representations, also known as hand-crafted representations (see Figure 4). Many of representation pipelines involve conventional machine learning algorithms, *e.g.* Bag of Visual Words with K-nearest neighbors to build a codebook at an intermediate step during the feature extraction processes. More recently, data-driven machine learning methods employing multiple layers to learn features and representations automatically, have been used extensively in the vision domains, also known as representation learning or deep learning. Hand crafted representations are examined in Section 2.2. Deeply learning based methods are discussed in detail in Chapter 3.

CLASSIFICATION Classification techniques developed at the pace of machine learning algorithms. Indeed, many algorithms that were designed for other fields were re-purposed to action recognition from other machine learning applications, *e.g.* Dynamic Time Warping (DTW) and Hidden Markov Model (HMM) were initially proposed for speech recognition [130, 131] and then used in HAR applications. Some methods that have been used in HAR applciations are DTW[173], HMMs [140, 164], K-nearest neighbors[69, 76], SVM[26, 128, 136, 145], Kalman filters [12, 21], and more recently, ANNs and deep networks (to be discussed in Chapter 3). More detailed account of classical machine learning classification methods can be found in [14, 207].

2.2 Hand-Crafted Representations

Building hand-crafted representations typically involved 2 major steps: Feature extraction and Encoding, each consisting of smaller steps. Feature extraction involves obtaining low level description of the input. Encoding methods

use these descriptors to build higher level representations of the input sample. This often involves machine learning techniques at intermediate steps. In order to enhance tolerance to transformation, encoding is usually followed by a pooling step before a classifier is applied.

Different works have categorized hand-crafted features in a variety of ways. We choose to follow the categorization by Bux, Angelov and Habib [14] who divided the features based on the type of description they provide, into global, local and semantic.

A survey specifically on different types features is written by Sargano, Angelov and Habib [143], while [14, 207] also give relatively recent account of state-to-the-art of hand-crafted features.

2.2.1 *Global features*

Global representations describe the entire space-time volume, and are usually derived directly from videos. Segmentation results that are fed directly into classifiers alone or in combinations with other features can be categorized as global representations. The categorization of global features can be overlapping and often methods and techniques complement each other in practical applications.

Silhouette

Silhouette based methods use background subtraction to obtain the shapes of moving objects in a scene known as shapes or silhouette [191]. These silhouette can be stacked over the time axis to form a space-time volume which are then used as representation. The silhouette can be in 2D or 3D. The 3D silhouette is easier to calculate now with RGBD data but 2D shape of the human body have also been used for action recognition applications [172]. Data from multiple orthogonal viewpoints has been considered by a few approaches in order to incorporate view-invariant properties [20, 187, 195].

Optical Flow

Optical Flow (OF) is an effective tool to extract Region of Interest(ROI) in a video, especially for a dynamic background [105] and plays a crucial role in our own work. Therefore, we would take this opportunity to present a more detailed account of the notion of Optical Flow.

A 2D displacement field describing the apparent motion of brightness patterns between two successive images is called the optical flow [57, 125, 169]. Consider a uniformly colored sphere rotating along its own axis in a scene. The reflected brightness will not capture this movement. For this reason, it is said to capture *apparent* motion. Consider a pair of images Δt apart in time such that a point (x, y, t) with intensity $I(x, y, t)$ moves to $I(x + \Delta x, y + \Delta y, t + \Delta t)$ in the Δt time. It is assumed that the apparent brightness of objects remains constant over short movements *i.e.* ,

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{1}$$

This is called the *brightness constancy constraint*. From this, the *Image Brightness Constancy Equation* can be derived:

$$\bigtriangledown I(\bar{u}) + \frac{\partial I}{\partial t} = 0 \tag{2}$$

where $\bar{u}$ is a velocity vector or vector field. There are two unknowns in Equation 2 and only one equation. The system of linear equations is underdetermined. This requires another equation or, more precisely, another constraint to be imposed on the motion field. A number of methods exist which propose different constraints on the optical flow field.

Horn and Schunck [57] suggested a variational formulation of the optical flow problem with a smoothness condition, minimizing:

$$E(\bar{u}) = \int \{\bigtriangledown(u) + \frac{\partial I}{\partial t})^2 + (\mid \bigtriangledown V_x \mid^2 + \mid \bigtriangledown V_y \mid^2\} \tag{3}$$

where V_x and V_y are the velocity vectors in x and y directions respectively.

Another method is the TVL_1 optical flow which minimizes a term that contains the L_1 norm and uses a regularization with a total variation of the optical flow. The Equation 2 holds when the image data is continuous in time. Often it is replaced by a non-linear formulation: $I_1(\bar{x} + \bar{u}) - I_0(\bar{x})$. Thus the Energy to be minimized is given by:

$$E = \int \{\mid I_1(\bar{x} + \bar{u}) - I_0(\bar{x}) \mid + \mid \bigtriangledown \bar{u} \mid\} dx \tag{4}$$

For a deeper understanding of Optical Flow, we invite the reader to refer to the book by Trucco and Verri [168]. OF is a widely used feature today in action recognition for its ability to isolate the motion in the video. Optical flow and silhouette-based methods can be considered as either segmentation methods or features extraction methods depending on the applications. Optical flow is also used more recently as input to deep nets, that are explored in detail in chapter 3. 3D scene flow [62, 171], sometimes also referred to as 3D Optical flow, which uses data from spatial as well as depth information has also been proposed and used in action recognition [8].

Space-time volume

Space-time based methods concatenate the frames of the video obtaining a 3D volume where the X and Y are the spatial dimensions and Z is the temporal dimension, considering the RGB data as a volume of 2D+t pixels. These methods consider all pixels of the image sequence or specifically the pixels involved in the motion in the scene, called as Space Time Volumes (STV). Gorelick et al. [47] proposed stacking silhouettes to obtain a space-time shape. Due to the inherent difference between the space and time axis, traditional 3D shape analysis could not be used. They solved this by deriving local space-time saliency and orientation features using the Poisson equation. Achard et al. [3] proposed a space-time micro volumes to incorporate temporal invariance in actions, *i.e.* variance in time taken to perform an action. Shechtman and Irani [148] propose a patch-wise matching of volumes increasing tolerance to change in scale and orientation.

Discrete Fourier Transform

Assuming that the foreground and background differ in intensity, the DFT is an image frame can be used to obtain information about the foreground. This property has also been used in action recognition [76] although the method is restricted by the necessity of a simple background.

2.2.2 *Local features*

Local features treat patches of the sample separately, instead of the entire sample as a whole. This allows higher tolerance to partial occlusions and noise when compared to global representations. These patches can be either (i) densely sampled [68] or (ii) points of interest can be detected in the sample [53]. Interest points or key points indicate a significant local variation of image intensities. Their use can reduce the number of patches on which the local features must be computed. In HAR, key-points based methods generally consider videos as $x - y - t$ volumes, treating the three dimensions similarly. Trajectory based methods employ dense sampling, tracking trajectories of points through time, treating $x - y$ dimensions differently from t.

Interest Points Based Methods

For images, these detected interest points are usually associated with edges or more commonly, corners, *i.e.* locations of abrupt change in color or intensities. Harris and Stephens [53] proposed one of the most popular 2D interest points

detector, Harris corner detector, used in object recognition in images. For video data, the community showed great interest towards the space-time key-points (STIPs). The seminal work by Laptev [82] proposed a space-time extension of the spatial corner points. Soon it was followed by alternative and some possibly richer approaches [32, 118, 120, 190]. The power of these key-points has been appreciated as low level building blocks for motion analysis and action recognition. Space-time key points mark special points where the signal undergoes a significant variation both in space and time, indicating spatio-temporal corners, and for this reason stable STIPs are quite rare. They carry meaningful information in particular for distinctive dynamic events, but may show lower effectiveness with more subtle actions or gestures. These points are calculated directly on video data, skipping the segmentation step, and tolerant to geometric transformation, perspective transformation, illumination variation and convolution transformation A survey on STIPs for HAR is written by Dawn and Shaikh [30].

We drew inspiration from STIP and formulated a method to identify interest points in videos using the Shearlet transform which we discuss in Section 2.4.

Trajectory based methods

Another set of local descriptors, trajectory based methods, treat the spatial dimensions as different from the temporal dimension, locating features in $x - y$ and tracking them in t and usually use dense sampling of the input. Wang and Schmid [177] and Wang et al. [178] proposed the Dense Trajectories (DT) and later the improved Dense Trajectories (iDT) approach that exploits dense optical flow to track motion over time. The iDT eliminate the camera motion encoded in the optical flow, to some extent while using human detection to ensure human motion does not affect the camera motion detection. Other works using trajectories have also been proposed [41, 123, 179].

Descriptors

Local Descriptors provide a description of the detected of sampled features. These must be sufficiently discriminative for the task, which in our case is action recognition, and tolerant to illumination changes, noise, occlusions and rotation. It is common to describe a key point or trajectory using descriptors calculated on a space-time volume around the detected point (*i.e.* a patch). It is possible to use the space-time volume of a patch as a descriptor.

Histogram of Oriented Gradients (HOG) descriptors [26] are used for human detection and simultaneous tracking and action recognition [104]. These features are invariant to illumination and tolerant to pose and viewpoint changes but are affected by change in scale. Laptev et al. [83] combine HOG with Histo-

gram of Optical Flow, obtaining spatio-temporal features, with HOG capturing significant spatial information and HOF describing the temporal information. The MBH (Motion Boundary Histograms) [27] has demonstrated ability to capture human motion. iDT [178] use a description of space time volumes around the trajectories, calculating HOG, HOF and MBH (Motion Boundary Histograms) [27] descriptors for a comparative analysis.

Scale Invariant Feature Transform (SIFT) [102, 103] and Speeded-Up Robust Features (SURF) [11], originally proposed for object recognition, are feature descriptors with inbuilt key point detectors. SURF and 3DSIFT [146] (the temporal counterpart of SIFT) that have been used for HAR applications [115, 179].

2.2.3 *Semantic features*

Semantic representation is the higher level representation that describe inherent characteristics of an input. This allows the method to be more reliable as well as more interpretable. These types of features can be highly effective to tackle the intra-class variability of action representation. They also help build a better understanding of the underlying aspects of the composition of an action. Semantic information can refer to a number of different types of information pertaining to the action, like the objects involved, the location or background, the attributes of the action and the pose of the human body or pose of parts of the body, called poselets.

The classification methods based on available 3D pose are invariant to viewpoint and appearance but largely depend on the accuracy of 3D pose itself. Pose-based methods have been used more recently with the wider use of depth sensors that allow a very precise localization of the human pose in 3D space. Some of the earlier works in 3D pose estimation from RGB video data used no prior human model [29] or used a prior model [86] or a 3D geometric depiction and kinematic description of the human body [85].

For a detailed analysis of the earlier semantic representation based methods for action recognition, refer to the work by Ziaeefard and Bergevin [214] or a more updated though less detailed part in [14].

2.2.4 *Encoding methods*

Local features provide local information which in turn captures only small amounts of information about the location. Additionally, for the purpose of recognition, local features can vary significantly within a single class. Moreover,

they may capture unimportant information *e.g.* background. Therefore before classification we need to (i) filter out the important elements and represent them in a canonical form, called *coding* and (ii) derive a higher level description to represent a whole image or large parts, called pooling. Hence, extracted features are usually encoded, before being fed into trained classifiers. Better encoding techniques can have a substantial impact on the recognition accuracy of the pipeline. One of the most commonly used methods is the Bag-of-Features [154] (BoF; or Bag-of-Visual-Words) approach involving Vector Quantization (VQ) [154] encoding which employs k-means algorithm. Other classical representation pipelines that have been employed in HAR are dictionary learning and genetic programming [97].

BoF algorithm is based on the Bag-of-Words [109] approach used in document matching which considered a document as a collection (or bag) of unordered words. The subset of the vocabulary used in a document and the frequency of each word in this subset constitute the representation of a document or sample.

Extending this paradigm to visual information is not straightforward since the vocabulary of words is missing. Moreover, the task here is action recognition which is a multiclass classification problem instead of matching. Thus the vocabulary[1] using a K-means clustering approach from features of a representative training data and the frequency of words is calculated using distance of the features from the centroids of these clusters. The pipeline traversed by a query sample, employing a standard BoF approach is as follows.

i *Local feature extraction*: Interest points are detected and described by a d-dimensional local descriptor ($X = \{x_z \in \mathbb{R}^d | z = 1, \ldots, N\}$).

ii *Vector Quantization*: The vector quantization step encodes each local descriptor x_z by a so-called Visual Dictionary ($C = \{c_i \in \mathbb{R}^d | i = 1, \ldots, K\}$) into a K-dimensional code vector α_z in a pre-defined feasible region :

$$\alpha_z = \arg\min_{\alpha_z} \|x_z - D\alpha_z\|, \text{s.t.} \tag{5}$$

In the BoF approach, is constrained to the set 0-1 vectors with only a single component equal to 1, which is known as the hard assignment. Each element $\alpha_{z,i}$ of the code vector α_z indicates the local descriptor response to the i^{th} visual word in the dictionary C. The dictionary is learned by K-means algorithm.

iii *Image representation*: The histogram of occurrences of visual words, with dimension $D = k$, is computed and weighted using inverse sample frequency terms, *i.e.* weights calculated based on frequency of each visual word in the training samples. Rarer visual words are found to be more disciminative.

1 vocabulary or codebook or dictionary

iv *Vector normalization*: The resulting vector is subsequently normalized. There are several variations on how to normalize the histogram. When seen as an empirical distribution, the BoF vector is normalized using the Manhattan distance. Another common choice consists in using Euclidean normalization.

Some of the other commonly used encoding methods, that are involved in similar or slightly different way in encoding inputs and have been used for HAR applications include, Sparse Coding (SPC) [196], Locality-constrained Linear Coding (LLC) [181], Fisher Vector (FV) [126], and Vector of locally aggregated descriptors (VLAD) [63, 64]. For more methods and details, we refer you to survey by Zhang et al. [207] and a study by Zhen and Shao [209]. Some encoding methods have been compared [17, 185, 209] and for action recognition, FV have been shown as one of the most discriminative encoding methods with demonstrated high compatibility with iDT features [209]. A Stacked FV (SFV) [124] has also been proposed which works even better with iDT features than FV.

Dictionary learning approaches are a broad set of approaches that learn a sparse representation of the data, hence being ideal for a classification task. Dictionary learning may be a preliminary step in a Bag-of-Features approach. The name is derived by the fact that the dictionary needs to be learned using training data, that can be considered representative of the query data that the system must classify. In the context of dictionary learning, the vocabulary can be termed as a dictionary (though the terms are used interchangeably in BoF as well) and its components are called atoms. For example, HAR has been addressed by dictionary learning by Wang et al. [176] with a hierarchical descriptor.

Genetic Programming is a powerful, although less explored, machine learning technique, that has been employed to learn features, based on natural evolution. Wang et al. [181] proposed a method that learnt spatio-temporal motion features by evolution over a population of 3D-operations like 3D-Gabor filters and wavelets for the action recognition task.

2.3 View Invariant Action recognition

View Invariance is an important aspect of action recognition. The appearance of an action can change considerably with the viewpoint. Figure 5 shows simultaneous frames from videos captured from 3 viewpoints synchronously, of a subject mixing something in a bowl. Note that the shape of the motion of the arm would also be different in all three cases.

(a) (b) (c)

Figure 5: A mixing action viewed from 3 different viewpoints.

View-invariant action recognition has been approached from a variety of different directions. The most primary aspect has been to build features that are view-invariant. For this, multi-view datasets have been employed that can provide information about actions from multiple viewpoints, which have been discussed in Section 1.2.

Existing Methods for Cross View Action Recognition

In early works view-invariance was approached as an epipolar geometry problem, resorting to a coarse 3D reconstruction of the performed action [142, 160, 198], while later works can be categorized into methodologies acting at a descriptor level or at a similarity level. The first type aims at the design of representations explicitly embedding view-invariant information [60, 67, 87], while the purpose of the latter, often based on Machine Learning, is to define specific strategies to evaluate the similarity between actions observed from multiple views [58, 194, 210].

In this category we also find methods addressing the problem with a transfer learning formulation, to provide the model with the capability of transferring information from one view to another [211] or to a common virtual view, sometimes in a 3D reference frame [90, 141].

Silhouettes extracted from a single view lacked any tolerance to viewpoint change. Many works approached this with silhouettes extracted from multiple views. Weinland, Ronfard and Boyer [187] used Motion History Volumes (MHV) with orthogonal camera views to estimate 4D silhouettes. Xu and Huang [195] used videos from two orthogonal viewpoints to extract an envelop shape representation. Cherla et al. [20] use width feature of the normalized silhouette box with Dynamic Time Warping.

Rogez, Guerrero and Orrite [141] constructed a discretized viewing hemisphere, dividing it into finite number of training viewpoints and training 2D-Pose and shape features to train separate models for each view. At query, they select the appropriate view-based model for the input sample to extract appropriate low-level features. Li, Camps and Sznaier [87] proposed a view-invariant

feature called Hankelets, the Hankel matrix of a short tracklet. One of the first works in View Invariance pertaining to videos, Rao, Yilmaz and Shah [135] was based on analysis of trajectories and identification of points tolerant to change in viewpoint. Zhu and Shao [212] propose an unsupervised approach to cross view action recognition, using low level trajectories encoded using locality-constraint linear coding (LCC). They also proposed [213] a weakly supervised cross-domain dictionary learning approach to visual recognition. Zheng et al. [211] used transferable dictionary pairs for supervised cross view action recognition.

Natarajan and Nevatia [112] presented an approach for recognizing activities using synthetic poses of actions and several low level features. Motion Capture (MoCap) data recordings of actions are used to render pose templates from multiple views and represented by a graphical representation. They use pedestrian detector, matches of image edges with model silhouettes and motion flow features and use similarity scores and a 2-layer Conditional Random Field to obtain a simultaneous tracking and recognition system. Note this method requires MoCap data for each class as a pre-requisite while query samples comprise RGB video information.

2.4 A Shearlet-based representation for action recognition

In this section, we discuss an approach we developed for visual applications and an analysis of its applicability on action recognition and potential for cross-view action recognition.

We considered upper body human action primitives, from the MoCA dataset. We restricted our attention to the actor, and did not exploit any contextual information which could be derived, for instance, by the presence of a tool or an object.

Our first attempt at addressing HAR belongs to the local representation approach. We took inspiration from Laptev [81] where instead of retaining the sole information provided by these hand-crafted space-time key-points, we learn *ad hoc* space-time local primitives for a given class of actions. Given a dynamic event, different meaningful local primitives can be observed and associated with an appropriate meaning in space and time [108]. To achieve this goal we follow an unsupervised approach and consider a signal representation based on Shearlets [78, 80]. Shearlets emerge among multi-resolution models by their ability to efficiently capture anisotropic features, to detect singularities [49, 79] and to be stable against noise and blurring [19, 36, 37]. The effective-

ness of Shearlets is supported by a well-established mathematical theory and confirmed by a variety of applications to image processing [35, 36, 78].

We propose a pipeline to represent the space-time information embedded in an image sequence, in an unsupervised setting. First, from the 2D+T Shearlet coefficients we represent a space-time neighborhood by appropriately encoding the signal behavior in space and time. Then, we learn a dictionary of space-time local primitives or atoms meaningful for a specific action set. To do so, we follow a BoK approach [24], applying a clustering procedure to all the space-time points of a training set of image frames. Finally, we represent a video sequence as a set of time series depicting the evolution of the primitives frequency over time.

In the preliminary results we obtained, we analyze this information and evaluate whether it is meaningful and stable to multiple repetitions of the same action and discriminative among different but similar actions. We also evaluate its robustness to view point variations and investigated the descriptive power of dictionaries learnt by different datasets. Instead of addressing view-invariance as a general property we focus on a set of different view points that describe typical observation points in human-human interaction (ego-view, frontal view, lateral view) as they are meaningful to a natural HMI.

2.4.1 *Shearlet Theory: an overview*

In this section, we briefly review the construction of the discrete shearlet transform of a $2D + T$ signal f by adapting the approach given in [77] for 3D signals.

Denoted by L^2 the Hilbert space of square-integrable functions $f : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{C}$ with the usual scalar product $\langle f, f' \rangle$, the discrete shearlet transform $SH[f]$ of a signal $f \in L^2$ is the sequence of coefficients

$$SH[f](\ell, j, k, m) = \langle f, \Psi_{\ell,j,k,m} \rangle$$

where $\{\Psi_{\ell,j,k,m}\}$ is a family of filters parametrized by

i A label $\ell = 0, \ldots, 3$ of 4 regions or pyramids $\mathcal{P}_\ell$ in the frequency domain;

ii The scale parameter $j \in \mathbb{N}$;

iii The shearing vector $k = (k_1, k_2)$ where $k_1, k_2 = -\lceil 2^{j/2} \rceil, \ldots, \lceil 2^{j/2} \rceil$;

iv The translation vector $m = (m_1, m_2, m_3) \in \mathbb{Z}^3$.

For $\ell = 0$ the filters, which do not depend on j and k, are

$$\Psi_{0,m}(x, y, t) = \varphi(x - cm_1)\varphi(y - cm_2)\varphi(t - cm_3), \tag{6}$$

where $c > 0$ is a step size and φ is a 1D-scaling function. The system $\{\Psi_{0,m}\}_m$ takes care of the low frequency cube $\mathcal{P}_0 = \{(\xi_1, \xi_2, \xi_3) \in \widehat{\mathbb{R}}^3 \mid |\xi_1| \leqslant 1, |\xi_2| \leqslant 1, |\xi_3| \leqslant 1\}$.

For $\ell = 1$ the filters are defined in terms of translations and two linear transformations (parabolic dilations and shearings)

$$A_{1,j} = \begin{pmatrix} 2^j & 0 & 0 \\ 0 & 2^{j/2} & 0 \\ 0 & 0 & 2^{j/2} \end{pmatrix} \qquad S_{1,k} = \begin{pmatrix} 1 & k_1 & k_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{so that}$$

$$\Psi_{1,j,k,m}(x, y, t) = 2^j \psi_1 \left(S_{1,k} A_{1,j} \begin{pmatrix} x \\ y \\ t \end{pmatrix} - \begin{pmatrix} c m_1 \\ \hat{c} m_2 \\ \hat{c} m_3 \end{pmatrix} \right), \tag{7}$$

where c is as in (6) and $\widehat{c} > 0$ is another step size (in the rest of the paper we assume that $c = \hat{c} = 1$ for sake of simplicity). The system $\{\Psi_{1,j,k,m}\}$ takes care of the high frequencies in the pyramid along the x-axis: $\mathcal{P}_1 = \{(\xi_1, \xi_2, \xi_3) \in \widehat{\mathbb{R}}^3 \mid |\xi_1| \geqslant 1, |\frac{\xi_2}{\xi_1}| \leqslant 1, |\frac{\xi_3}{\xi_1}| \leqslant 1\}$. For $\ell = 2, 3$ we had a similar definition by interchanging the role of x and y (for $\ell = 2$) and of x and t (for $\ell = 3$).

Our algorithm is based on a property that allows association with any shearing vector $k = (k_1, k_2)$ a direction (without orientation) parametrized by two angles, namely *latitude* and *longitude*, given by

$$(\cos\alpha \cos\beta, \cos\alpha \sin\beta, \sin\alpha) \qquad \alpha, \beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]. \tag{8}$$

The correspondence depends on ℓ and, for the first pyramid, it is given by

$$\tan\alpha = \frac{2^{-j/2} k_2}{\sqrt{1 + 2^{-j} k_1^2}} \quad \tan\beta = 2^{-j/2} k_1 \quad \alpha, \beta \in [-\frac{\pi}{4}, \frac{\pi}{4}].$$

The fact that Shearlets are sensitive to orientations allows us to discriminate among spatial-temporal features of different kinds [107, 108].

2.4.2 *Building dictionaries of space-time primitives*

1 - Space-time point representation (Fig. 6). We start by considering a point $\hat{m}$ for the fixed scale $\hat{j}$ and the subset of shearings encoding different directions: $\mathbf{K} = \left\{ k = (k_1, k_2) \,|\, k_1, k_2 = -\lceil 2^{\hat{j}/2} \rceil, \ldots, \lceil 2^{\hat{j}/2} \rceil \right\}$. We perform the following steps:

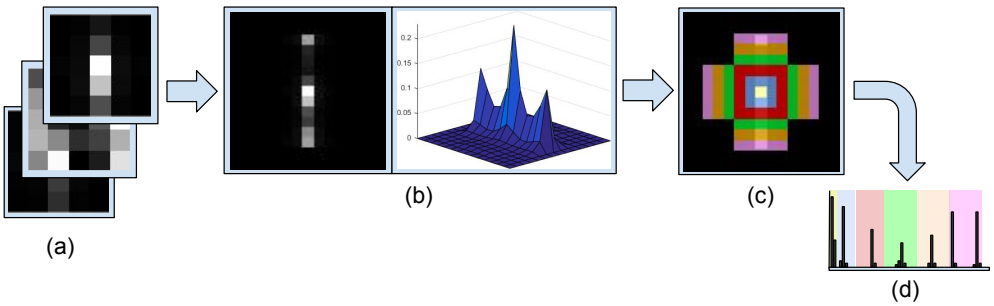


Figure 6: 2D + T point representation: (a) Matrices $C_1(r,c)$, $C_2(r,c)$ and $C_3(r,c)$; (b) Object **C** both in gray-levels and 3D visualization; (c) Coefficients grouping; (d) The obtained representation **D**.

FIG. 6A We reorganize the information provided by $SH[f](\ell, \hat{j}, k, \hat{m})$ in three $M \times M$ matrices, each one associated with a pyramid ℓ, where each entry is related to a specific shearing: $C_\ell(r,c) = SH[f](\ell, \hat{j}, k_{rc}, \hat{m})$ with $\ell = 1,2,3$, where r and c, are discrete versions of k_1 and k_2.

FIG. 6B We merge the three matrices in a single one. The obtained overall representation **C** is centered on k_{max}, the shearing corresponding to the coefficient with the maximum value in the set $SH[f](\ell, \hat{j}, k, \hat{m})$, with $\ell \in \{1,2,3\}$ and $k \in \mathbf{K}$. The matrix **C** models how the shearlet coefficients vary in a neighborhood of the direction where there is the maximum variation, and it is built in a way so that the distance of every entry of **C** with respect to the center is proportional to the distance of the corresponding angles (as defined in (8)) from the angles associated with k_{max}. Different kinds of spatio-temporal elements can be associated with different kinds of local variations in **C** (see for instance Fig. 10).

FIG. 6C We now compute a compact rotation-invariant representation for point $\hat{m}$. We group the available shearings in subsets $\bar{s}_i$, according to the following rule: $\bar{s}_0 = \{k_{max}\}$ and $\bar{s}_i$ will contain the shearings in the *i-th* ring of values from k_{max} in **C**.We extract the values corresponding to the coefficients for $\bar{s}_1$ (by looking at the 8-neighborhood of k_{max}), then we consider the adjacent outer ring (that is, the 24-neighborhood without its 8-neighborhood) to have the coefficients corresponding to $\bar{s}_2$, and so on.

FIG. 6D We built a vector containing the values of the coefficients corresponding to each set: $\mathbf{D}(\hat{m}) = coeff_{\bar{s}_0} \frown coeff_{\bar{s}_1} \frown coeff_{\bar{s}_2} \frown \ldots$; $coeff_{\bar{s}_i}$ is the set of coefficients associate with each shearings subset $\bar{s}_i$:

$$coeff_{\bar{s}_0} = SH[f](\ell_{k_{max}}, \hat{j}, k_{max}, \hat{m})$$
$$coeff_{\bar{s}_i} = \left\{ SH[f](\ell_{\bar{s}_i}, \hat{j}, k_{\bar{s}_i}, \hat{m}), k_{\bar{s}_i} \in \bar{s}_i \right\},$$

where $\ell_{k_{max}}$ is the pyramid associated with the shearing k_{max} and where $\ell_{\bar{s}_i}$ represents the pyramid associated with each shearing $k_{\bar{s}_i}$.

2 - Learning a dictionary of space-time primitives (Fig. 7).

FIG. 7A This phase considers a set of meaningful frames in a (set of) sequence(s). The frames are chosen automatically through a key-point detection process [107]. We select the N_f frames with the highest number of interest points and we assume that these are the most representative of an action event.

FIG. 7B We represent each point $\hat{m}$ of every selected frame by means of $\mathbf{D}(\hat{m})$, for a fixed scale $\hat{j}$. On each frame, we apply K-means and obtain a set of K cluster centroids, which we use as space-time primitives or atoms.

FIG. 7C We re-apply K-means on all the previously obtained atoms [108]. We end up with a dictionary $\mathcal{D}$ of N_a space-time primitives.

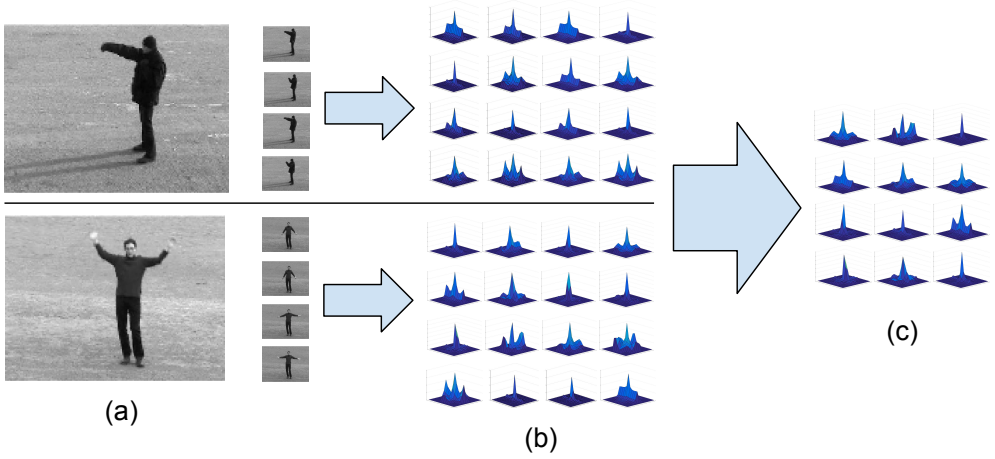


Figure 7: Learning the dictionary. (a) Automatic selection of meaningful frames from the training set; (b) Atoms learnt by each sequence; (c) Dictionary summarization on the whole training set.

3 - Encoding a video sequence with respect to a dictionary (Fig. 8). We now considered a sequence V of a given action.

FIG. 8B For each image frame $I_t \in V$ we follow a BoK approach and quantize points of I_t w.r.t the dictionary atoms, obtaining F_i^t frequency values (how many points in frame I_t can be associated with the $i-th$ atom).

FIG. 8C We filter out still primitives that are not useful to our purpose. To do this, we consider a point-wise index which we call *dynamism measure* (DM):

$$DM[\hat{m}] = SH[f](\ell_{k_{max}}, j, k_{max}, \hat{m}) \cdot \cos(\Theta_{k_{max}}, \vec{n}) \tag{9}$$

where for a given point $\hat{m}$ we considered the value corresponding to its maximum shearlet coefficient and its associated shearing parameter k_{max}; $\Theta_{k_{max}}$ is the associated direction obtained using (8) and $\vec{n}$ is

the normal vector to the xy plane in our signal (i.e. aligned with the temporal axis). To discard still patterns we consider only the values of $DM[\hat{m}]$ which are above a given threshold τ. The angle $\Theta_{k_{max}}$ decided whether a point belongs to a spatio-temporal structure which is moving or not[2], while the $SH[f](\ell_{k_{max}}, j, k_{max}, \hat{m})$ factor selects points representing a *strong* spatio-temporal change. Finally, we compute temporal sequences of frequency values across time, obtaining N_a time series or profiles $\{P_j\}_{j=1}^{N_a}$, which summarized the content of the video sequence.

(a) (b) (c)

Figure 8: Action encoding: (a) A sample frame; (b) The quantization w.r.t. the dictionary atoms; (c) Examples of temporal profiles (see text for details).

2.4.3 *Experimental analysis*

2.4.3.1 *Experimental protocol*

The data we consider in this experiment is drawn from the Multimodal Cooking Actions Dataset (MoCA) [114] discussed in detail in Chapter 5.

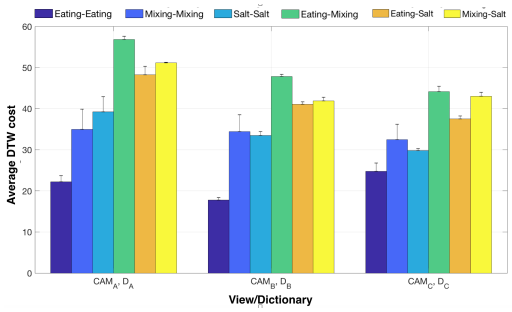
For this preliminary analysis we consider a subset of 3 actions. For each action and each view we consider 3 action instances. In the following experiments we consider dictionaries learnt from *Eating* actions only. For the detection phase (see [107]), we fix the number of selected frames N_f to 4 and consider only shearlet coefficients at scale 2. For the dictionary learning phase, the number of centroids per frame is $K = 8$, and the final dictionary size is $N_a = 12$.
We evaluate the dissimilarity between action pairs by means of Dynamic Time Warping (DTW). Given two videos V^1 and V^2 depicting a certain action instance and described by two sets of temporal profiles $P^1 = \{P_i^1\}_{i=1}^{N_a}$ and $P^2 = \{P_i^2\}_{i=1}^{N_a}$ then $Dis(V^1, V^2) = avg_{i=1}^{N_a} DTW(P_i^1, P_i^2)$. Z-normalization is applied to the temporal profiles before computing the dissimilarity.

2 Points belonging to still spatio-temporal structure spawn surfaces over time, and the normal vector $\Theta_{k_{max}}$ for those points will belong to the xy plane, bringing the value for $\cos(\Theta_{k_{max}}, \vec{n})$ to be 0.

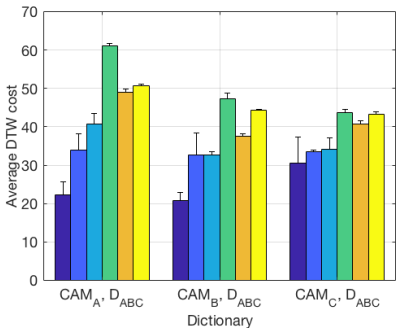
2.4.3.2 *Preliminary investigation*

We investigate discriminative power of the approach in addressing the primary HAR challenges: intra-class variations, inter-class similarities. Moreover we analyse the effect of the dictionary source to the results and the potential of this approach for view-invariance.

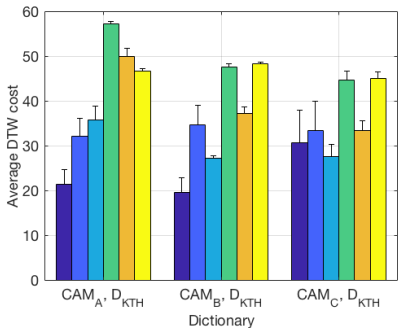
1. How informative are the learnt space-time dictionaries to discriminate among different actions captured from the same view? In this experiment



(a) View dictionary



(b) Combined dictionary



(c) KTH dictionary

Figure 9: Average DTW cost obtained when comparing actions of the same view using different dictionaries.

we consider comparisons between actions observed from a given viewpoint, described according to a dictionary obtained from the same view: we refer to such dictionaries as D_A, D_B, and D_C. Fig. 9a shows the average DTW cost in aligning the instances of the action classes. It can be observed that on average the comparisons of actions from the same class have a lower cost. Among the 3, CAM_C appeared to be the most challenging viewpoint. It can also be noted that *Eating* action is the best performing, as dictionaries are built on eating examples. At the same time a good generalization to other actions can be observed.

2. What is the relationship between different dictionaries learnt from different viewpoint data? Is there any benefit in learning dictionaries from differ-

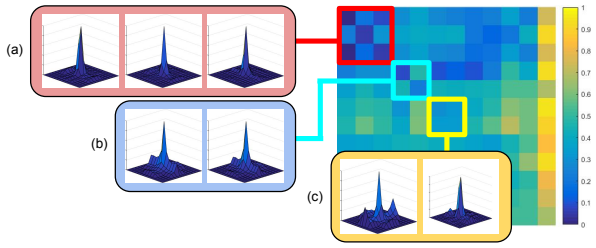
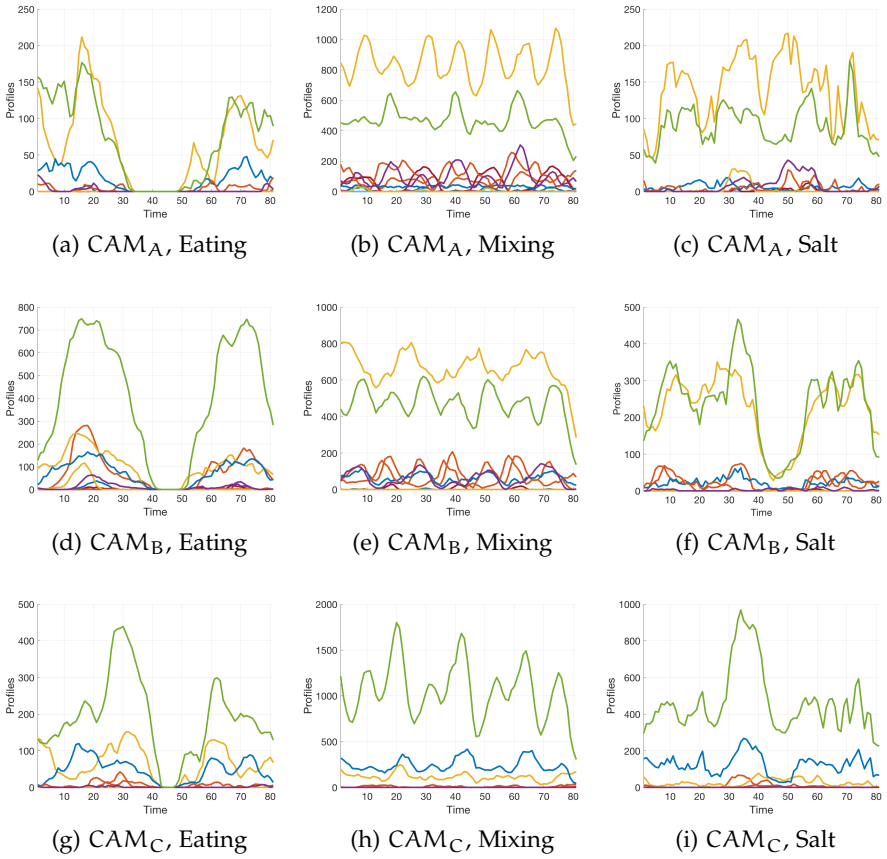


Figure 10: An example of dissimilarity matrix between atoms of two different dictionaries (from CAM_A and CAM_B), with a selection of prototypes encoding different dynamic properties of the signal.

ent views? To answer this question, we compare dictionaries specific to different views, and observe that they encode similar spatio-temporal primitives. We build a dissimilarity matrix collecting the Euclidean distances between atoms of the two dictionaries. The atoms were then matched using the Hungarian algorithm, and their contributions were sorted in the dissimilarity matrix accordingly. As a consequence, on the main diagonal we find agglomerations of atoms belonging to different dictionaries but encoding the same kind of spatio-temporal information. Fig. 10 shows an example where dictionaries referring to CAM_A and CAM_B are considered, and where groups of atoms carrying similar information are highlighted. At the top of the diagonal a group of 3 atoms (Fig. 10a) described a moving edge-like structures, which correspond to a surface in the space-time domain. Similarly, the primitives in Fig. 10b and 10c represent corner-like structures with a different amount of dynamic variations in the direction around the principal one.
As we observe a large overlap between different dictionaries, we also consider the benefits of learning a joint dictionary from the 3 views, as this choice would simplify inter-view comparisons.Fig. 9b shows how stable the performance was when adopting D_{ABC} for all the data.

3. To what extent the space-time representation is view-invariant? Fig. 11 provides a preliminary qualitative answer to the question. The plots represents the average profiles of all actions instances. *Eating* is characterized by the highest stability across views, while *Mixing* presents some differences in CAM_C with respect to the other two views. This may be explained with the fact the action is performed following a quasi-planar shape on the table, favouring a clear and regular apparent motion from the top view. *Salt* is a less constrained action characterized by a higher degree of instability over time and across views. Fig. 12a reports the average DTW costs obtained from pairs of views. On the left (D_{ABC}) we confirm that *Eating* is stable across views, while a higher intra-class variability is associated with *Mixing*. We also notice a similarity between *Eating* and *Salt*. A visual inspection of the corresponding profiles in Fig. 11 confirms the presence of common temporal patterns.
We observe that the different temporal profiles are characterized by an uneven amount of stability. This suggested that a selection of the profiles to be used



(a) CAM_A, Eating (b) CAM_A, Mixing (c) CAM_A, Salt

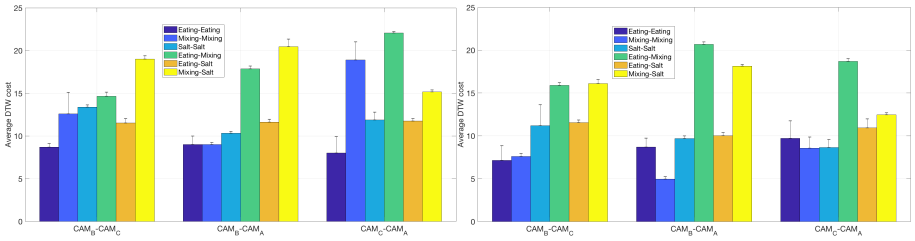
(d) CAM_B, Eating (e) CAM_B, Mixing (f) CAM_B, Salt

(g) CAM_C, Eating (h) CAM_C, Mixing (i) CAM_C, Salt

Figure 11: Average temporal profiles of different action instances. Each row corresponds to a view (CAM_A, CAM_B, CAM_C), while each column refers to an action (Eating, Mixing, Salt). The dictionary D_{ABC} is employed.

in the comparison may be of benefit. This aspect was left to investigate, so as a proof of concept, in Fig. 12b we consider only one profile, the green one in Fig. 11.

4. Is it really useful to learn an ad hoc dictionary for a given set of data? As a final investigation, we reason on the necessity of using data of the considered scenario. To this purpose we consider an unrelated benchmark (KTH [144]) showing full body actions. Fig. 9c shows the results obtained in this case. We notice a small degradation, but the overall performance is still acceptable. This spoke in favor of the potential of our space-time primitives to transfer knowledge between different settings.



(a) (b)

Figure 12: Comparison between descriptions from different views.

2.4.4 *Discussion*

We presented a work on representing actions through space-time primitives learnt from data. The preliminary results on a small subset of data include useful insights on how to proceed: the representation is rich and incorporates not only space-time corners but also other local structures with a significant dynamic information; the learnt atoms are quite stable across views, with strong discriminative power. The action representation is again quite stable across views, even if some actions seem to be intrinsically view-variant, and some views are more meaningful than others. Representations obtained from front and lateral views are very closely related, as expected.

2.5 Conclusion

In this chapter, we discussed the various types of classical methods that have been applied to Human Action Recognition. The types of features used and the steps followed by these methods have been considered to some degree. Moreover we introduced the works that have tried to tackle the problem of cross-view or view-invariant action recognition.

We discussed a method, that detects and describes spatio-temporal feature points using the Shearlet Transform and builds a representation from these features. We presented a preliminary analysis on its capability to discriminate between actions and potential for invariance to viewpoints.

Shearlet-based representation was really powerful in describing which spatio-temporal elements appear in the sequences we analyzed. Also, this representation showed some stability, giving us the ability to recognize patterns related to how these structures appear and disappear over time.The capability of the Shearlet decomposition to describe the direction and the magnitude character-

izing a movement exceeded our expectations, and opened the door for further developments.

The information that the Shearlet Transform provides has been demonstrated as very rich. One direction of investigation is to build a complementary machine learning module that can leverage this capability for an action recognition application. With the interest in deep networks growing, it was interesting to investigate a deep network pipeline that could leverage the potential of these features for view-invariant HAR.

Computing the Shearlet Transform of a video signal is an expensive operation, in terms of both memory space and time required. First, to calculate the Transform the entire sequence is required, since the calculation of the coefficients regarding a given point $m = (x, y, t)$ is carried out by both considering previous, current, and future information (i.e. frames in the sequence). Moreover, the whole video sequence has to be considered within each of these calculations, and since each operation involves the calculation of a few forward and backward 3D Fourier Transform (see [77] and the available code for details) it is trivial to see how the memory and space requirements of the computation explode. For the same reason, another drawback of this approach is that the information cannot be computed in real-time, because to calculate the information at time t both previous and future frames are needed.

Moreover, computing all the Shearlet coefficients for a whole video is excessively memory consuming. Thus, we had to slice long sequences into subsequences, a few seconds long. This is needed to contain the space required in memory to store all the results of the decomposition, making the calculation feasible. Nevertheless, this approach did not fit our requirement of HAR with limited resources.

For this reason, we decided to also search for other features that may have incorporated view-invariant properties but with less computational burden to the classifiers.

3

Deep Learning in Action Recognition

Deep learning methods have been instrumental in changing the landscape of applied machine learning in recent years. In this chapter, we discuss some basic concepts of deep networks and define relevant terms, developmental landmarks and state of the art of methods used in Action recognition and particularly Cross-View Action Recognition.

Section 3.1 provides an overview of general deep learning methods related terms and concepts in computer vision. We elaborate on Batch Normalization, in 3.2. Thereafter, we discuss Transfer Learning and Domain Adaptation methods in Section 3.3 and Deep Learning based methods and techniques for Action Recognition in Section 3.4. And lastly, with most relevance to the next chapters, the recent works in view-invariant and cross-view action recognition are detailed in Section 3.5 followed by conclusions. Relevant definitions and explanations have been included wherever possible.

3.1 Theoretical Overview of Deep Neural Networks

The basic building block of a neural network is a neuron, also known as a unit or a node. The internal structure of a node is shown in Figure 13. Multiple inputs, $X = \{x_1, x_2, ...x_n\}$, are fed into a neuron, j, which combines them to form a single output, O_j. The output of the neuron O_j is given by

$$O_j = \varphi(\sum_{i=1}^{n}(x_i * w_{i,j}) + b_j) \tag{10}$$

where:

$w_{i,j}$: weight associated with input x_i

b_j : bias for the neuron, also sometimes referred to as $w_0 x_0$ where $x_0 = 1$

φ : Activation function

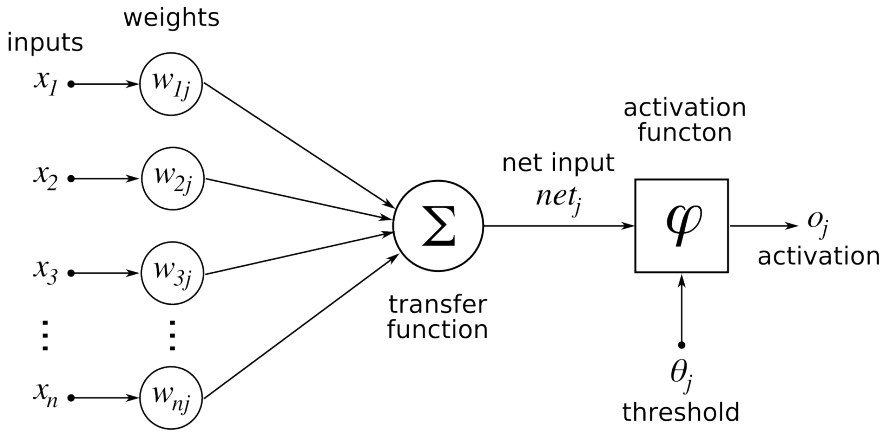


Figure 13: An artificial Neuron: the basic building block of a neural network.

The activation function is usually a non-linear transformation like sigmoid or ReLU functions. A variety of activation functions are used in various fields. If it is a thresholding function, the neuron can be termed as a perceptron, a binary classifier:

$$O_j = \varphi(\bar{x}.\bar{w} + b_j) \tag{11}$$

One or more neurons connected in parallel are together termed as a Single Layered Perceptron (SLP). Multiple such layers connected serially lead to a significant increase in the range of complex data that a model can learn, and is called a Multi-Layered Perceptron (MLP), shown in Figure 14. The internal layers, that are neither input nor output are termed as hidden layers.

Neural networks are trained using the process of Backpropagation. The goal is to minimize an error function, called the cost function or loss function, a classical choice being mean squared error (MSE) function:

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{12}$$

where:
y_i : target output
$\hat{y}_i$: estimated output
N : Number of input-output pairs in the training set
X : $\{(\bar{x}, \bar{y}), \ldots, \bar{x}_N, (\bar{y}_N)\}$ - N input-output pairs of training data
θ : parameters of the neural network

In an MLP, also termed as a fully-connected network, $\hat{y}_i$ is a function of inputs and weights of the final layer, which in turn are functions of inputs and weights of the previous layer, and so on until the first layer, and thus, $E = E(X, \theta)$. The minimization is done using the gradient descent method. A simplified version

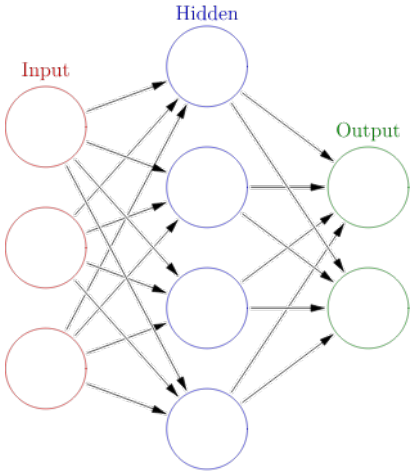


Figure 14: Fully Connected Neural Network, also referred to as a multi-layered perceptron.

of the error model is shown in Figure 15. The method iteratively minimized the function based on the assumption that for a function $f(x)$, the local minima is in the direction of its negative gradient:

$$f_{t+1}(x) = f_t(x) - \gamma \nabla f_t(x) \tag{13}$$

where γ is the learning rate or the rate at which the function moves towards the local minima and for any n^{th} layer in the network

$$\nabla E_n = (\frac{\partial E_n}{\partial w_{n,1}}, \frac{\partial E_n}{\partial w_{n,2}}, \ldots, \frac{\partial E_n}{\partial w_{n,l}}) \tag{14}$$

where $w_{n,l}$ denotes the weight at the l^{th} node in the layer. Hence weights are incremented by

$$\Delta w_{n,i} = \gamma \frac{\partial E_n}{\partial w_{n,i}}) \tag{15}$$

or as a whole,

$$\theta_{t+1} \longleftarrow \theta_t - \gamma \frac{\partial E(X, \theta^t)}{\partial \theta} \equiv \theta_t - \gamma \sum_{i=1}^{N} \frac{\partial E_i(X, \theta^t)}{\partial \theta} \tag{16}$$

The error propagates backwards through the network, giving the name of the method. This globally supervised learning procedure is the basis of training almost all neural networks today.

In 1998, LeCun et al. [84] proposed LeNet, a Convolutional Neural Network (CNN) with backpropagation and 5 layers. An example of a simple is shown in Figure 16. In a convolutional neural network, at least one layer uses convolution to calculate its intermediate values before applying the non linearity of the layer. The weights are composed of convolutional filters, also called kernels or

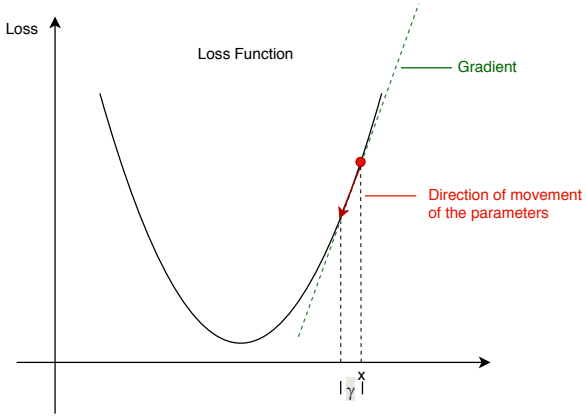


Figure 15: Gradient Descent Method

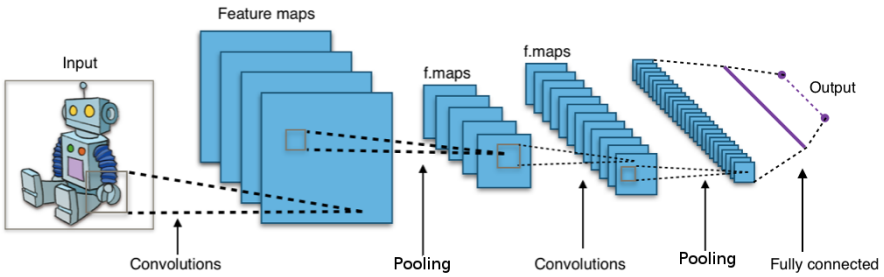


Figure 16: A simple example of a Convolutional Neural Network.

feature maps. The input propagates through the network by convolution with these filters.

Convolutional filters are usually much smaller than the input, thereby leading to less number of parameters, ensuring a *sparse interaction* and reducing the computation and memory requirements. Since the same set of filters are used for the entire input, there is *parameter sharing* farther reducing memory requirements. And the third advantage of adopting convolution in this context is that the convolution function is equivariate to translation. That is, if a component of the input is translated, the resulting output of the convolution would change accordingly. More precisely, a function $f(x)$ is equivariant to a function $g(x)$ if $f(g(x)) = g(f(x))$. These properties have been instrumental to the effectiveness of CNNs in computer vision applications.

LeNet also used what we now term as average pooling layers following each convolutional layer, reducing the size of the sample as it propagates through the network. LeNet used a Sigmoid activation function:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-1}} \tag{17}$$

The LeNet model was limited by the availability of computation resources at the time, thus limiting interest in the area for the following years. Meanwhile classical, feature-based computer vision approaches employing classic machine learning techniques were making rapid progress in many visual tasks including HAR (discussed in Chapter 2). They were feasible in terms of computation time and training data requirements, and explainable in their results. On the other hand, as compared to classical machine learning methods, neural networks contained large number of parameters to be learnt that required large amounts of data. It is only the feasible availability of GPUs in recent years that has allowed a more practical view at Deep Networks, breathing new life into the research.

Large datasets, like the prominent ImageNet [31] image dataset released in 2009, made possible the rise of deep nets to prominence. In 2012, Alexnet [74] participated in the Imagenet challenge. While Deep learning methods were developed and proposed in the intermediate term, Alexnet outperformed all other methods on the Imagenet challenge leaderboard by a margin. These were well established classical machine learning methods with proven effectiveness in the visual domain for preceding decades. This demonstrated the potential of Deep networks, at least to some extent. Thus began the chapter of Deep Learning and Neural Networks in Machine Learning and all fields to which it applies.

Alexnet was similar to LeNet in structure but with key differences. It had more layers and channels (filters per layer) thus extracting more information. It also used overlapping max-pooling, and ReLU (Rectified Linear Unit) activation function, eliminating negative activation:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{18}$$

Today, some of the vision based tasks where Deep Networks have made a remarkable difference are image classification [54, 59, 74, 151, 161], object detection [44, 137, 138], semantic segmentation [7, 18, 101], pose estimation [166], video classification [70] and optical flow estimation [33, 100]. Considering that the state of the art of deep learning is immense, in this chapter we will primarily discuss methods relevant to understanding the current state of methods addressing action recognition, with a special reference to cross-view action recognition.

Batch Normalization is an important component in Neural Nets today with an impact on our work. It is discussed in detail in Section 3.2. Some of the other important layers and components of CNNs relevant to this document, most of which are common practices today, are the following:

STOCHASTIC GRADIENT DESCENT (SGD) This is an iterative optimization method, a stochastic approximation of the gradient descent algorithm. With

the datasets becoming increasingly large, it is computationally inefficient to find the value of the loss function for the entire training dataset simultaneously. With this optimization method, the gradient is calculated *for each sample* and the model parameters are updated right away. Thus, Equation 16 is approximated to:

$$\theta_{t+1} \longleftarrow \theta_t - \gamma \frac{\partial E_i(X, \theta^t)}{\partial \theta} \tag{19}$$

This can lead to a erratic behaviour of the model weights. therefore, a common compromise is a 'mini-batch' Gradient Descent that updates the parameters with m samples at a time

$$\theta_{t+1} \longleftarrow \theta_t - \gamma \frac{\partial E(x_m, \theta^t)}{\partial \theta} \tag{20}$$

where x_m is a mini-batch of data with m samples and the quality of the approximation of the gradient over the mini-batch improves with increase in m.

DROPOUT Dropout is a regularization method wherein a percentage of randomly selected hidden and input nodes are removed or 'dropped' for an iteration, and the remaining network is trained regularly. Thereafter, these nodes are placed back with their parameter values unchanged from before and the next iteration begins.

SOFTMAX Usually placed at the final step of a single-class classification network, this layer transforms the output of the classification layer, usually referred to as scores, to values between 0 and 1 using the following transformation function, $\sigma : \mathbb{R}^K \longrightarrow \mathbb{R}^K$:

$$\sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{21}$$

for $i = 1, \ldots, K$ and $Z = (z_i, z_2, \ldots, z_K) \in \mathbb{R}^K$

INCEPTION MODELS A series of models worth mentioning are the Inception models [161] which employ internal blocks as some of the intermediate layers. The semantic representation of a block is shown in Figure 17. Each inception blocks comprises multiple parallel streams, each stream consisting of convolutional layers and, in one case, also a max-pooling layer, the results of which are concatenated at the end of the block. Subsequent upgrades of the model led to multiple publications [162, 163] and versions. An important model in action recognition that we will discuss in section 2.3, Inception3D [16] is based on the Inception (also called GoogleNet) model with 3D convolutions.

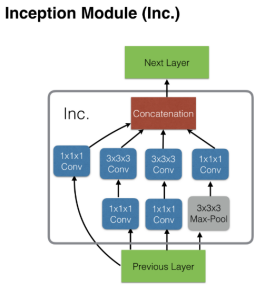


Figure 17: Inception Module

3.2 Batch Normalization

In practical machine learning applications, it is often the case that training data is not an appropriate representation of the final data that the methods are applied to *i.e.* the distribution of the training data does not match with the distribution of the test data. This problem arises because in practical applications, the data used finally can have a different distribution from the data that the developers used to train the model. This problem is referred to as *Covariate Shift* [61]. It has been studied at length in the field on machine learning and today is addressed by Domain Adaptation methods which will be illustrated in Section 3.3. With mini-batch training, the difference of the distribution of data also exists between mini-batches. Referring to the Equation 20, of the parameter update, it is worth noticing that θ would re-adjust with every iteration to compensate for change in the distribution of x_m, if such a change is present.

The structure of a feed-forward deep network, where the output of each layer being the input to the next layer, a change in the parameters of each layer affects the next, amplifying through the network. At the intermediate layers, this problem of change in the distribution of the activations due to the change in the parameters of the network, is called *Internal Covariate Shift*. To solve this problem, Ioffe and Szegedy [61] proposed the Batch Normalization (BN) Transform: Consider a mini-batch B with m samples. For a single activation x:

Input: Value of 'x' over a mini-batch: $B = \{x_{1,\dots,m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

$\mu_B \longleftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ // Mini-batch mean
$\sigma_B^2 \longleftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$ // mini-batch variance
$\hat{x}_i \longleftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \sigma}}$ // normalize
$y_i \longleftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$ // scale and shift

Where x_i is a single activation, μ and σ are the means and variance respectively and γ and β are the scale and shift parameters that are learnt during the training step.

During training, μ and σ are calculated from the mini-batch, termed local statistics. For evaluation, in order to make the method deterministic, the μ and σ of the entire batch, or training set are saved during the training process and used for new samples during evaluation. With the BN transformation, the inputs to each layer fall in the same distribution over successive mini-batches, allowing a smoother learning curve to the parameters, reducing the dependency of parameters of each layer on the parameters of other layers and encouraging more distinctive features. Incorporating Batch Normalization in an architecture combats vanishing gradients, demonstrates a regularization effect and faster convergence. It is now common practice to incorporate into every major layer of a neural network.

3.3 Transfer Learning and Domain Adaptation

Covariate shift in data can be reduced with investment of resources in collecting and labelling more appropriate data than the one available. On the other hand, the time and effort involved in such an endeavour can be very resource consuming, especially in case of deep learning. The necessity of this process can depend on the amount of existing training data, the extent of covariate shift and the approach. There are a variety of approaches that have been used to address and solve these problem, in particular:

- **Transfer Learning (TL)**: Re-purposing representations learnt for one task, for a different task. The similarity in datasets, type of data and the two tasks affect the usability of this method. A significant amount of training of the transferred representation on more appropriate training data, also called *fine-tuning*, can be required if the difference is high. In other cases, the representation may be used as-is or with minimal fine-tuning.

Thereby, this method reduces computation and data requirement by similar portions.

- **Semi-supervised/ unsupervised/ self-supervised learning**: Training on large amounts of unlabelled data to learn the representation. This method saves on the labelling process but still requires large amounts of data, (preferably of appropriate distribution) and the computational resources of end-to-end training. A detailed survey on this topic is presented by Chum et al. [22].

- **Domain Adaptation (DA)**: Using a different existing dataset to train. When the distribution of the source data is different from the target data, the gap between the distributions is bridged using a variety of machine learning methods. Resources required to obtain a large, labelled, appropriate dataset are saved, but computational requirements of the training remain unchanged.

- **Data Augmentation**: Generating the appropriate training dataset. To supplement the data scarcity, new data can generated synthetically, most often today using Generative Adversarial Networks [46] and augment any data already available. This method is usually employed when a medium amounts of data is already available but not enough to train the intended model. This data generation process can be very computationally expensive. One example in Action recognition is by Souza12 et al. [157].

To reduce the computation requirements of our own method and since the available dataset in cross-view action recognition tend to be small in size, we chose to explore Transfer Learning and Domain Adaptation which we discuss in the next section.

3.3.1 *Transfer Learning*

The potential of Transfer Learning (TL) in a variety of application domains emerged in the last decade, as the availability of large quantities of image data has been a fertile ground for deep architectures. Due to the significant effort required for gathering the right amount of data and training them, the use of pre-trained models has become a common practice – a classical choice in the computer vision domain, is a model pre-trained on ImageNet [74, 117, 119, 161]. The rationale behind this possibility is explained observing that different layers of these architectures embed different amount of information pertaining to the original task. While the last layers provide specific features – *i.e.* features strongly associated to the specific tasks – the initial layers exhibit the curious phenomenon of producing general features, i.e. features that resemble a common type of information regardless the specific problem and

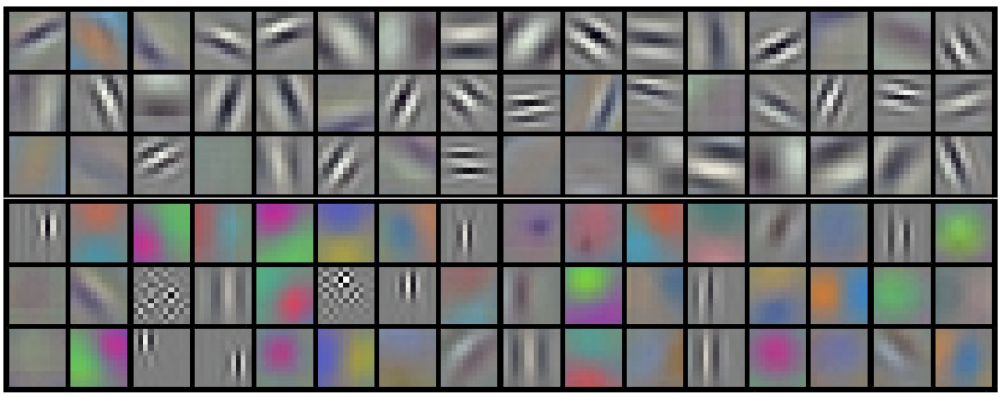


Figure 18: Convolutional Filters from initial layer of Alexnet [74] look like Gabor Filters and color blobs.

data they describe [200]. For example, filters of the initial layers of CNN architectures usually bare strong resemblance to Gabor filters and color blobs (see Figure 18). This suggests a transition from general to specific features should be present at some point of the architecture that, if appropriately used, may allow for *transfer learning* from one problem to another. The parameters θ can be separated into θ_g, generic parameters and θ_s, task specific parameters and one can choose to replace, re-initialize or fine-tune only θ_s for a new task. It is logical to reuse parameters instead of learning from scratch each time.

The extent of this transferrability from a pre-trained network to a new task depends on two factors:

i The similarity between source and target training data.

ii The similarity of the source and target tasks.

The more the similarities, the more advanced a representation can be transferred. Once these representation building pre-trained layers are re-purposed, it is common practice to train the remaining layers, and if possible, fine-tune the pre-trained layers on the relevant training data. If ample training data is available, pre-training is still often incorporated due of a regularization effect that is retained even after extensive training on top. A analysis on the above is available by Yosinski et al. [200]. Other relevant sources on Transfer Learning are proposed by Dai et al. [25] and Weiss, Khoshgoftaar and Wang [189]. For these reasons, and the scarcity of data in most multi-view datasets, we chose to use transfer learning in our work.

A study published in 2017 [52] demonstrates, with detailed experimental analysis, the capacity of features in terms of representing video data and the datasets that provide effective pre-training for action recognition tasks. The study and more on the topic of choosing pre-trained networks will be discussed in Section 4.1.

3.3.2 *Domain Adaptation*

In the case where the task of the data is similar but the distribution of the training data is different from the target task, and there is insufficient or no data available to train on with the target distribution, we turn to Domain Adaptation (DA). Transferring representation from one task to another often employs the use of DA techniques. When the source and target data is in two different feature spaces, this is termed as heterogeneous DA, *i.e.* the transfer function $t : \mathbb{R}^n \longrightarrow R^m$. If they exist in the same feature space, its called homogeneous DA, *i.e.* $t : \mathbb{R}^n \longrightarrow R^n$. Visual tasks are usually considered to be homogeneous DA since different environments, background, illumination, viewpoint, sensor or post-processing can cause a shift between the train and test distributions but the space is common. Meanwhile moving from text to images or video to speech or even from images to videos would fall in the category of heterogeneous DA. Some surveys detailing more on classical DA for visual tasks are by Csurka [23] and Patel et al. [122] while one on deep visual Domain Adaptation is by Wang and Deng [184].

Recently, with the dawn of deep networks, a number of DA different approaches have surfaced, to deal with features from these networks. Discrepancy based methods [117, 200] advocate that fine-tuning can remove the domain shift and are very commonly used, in Transfer Learning applications and Deep Networks today. Inversely, adversarial methods train a classifier adversarially to a domain classifier, leading to features that are discriminative to the task while being indiscriminate to the domain [42]. One example of this in cross-view action recognition is [88]. Generative Adversarial Networks generate feature level [99] or pixel level[150, 199] instances to train networks for target domains. Reconstruction based methods try to generate a shared representation between the source and target domain.

In case of cross view action recognition, we choose to use no data at all from the target domain, or the target viewpoint. This limits our options considerably. We take inspiration from a homogeneous method similar to the work by Li et al. [91], adapting it to a scenario where normalization statistics of the entire target data is not available to the classifiers.

3.4 Deep Learning based Action Recognition

Some of the most recent and successful action recognition methods use Deep Learning layers either for part of the process or end-to-end learning. Copious amounts of data is a basic requirement in training complex deep networks. With video input this tasks becomes more computationally expensive and re-

Table 3: Some Recent and Popular Benchmark Datasets in Image and Action Recognition

Dataset	Type	Total Classes	Total Samples	Size in Gigabytes	Year of Release
ImageNet [31]	Images	21,841	14,197,122	250	2009
MS COCO [94]	Images	80	328,000	18	2014
Kinetics-400 [71]	Videos	400	306,245	400	2017
ActivityNet [39]	Videos	200	28,108	600	2016
UCF101 [156]	Videos	101	13,320	6.5	2012

quires more data. The ImageNet dataset provided this in the image domain. The success of the image domain has become possible to replicate in the video domain only very recently, with a significant emphasis on action recognition. Some of the largest and most important image and action recognition datasets are mentioned in Table 3.

Other large scale datasets, Sport-1M [70] and Youtube-8M [2] have been proposed but due to noisy annotations and inexact trimming of video clips, they have been put to limited use by there research community and therefore, we will not be mentioning them hereafter. ActivityNet and Kinetics provided availability of a wide variety of data of appropriate size to distill transferable information, though deep models had already been applied to action recognition with improvements with respect to state-of-the-art [55, 151, 167, 183, 193] employing the previously existing datasets.

3.4.1 *Actions as 2D+t volumes*

The first category of action recognition methods we discuss deal with videos as a 2D+t volume, forming in-effect a 3D input where one dimension is time. This is a classical choice as we discussed in Section 2.2.

The image related tasks like object recognition have been using 2D convolutions in their architecture extensively. In action recognition, the temporal progression is a vital aspect of the information provided by a video slip. The spatial information can be complemented by the temporal information. Therefore considering the input a 2D+t volume, it is a natural extension to extend the 2D operation to 3D convolutions and leverage the advantage of a convolution function in the temporal axis. The idea of 3D convolutions extracting spatiotemporal features appeared earlier [6], one of the the first models to use

it for end-to-end training was by Ji et al. [65]. A recent detailed survey of the state of the art of action recognition has been presented by Zhang et al. [204].

Convolutional3D descriptors (C3D) were proposed by Tran et al. [167], which built a design of a 3D CNN architecture, combining appearance features with motion information, and training on RGB input. On a similar track, Sun et al. [159] applied the factorization methods to decompose the 3D convolution kernels to 2D spatial and 1D temporal kernels, and used the spatio-temporal features in different layers of CNNs. Long-Term Temporal Convolutions [170] explore long temporal convolutions of varied temporal resolutions to capture the full scale of actions.

Another type of popular architectures is the two-steam CNN which takes RGB frames and pre-computed optical flow as inputs to two different streams, the spatial and temporal streams, and combines the final scores, a late fusion technique, to classify actions. The first work to propose a two-stream network [151], used single RGB frames in the spatial stream and a trajectory representation loosely inspired by Dense Trajectories (iDT) [178] for the temporal steam built directly on optical flow. On the other hand, Wang, Qiao and Tang [182] combined the original iDT (discussed in Section 2.2) with two-stream CNN networks to build the TDD (Trajectory-Pooled Deep-Convolutional) descriptors. Feichtenhofer, Pinz and Zisserman [40] used the architecture proposed by [151] and made an in-depth analysis of a number of fusion techniques for two-stream network in order to leverage the temporal correspondence between the spatial and the temporal stream.

Stemming from the popular inception models [161], and using the two-stream method, Inception3D (I3D) [16] has also made a name for itself. To build this architecture, they used the inception architecture with Imagenet pre-training and inflated the 3D convolutional kernels to 3D before training in on the Kinetics dataset. A late fusion was used for the two steams but each stream took a 3D volume as input. The input to the spatial stream was the RGB video as a volume while the temporal stream uses TVL1 optical flow [125]. They also choose to use a 3D Convolutional layer for the classification instead of a fully-connected layer, allowing for flexible input sizes.

Temporal Segment Network (TSN) [183] used a multi-stream approach. It used the BN-Inception [61] (discussed in section 3.1) as backbone to build a two-stream networks. It fed snippets of the input video into parallel instances of these two-stream network (which share parameters between same modality streams) with a single frame of the snippet fed into the spatial stream and the optical flow into the temporal stream of each instance of the network. A two step fusion was employed, the first for the scores of actions from same streams of all instances and the next for the scores of the two streams, leading to an action classification. On a very different approach, [157] use a generative approach with the backbone of TSN with BN to augment the lack of data by generating synthetic action sequences.

Using a single frame of the video input in the spatial stream of a two-stream network is a frequent choice. It allows the system to observe the context of the action as well as potentially allow objects in the scene to contribute to recognition of the action while the amount of processing during the evaluation in increased too but only by a fraction when compared to the temporal stream. The trade off is usually acceptable. But since only a single frame is chosen, this choice can be detrimental to the process during long actions. TSN remedies this by using a frame for every clipping of the sample but only to some extent. Inception3D uses the entire RGB video volume even in the spatial stream. This increases performance at the cost of increased computation both during training and deployment.

Other works have explored analysis of these networks not only with change in the point of fusion of the streams [40] but also method of fusion. [121] proposes a method of using optical flow to amplify the features from the spatial stream to improve the results the original two stream network [151].

Some recent surveys on similar methods can be found in [5, 193, 197].

3.4.2 *Actions as Time Sequences*

Another category of neural networks are the Recurrent Neural Networks (RNN), which treat data as time sequences. RNNs employ feedback loops or 'memory cells' that store the information of the previous state and pass it on at the next step(s). RNNs [95] have accomplished success in many fields that involve time series like image captioning, speech synthesis, and music generation, musical information retrieval, natural language processing and so on. In action recognition, the input to an RNN is usually a hand-crafted or deeply learnt feature. For example, Baccouche et al. [6] used a convolutional feature extractor, using 3D convolutions, to extract features that were fed into an RNN network. RNNs have been used as an intermediate step in a weakly supervised scenario [139] or for fine-grained action detection [152].

The memory capacity for these networks, though, tends to be limited. The gradient of the loss function decays exponentially with time, allowing the influence of an instance within the time sequence to vanish quickly, making it difficult for these networks to learn long term dependencies. Long Short Term Memory (LSTM) networks [56] mitigated this problem by having an additional component in their 'memory cells', allowing them to retain longer sequences of information. This property allowed them more effective than RNNs for action recognition applications. The input to LSTMs are usually features too, extracted from other methods. For example, Yue-Hei Ng et al. [201] used features extracted from GoogleNet [161] for action recognition and [89] pro-

posed a multi-stream method where individual LSTMs are associated with each stream.

While LSTMs have shown promise, the memory capacity is still not long enough for these methods to be competitive with CNN based state of the art methods for action recogniiton today. Thus we decided to use CNN based methods in our work.

3.4.3 *Semi-Deep methods*

Deep networks learn representations from raw data that can be employed with more conventional classification methods.

In [34], a Spatio-Temporal Vector of Locally Max Pooled Features (ST-VLMPF) is proposed wherein deeply representations are extracted as features and then processed using prior knowledge to build an dictionary encoding that is used for classification. While the goal of the paper was to propose the encoding, it also compares 3 different representations, extracted from different types of networks: The spatial (individual frames) and temporal (optical flow) networks [151] process the frames individually so the extracted features for each frame are concatenated, to begin with, while the spatiotemporal model [167] treats the input as a volume. This work, while showing better performance than similar methods of the time, also demonstrates that the temporal stream, with optical flow inputs consistently outperformed the other two feature extractors. In our work, we chose to use only a temporal stream with optical flow input, for the action recognition task.

Another work that builds on deeply learnt features is by Girdhar et al. [43] where they employ the two-stream network [151] and extract the features from the two streams separately, then using their proposed ActionVLAD layer (inspired by VLAD [63]), conducts a pooling to build a vector that can be used for the final classification. In addition, they also compare a variety of possible points of fusion of the streams as well as the a comparison between the two streams. The method shows that a late fusion and optical flow stream obtain the highest accuracies.

3.5 Cross-view Action Recognition

The challenge of Cross View Action Recognition (CVAR) is to train the action classification method on a limited number of viewpoints and recognizing

actions from other, unseen viewpoints with no explicit knowledge of the relationship between different views.

CVAR calls for multiview datasets with simultaneous recording to actions from multiple viewpoints. Additionally, this area also attracts works that employ depth and/or skeleton information due to their inherently view-invariant properties. Benchmark datasets IXMAS [187] and NUCLA [180] provide volumetric and skeleton information respectively. More recently, NTU RGB+D dataset [147], a large scale dataset, has facilitated more research by providing a large amount of data for deep networks to train on in addition to multimodal data comprising depth maps, motion capture and even infrared data.

For this reason, many recent methods use 3D skeleton to solve the cross-view problem [28, 72, 93, 96, 98, 147, 149, 180, 203]. Others still use depth images [15, 132, 133]. Some methods use depth data to transfer information to a 3D canonical view [132, 133]. Li et al. [88] use scene flow [171], sometimes referred to as 3D optical flow derived from RGBD data using the Primal-Dual framework [62] with an adversarial training loss on view classification in addition to an action classification loss to improve the view-invariance of the features. Variants of LSTMs have also been used [96, 147] with these modalities. Liu and Yuan [98] extended a pose estimation method to recognise actions. A survey on depth based methods was written by Liang and Zheng [92] and another by Aggarwal and Xia [4] and another specifically on skeleton based methods as well as on how these skeleton features are calculated, is offered by Han et al. [51].

On the other hand, our own interest is focused specifically on the use of RGB videos for the CVAR problem. We discussed the classical approaches to this scenario in Section 2.3. Few methods on the same lines have been proposed in deep learning. Codebook based methods have continued to be used with deep classifiers [73]. Baradel et al. [10] proposed glimpse clouds exploiting 2D interest points to build an attention mechanism and used pose supervision during training. Baradel, Wolf and Mille [9] extended an attention mechanism implemented with RNNs. Wang et al. [175] use a multi-stream network to extract view-specific features, requiring a computationally expensive training from multiple views.

3.6 Conclusion

In this chapter we presented background information on deep learning techniques that are relevant to our work and its understanding. Additionally, we discussed the state of the art of methods used in action recognition and cross-view action recognition based on the deep learning paradigm.

Action Recognition has been explored by a wide variety of methods and each has its advantages and disadvantages. Incorporation of the temporal component has been vital to the task. Deep Learning approaches have shown great potential in the area. Batch Normalization is shown to reduce the effect of internal covariate shift, also leading to faster convergence, therefore reducing the computational resources required to train a model. An appropriate choice of pre-trained model can be instrumental in reducing the resources required, in terms of the computation involved during training as well as the size of the training set. We considered all of these factors while designing our methodological pipeline.

Unlike the majority of approaches, that incorporate either 3D or multimodal information, our method is purely based on videos, ensuring a wider applicability potential and a potentially reduced investment in infrastructure. Considering the observations made on the available methods and literature, our work exploits the concept of transferring information from one view to another, leveraging the transferrability potential of pre-learnt deep features.

PART II

Deeply Learned Features for Cross-View Action Recognition

This part of the document contains the contribution of the thesis comprising the method used, the experiments conducted and the results. We discuss the results, outlining important observations and the significance of these observation for the research community at large.

4

The proposed methodology

Our primary goal is to design a Human Action Recognition (HAR) pipeline tolerant to view-point changes, constituting sustainable components that require limited resources in terms of computation and training data. Within the scenario of Deep Learning, this requires a strong commitment towards sustainability, but allows the leveraging of existing resources. To this end, we adopt a transfer learning approach, exploring the appropriateness of features trained for other action recognition tasks with no explicit focus on view invariant properties. Our hypothesis at this stage is that these trained representations may implicitly incorporate the relevant information which can be leveraged for cross-view action recognition.

We propose a two steps method detailed as follows:

i First, we represent the input video sequences by computing mid-level features extracted by a network pre-trained on large-scale source datasets.

ii Then, we design an appropriate classifier, and train it on the target dataset.

As we will highlight later in the chapter, view-invariant properties are not explicitly taken into consideration for either, the choice of the architecture, or in the selected source dataset. The richness of the pre-trained features associated with an ad hoc classification procedure allow us to capture view-invariant elements in the action recognition model we design. Domain Adaptation methods, specifically batch normalization are incorporated in the classification to boost the robustness to view-change.

In this chapter, we will discuss the detailed components of the methodological pipeline and the factors that influenced the design choices at each step. Sections 4.1 and 4.2 are dedicated to step (i), discussing the choice of features and feature extraction process. Step (ii) is tackled in Sections 4.3 and 4.4, discussing the factors that influenced the classification step and the different classifiers

that we compare in our study. Section 4.5 discusses implementation details of the model.

4.1 The pre-trained model

The potential of transfer learning has been largely exploited in the images-based areas [189], while there have been fewer contributions in videos-based fields. This is due to the fact that for spatio-temporal data, the available pre-trained networks are fairly recent. There are a number of interesting models that have been proposed that can be used to extract features like the 2-steam network[151], Trajectory-pooled Deep-convolutional Descriptors (TDD) [182], Convolution 3D (C3D)[167], Pseudo-3D ResNet (P3D) [129], Inception3D (I3D) [16] and Temporal Segment Networks (TSN) [183].

Model	Kinetics	UCF-101	HMDB-51
2-stream I3D (I+K) [16]	**74.2**	**98.0**	**80.7**
2-stream I3D (K) [16]	71.6	97.8	80.9
C3D [167]	-	82.3	-
P3D [129]	-	88.6	-
TDD [182]	-	90.3	63.2
TSN [186]	-	94.2	69.4
2-stream CNN [71, 151]	62.8	88.0	59.4
2-stream CNN(I) [71, 151]	65.6	91.2	58.3

Table 4: Top-1 Accuracies of various networks architectures on 3 benchmark datasets. Works presenting the particular accuracies are cited. Letter in () indicates pre-training. I: ImageNet. K: Kinetics

When choosing a pre-trained network, two factors are to be considered: architecture and source dataset.

Architecture: The model should be capable of modeling the data in accordance with the complexity of the task.

Source dataset: The dataset should be able to effectively train the model without overfitting.

There are a number of potential candidate architectures that have been used in HAR applications. Although some of these networks incorporate 2D convolutions like 2-stream network [151], TDD [182] or TSN [183], we prefer to focus an architectures that employ 3D convolutions. Intuitively, 3D-CNNs extract spatio-temporal features from input data which should perform better on

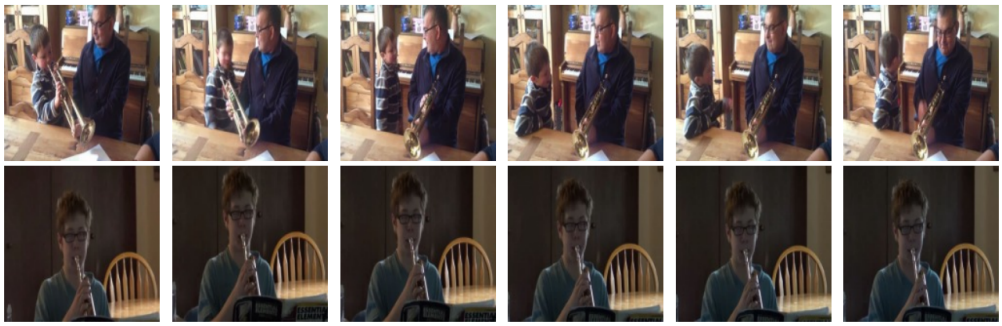
videos than spatial features. Quantitatively too, 3D CNNs have shown better results on benchmark datasets [52, 71]. Within methods using 3D convolutions, as well as overall, I3D demonstrated best accuracies, both for Kinetics as well as when finetuned on other datasets (with Kinetics pre-training). This is quite easily demonstrated by the results in Table 4. In addition to the final accuracies, I3D encodes the data without cropping (unlike TSN and its derivatives or C3D) and is independent of input length which can be instrumental given the intra-class temporal variations in HAR. Thus we chose I3D as a starting point for our pipeline.

On the topic of the source dataset, we bring to the reader's notice a study conducted by Hara, Kataoka and Satoh [52] comparing 4 action datasets that are widely used currently: UCF-101, HMDB-51, ActivityNet and Kinetics. They found that out of these 4 datasets, Kinetics dataset is the only one that could deeply train a model for action recognition task, in this case a ResNet-18 architecture, without overfitting. The architectures that we consider for feature extraction have more learnable parameters than the ResNet-18 and it stands to reason that Kinetics pre-training would be the most effective means to our end. Frames from sample videos are shown in Figure 19. Although the variations in a large scale dataset are difficult to demonstrate with a few examples, these sequences exhibit change in scale, illumination, environment and background, visible parts of the actor and most importantly for this work, camera angular view-point. Thus it is likely that a model can implicitly develop cross-view correlation properties with training on Kinetics dataset.

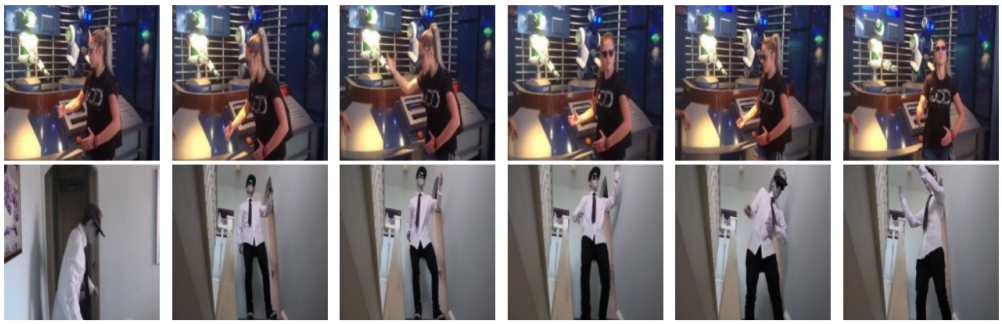
Inception3D, optionally, first leveraged a pre-training of the 2D kernels on ImageNet, before the kernels are inflated to 3D and trained on Kinetics dataset. This provides even richer feature and a further regularization effect. Therefore, we choose the Inception3D network with both ImageNet and Kinetics pre-training. I3D is a two-stream network, with RGB and optical flow as inputs to the two identically structured (and separately trained) streams and a late fusion. A single stream of the model is shown in Figure 20. For more details, we refer the reader to Section 3.4.1.

Optical Flow (OF) representation captures the motion in a video stream. In HAR, the region of the scene involved in the motion, and the motion itself are often the main focus of any analysis. Optical flow isolates the motion, thus allowing the temporal network a head start towards analyzing the motion and removing the clutter in the space. Additionally, variation in color of objects or clothes and skin or any variation in background are not incorporated in the representation, improving the generalization ability of the stream. The spatial stream adds the scene information to the network, adding value in terms of shapes, objects, background and colors to the analysis. Many works involving two-stream networks [16, 151, 170] concur that the temporal or OF stream gives better results than the spatial stream but the two combined have a slightly better performance than either. We conducted preliminary tests which concurred that the discriminatory power of features extracted from the RGB stream was

(a) dribbling basketball



(b) playing trumpet



(c) robot dancing

Figure 19: Samples of actions from the Kinetics dataset

significantly lower than that from OF stream, for the multi-view tasks we considered. Incorporating two streams may add accuracy but increases (and depending on the method even doubles) the resources employed in training as well as evaluation. Thus, with the motivation of limited resources in mind, we choose to only extract features from the optical flow, which are not only more discriminative but also less prone to overfitting, as stated by Carreira and Zisserman [16]. We also demonstrate the validity of this choice by a relatively straight forward experimental analysis between the two streams of the I3D network in the next chapter, hereby ensuring that this choice is optimal for our task.

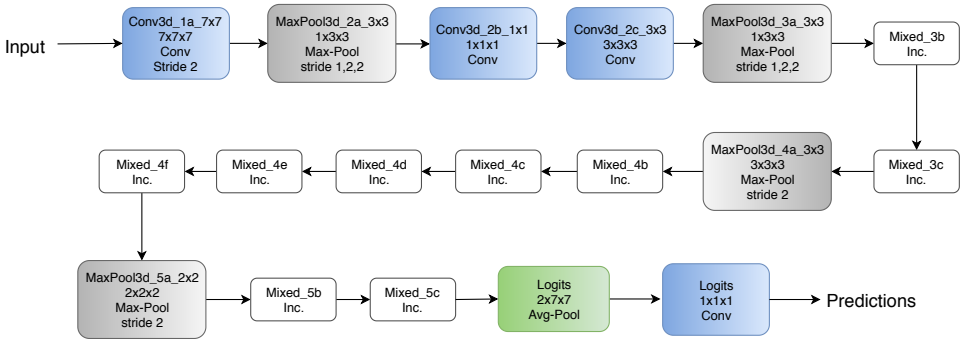


Figure 20: Layout of a single stream of the Inception 3D.

4.2 Feature Extraction

The point at which features are extracted from the identified pre-trained network, is decided based on information from the studies discussed in Section 3.3: similarity of the tasks and the source datasets of the original pre-training of the network and the target application. In order to pin-point an appropriate point of extraction of the representation from the network for a new task, we examine the tasks and the datasets and make the following observations:

i The original and target tasks are activity and action recognition, respectively.

ii The source dataset (Kinetics) and target datasets may differ in the temporal length and types of actions in consideration.

iii The lower level data structure share similarities (indoor environments and biological movements that comprise target datasets are included in source datasets), while the high level information (scale and view-point as well as specific actions and objects involved) may change considerably.

For these reasons, mid-high level features seemed appropriate for our task. A late point of extraction appeared to be the best compromise on the selection of a representation which was as high level as possible, while retaining a good transferrability power. In order to obtain an optimal pipeline for diverse datasets, we conduct a comparative analysis between three features extraction points, after 12, 15 and 17 layers of the 18-layer-network described in 20:

- 12: After layer named 'mixed_4e'
- 15: After layer named 'mixed_5b'

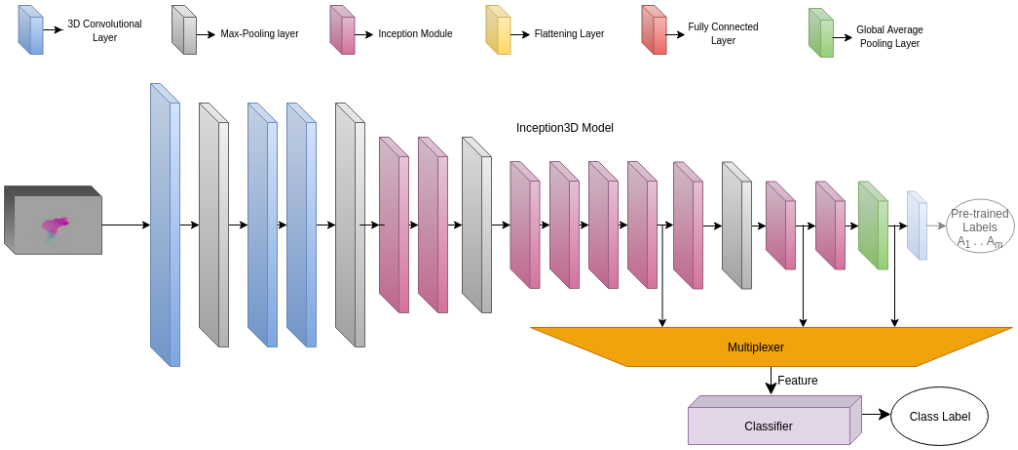


Figure 21: Feature extraction comparative analysis: features from 3 different points are fed into the classifier(s)

Table 5: Size of the features extracted at different points of the network for a 60 frame input.

Layer Label	Numerical position [0,18]	Feature dimension $(t_x, s_{1x}, s_{2x}, f_x)$	Total number of values
Mixed_4e	12	(16,14,14,528)	1,655,808
Mixed_5b	15	(8,7,7,832)	326,144
Avg_Pool	17	(3,1,1,1024)	3,072

- 17: After layer named 'Avg-Pool'

Figure 21 shows a more summarized version of the model, with the extraction points indicated along with the remaining pipeline. These features differ in their discriminatory power as well as the size, which would be have to be considered in the comparison.

In I3D, features extracted for any given input sample and any given layer x are formed by 4 different components $(t_x, s_{1x}, s_{2x}, f_x)$: s_{1x} and s_{2x} values correspond to the spatial size, t_x corresponds to the temporal size, or *feature frames*, and the f_x corresponds to the channels of feature points extracted. Input spatial sizes (width and height) are kept equal throughout our study, such that $s_{1x} = s_{2x} = c$ with c a constant value for features at a certain depth. f_x is independent of the input and depends only on the layer of the model from where the features are extracted. Instead, t_x depends on the number of frames in the input. Value of the *feature frames* vary for each sample. The quantitative details of the three different points of extraction considered are summarized in Table 5.

4.3 Batch Normalization: Support for cross-view recognition

Batch Normalization (BN) [61] is a frequently employed in training Deep Nets today not only for its primary function to minimizing the internal covariate shift between mini-batches but also reducing the number of epochs required by a network to train as discussed in Section 3.2. Internal Covariate Shift is the phenomenon of change in the distribution of a layer influenced by the change in the distribution of the previous layer.

We found that there exists a covariate shift between the I3D features derived from different viewpoints. To demonstrate this, we construct a simplified representation of the distribution of data. In this representation, for action sample feature set:

$$X = \{x_i^j \mid i = (1, \ldots, n), j = (1, \ldots, v)\} \tag{22}$$

with n samples and v views, we calculate μ_i^j and σ_i^j, the respective mean vector and variance vector of the sample. Given x_i^j of size (t_x, s_x, s_x, f_x), μ_i^j and σ_i^j are vectors of size (f_x). We keep the statistics of the different channels separate, and calculate the mean and variance over all activations in a channel. This is done because the information associated to a channel can be different from another and not directly comparable. To make the representation more stable, p samples from the same viewpoint are concatenated into a single datum. Thus, a single datum, z_k^j, of the set Z, is a vector of p samples from the same viewpoint:

$$z_k^j = \{\mu_1^j, \sigma_1^j, \ldots, \mu_p^j, \sigma_p^j\} \tag{23}$$

for $k = (1, \ldots, K)$ The dimensionality of the Z set is reduced using the t-SNE dimensionality reduction method [106].

The resulting representation for two datasets, IXMAS and MoCA are shown in Figure 22. For these plots, $p = 10$, and the data is dimensionally reduced by t-SNE [106] to 3 dimensional plots. IXMAS and MoCA have 11 and 20 classes each and the color labelling is associated with the viewpoints of the data points. It is important to note here that dimensionality reduction leads to loss of information. Therefore, this representation is a simplified view of a highly complex set of features. With that in mind, note the separation between the data-points from different views.

Representative samples of the datasets are shown in Figures 26 and 28. The separation of the distribution is more distinct between viewpoints that are visually more different at the video level. Note the distinct positioning of samples from view 4 for dataset IXMAS in Figure 22a, which represents a top view and positioning of samples from the view 1 for dataset MoCA in Figure 22b, which represents the egocentric view. Both of these viewpoints have

a distinct prospective of the actions with respect to the other viewpoints in the dataset. Since it is evident that the representation, in fact, has some view-specific information, the classifier designed for these features would need to compensate for this shift.

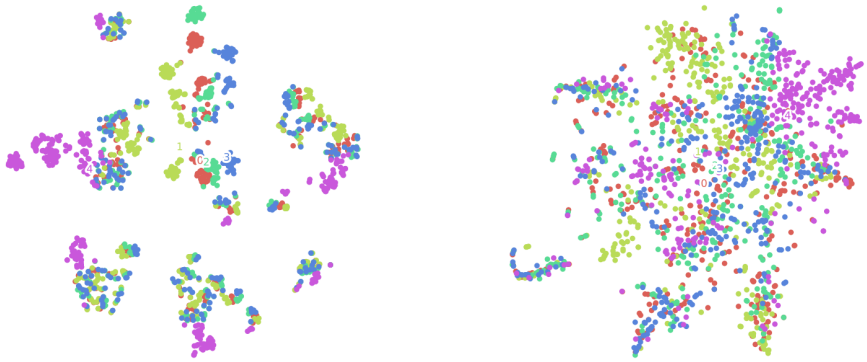
(a) IXMAS (b) MoCA

Figure 22: Dimensionally Reduced representation of concatenated mean and variances of samples from (a) IXMAS and (b) MOCA datasets. The numbers refer to the viewpoints and they are located at the position of the centroid of the respective viewpoints.

Batch Normalization has been effective for internal covariate shift during training, thus it can stand to reason that it can also be helpful to solve the problem of covariate shift in this case. In fact, BN has been applied in the context of Domain Adaptation by Li et al. [91]. They proposed an Adaptive Batch Normalization layer in an image context, to bridge the gap between the feature space by normalization with statistics of the entire target data during evaluation. We employ a similar method, with key differences, to boost the capacity of the classifiers to deal with completely unseen viewpoints.

We propose the Modified Batch Normalization (Modified-BN) for the classification layers. During the training step, the BN functions conventionally as defined in [61], with local statistics, *i.e.* mean, μ_{x_m} and variance, σ_{x_m} calculated for each mini-batch x_m of the training batch X_t, and learning the scale γ and shift β parameters over the training set. During the evaluation, instead of using μ_{X_t} and σ_{X_t}, we propose normalizing with local statistics on the evaluation mini-batches. This is done due to the variation in the distribution of data from different view points. This choice of normalization parameters allows the data to reduce the covariate shift, present due to difference in viewpoint, hence promoting transferability between views, especially in case of widely different viewpoints [91].

To demonstrate the phenomenon and the potential, we propose the following visualization: Figure 23 shows the dimensionally reduced representation of the features, obtained with t-SNE [106], labeled according to their viewpoints. On the left, Figure 23a shows the original features, on the right, in Figure 23b shows the samples after they have been normalized in mini-batches of 10 samples from the same view, with the shift, $\beta = 0$ and scale, $\gamma = 1$. This



(a) (b)

Figure 23: Dimensionally Reduced representation of samples from IXMAS (all classes), (a) originally and (b) after normalization by batch. Different colors denote samples from different viewpoints.

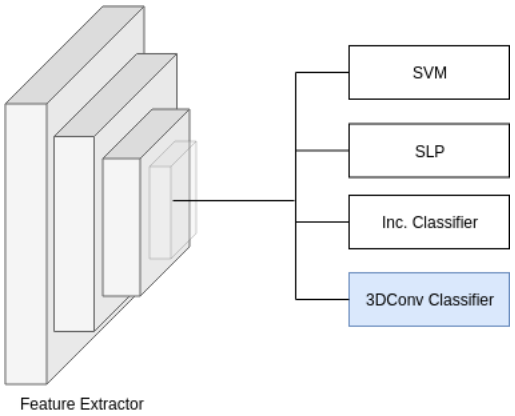


Figure 24: Structure of the comparative analysis: classifiers of different complexity are paired with the pre-trained features.

suggests that the separation between the data from various views is reduced with the normalization, when normalized with mean and variance of the same mini-batch.

The clips in a single mini-batch do not need to necessarily belong to the same viewpoint for this method to add a positive boost.

4.4 Classification

The next step of the pipeline, is to classify the features extracted from the deep network. To this end, we propose a classifier: the Cross-View 3D Convolutional classifier (CV-3). Moreover, putting it into prospective, we propose a comparative analysis with a variety of classical and deep classifiers. A semantic representation of this step is shown in Figure 24. All classifiers that can incorporate a learnable normalization in their design, incorporate the Modified BN layer that we have proposed (in Section 4.3), in order to improve the quality of the comparisons. Our choice of the classifiers has been deliberate, with each contributing a significant information to the understanding of the data and the problem. The simpler classifiers gauge the ability of the representation itself. The more complex ones gauge the potential of the representation, testing the limits of information that can be leveraged from the features.

THE CROSS-VIEW 3D CONVOLUTIONAL CLASSIFIER (CV-3): We consider the extracted feature vectors as a reference representation and designed a classifier, appropriate for cross-view data scenario with, primarily, three 3D Convolutional layers. The advantage of this classifier is in its simple design, providing a balance between minimizing resources required for training, in terms of data, time and computation power versus providing sufficient depth and width to build a mapping from the features to the accurate classification, while incorporating 2 BN layers.

The CV-3 classifier is described in Figure 25. It consists of 3 convolutional layers, each employing 3D convolutions, with reducing number of channels over successive layers, thus leveraging both spatial and temporal positioning of information that is preserved in the features. The first two 3D Convolution layers incorporate batch normalization, a ReLU activation and are followed by average pooling layers. Thus the pooling layers ensure a reduction of the the size of the activations with successive layers. The BN layers, as discussed in Section 4.3, promote view-invariance by reducing the BN shift. The final 3D Convolution layer, instead, lacks these trailing transformations and directly results in a vector of values with the same size as the classes in the dataset that are put through a softmax function resulting in the probability values associated to the classification. This 3D convolution layer acts as a classification layer, a role usually played by a fully connected layer. However, we noticed that the replacement did not affect the results but reduced parameters and computations.

INCEPTION CLASSIFIER (INC): This classifier provides a natural baseline in terms of a complex classifier to be compared with our proposed structure. It consists of the set of layers from the Inception3D networks that remain beyond the feature extraction point. Thus for the 3 different points

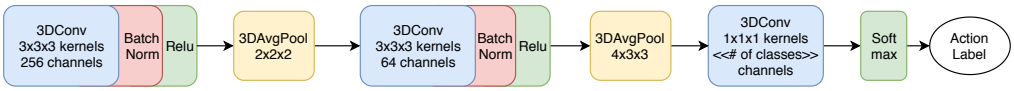


Figure 25: Layout of the 3D Convolutional Classifier.

of feature extraction, this classifier differs, employing between 6 to only 1 layer. On the other hand, since the number of parameters can be too large for the small cross-view datasets to train, it had to be allowed to leverage the ImageNet and Kinetics pre-training for the Inception modules in the classifier when they are present.

SINGLE LAYERED PERCEPTRON (SLP): This is the simplest neural network based classifier that we have incorporated in this study. It consists of a single fully connected layer, followed by Batch Normalization and a Softmax non-linearity to obtain the classification probabilities. This classifier provides us with an understanding of the information that the representation readily provides to obtain cross-view action classification.

SUPPORT VECTOR MACHINES (SVMS): Another addition to the analysis is a popular classic classifier, the Support vector Machine, SVM. An SVM is a machine learning algorithm that constructs hyperplanes that provide maximum linear separation between samples from different classes. We employ an SVM with a linear kernel as well as a Gaussian kernel for the comparison.

4.5 Implementation details

We first estimate optical flow with the TV-L1 algorithm [202] and feed it into the I3D model for feature extraction. The implementation of the deep architectures relies on TensorFlow [1] and pre-processing has been applied according to [16]. The features are obtained from the pre-trained I3D model with flow network checkpoint trained on both ImageNet [31] and Kinetics [71] datasets. All models have been trained on a single 32 GB GPU machine with Mini-batch Gradient Descent on batches of size 10. The learning rate was decreased by a factor of 10x at fixed intervals of epochs. Dropout of 0.5 was used before each layer during training. Evaluation was done in batches of 10 with locally calculated batch normalization parameters. All classifiers are trained from scratch.

For all classifier architectures, we considered constant value $t_x = T$. For samples with initial temporal size, or frames in an instance, smaller than T, we looped the sample to make the size correlate. For samples with temporal size greater than T, clips of size T are extracted from the sample at each epoch during

training. Video clippings with less than 8 frames had to be dropped from analysis due to limitation of the feature extractor. For all experiments presented in this thesis, $T = 8$ feature frames which corresponds to 60 frames of the input sample. During evaluation, the central feature frames are cropped in order to create the batches for the modified BN. This is necessary because creating a batch requires all instances in a batch to be of the same length. For experiments employing the original BN, instead, a moving window was executed along feature frames of each sample and the class with the highest average score across the sample is selected as the result.

4.6 Conclusion

In this chapter, we presented a pipeline to train a network that is robust to cross-view action recognition that requires limited resources in terms of training data and computation. We dealt with the possible options for pre-trained models and source datasets they could be trained on for this pipeline and described the decision making process that led to the choice of Inception3D architecture with the Kinetics pre-training, for the feature extractor. We also discussed our motivation and method of choice for the specific point of the feature extraction from the architecture.

Once extracted, these features have to be classified. To that end, we analysed the features and outlined a requirement of the classifier, on the lines of its ability to deal with covariate shift. Next we discussed the classifiers that facilitated the analysis of these features, the most important of which is our proposed classifier for this method. In all, we proposed 2 major comparative analyses, one on the feature points and the other on an array of different classifiers. We also propose a comparison between the two stream (spatial and temporal) of the Inception3D model, to support the results of previous studies.

Our pipeline is in accordance with the aim of employing limited resources. The use of a pre-trained network as a feature extractor has been instrumental in reducing the resources required by the method. Despite this step, the effectiveness of the method is preserved by the design of the classifiers. The results of the comparisons, and of our HAR pipeline are presented comprehensively in the next chapter.

5

Experimental Results and Analysis

In this chapter, we present results of the experiments conducted during the course the study. Building on these results, we discuss their implications towards the primary questions that we ask in this study. We also examine the relationship between viewpoints based on observations derived from these experiments.

Section 5.1 introduces the chapter in detail. Section 5.2 presents details about the datasets employed during the course of this work. From there, we move to the preliminary testing. Section 5.3 presents the preliminary results demonstrating the functionality of the methodological pipeline in a simple action recognition scenario. Section 5.4 presents the results of the experiment conducted in order to identify the optimal choices for the effectiveness and efficiency of the pipeline.

Thereafter, we present results that allow for insightful observations related to the process as well as the relationship between the viewpoints. Sections 5.5 and 5.6 show the results of the two primary training scenarios that we have considered in this work, Single View Learning and Multiple View Learning. Section 5.7 presents the resource usage of our method. Lastly, Section 5.8 discusses the implications of these results.

5.1 Introduction

The primary goal of the work remains the design of a system capable of effective cross-view action recognition with limited resources. The secondary aim, on the other hand, has been to study and understand the relationship between viewpoints with respect to an action. This requires a study of a diverse set of actions in diverse environments and from diverse viewpoints. Large scale datasets usually contain a large set of viewpoints since they are usually video clips taken from user generated data like Kinetics [71] or Youtube-8M [2] dataset.

But a systematic study on such datasets, for our secondary aim is not practical. Therefore, we employed a set of diverse set of multiview action datasets.

The different datasets also allowed us to demonstrate the capability of our method under different circumstances. They vary in setups and backgrounds, number and type of viewpoints, action classes, number of subjects, the visible section of the body (upper body or full body), occlusions, and the number of instances available for training. Each dataset adds value to the analysis with its particular characteristics. These characteristics also make them more appropriate for certain modalities, which also facilitate building a better understanding of these actions. In our work, we consider two modalities:

Single View Learning: We propose this learning task as a benchmark to access a method's capability to extract view-invariant information about actions from samples collected from a single view point. Hence the training split consists of data from a single viewpoint, and is tested on the other viewpoints.

Multiple View Learning: The more frequently use learning scenario in Cross-view learning tasks, samples extracted from more than one viewpoint are used to train the classifiers. This scenario tests a classifier's capability to incorporate information from multiple viewpoints in order to build a richer representation that it uses to classify the target view.

Some of the datasets are more appropriate for one of these tasks than the other on account of their particular characteristics or because of how they have been used by other works, in the past. These are discussed in more detail in Section 5.2.

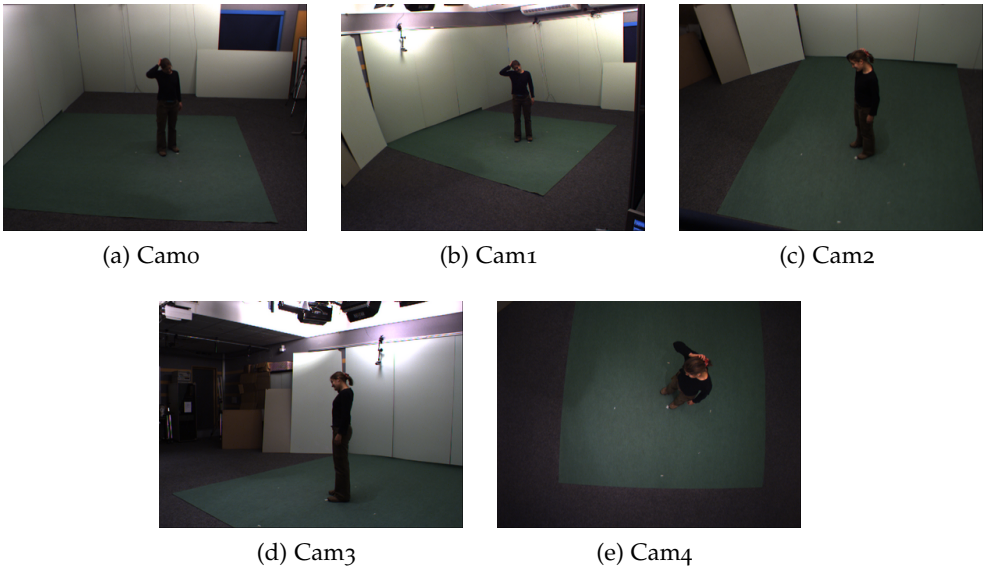
5.2 Datasets and protocols

In this section we discuss the details of the 4 datasets that we adopt in our work, and summarize their key properties in Table 6. Each dataset included serves a particular purpose and added a different type of analysis to the work. Some of the datasets were more appropriate for a certain modality of training and testing due to the nature of their data. All of these datasets incorporate information other than video, or data acquired from other sensors, which are mentioned in this section. This information is used by other methods that are presented in the comparative analysis presented later in this chapter. In agreement with the main motivation of this thesis, we only use RGB video data in our own pipeline.

Table 6: Datasets with multiview visual data. FB: Full Body. UB: Upper Body. IR: Infra Red. Sil: silhouettes. All datasets are captured indoors with controlled illumination

Dataset	Classes	Views	Subjects	Total Clips	FB/UB	Data
IXMAS [187]	13	5	10	1650	FB	RGB, 4D Sil
N-UCLA [180]	10	3	10	1500	FB	RGBD
NTU RGB+D [147]	60	5	40	56880	FB	RGBD, Pose, IR
MoCA [114]	20	3	1	1500	UB	RGB, Pose

IXMAS



(a) Cam0 (b) Cam1 (c) Cam2

(d) Cam3 (e) Cam4

Figure 26: Sample frames from the IXMAS dataset (actor *alba* and action *scratchhead*).

The INRIA Xmas Motion Acquisition Sequences (IXMAS) [187] is a benchmark multiview dataset with 10 subjects and 11 actions: *check watch, cross arms, scratch head, sit down, get up, turn around, walk wave, punch, kick* and *pickup*. The videos are recorded from 4 wide viewpoints (camo-cam3; see Figure 26a-26d) and 1 top view (cam4; see Figure 26e). The environment is indoor, uncluttered and with controlled illumination and minimal background variability. Considering the viewpoints, this dataset can be considered representative of a surveillance task scenario. Each actor repeats each action 3 times. Most video sequences are longer than 60 frames and therefore, randomly selected clips of sequential frames were extracted from the sample for each iteration during training. The dataset also provides 4D silhouettes, in addition to the videos, of the action instances.

(a) V01 (b) V02 (c) V03

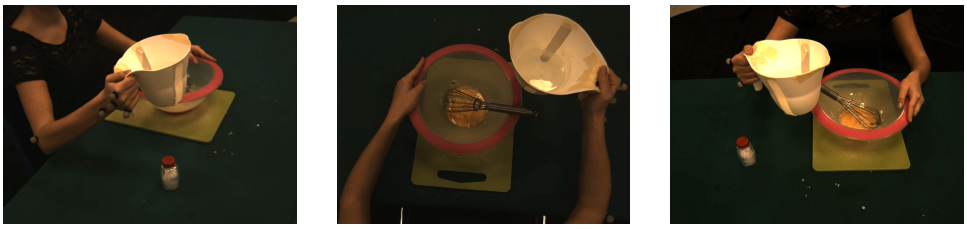
Figure 27: Sample frames from NUCLA dataset (actor *s06* and action *pick up*).

The 5 distinct viewpoints of IXMAS make it an extremely interesting case study for the single view training task. With this scenario, a study and understanding of the relationships between these viewpoints is enabled. On the other hand, 30 samples per action are used for the training which we consider as the limiting case of minimum training samples for the classifiers.

NUCLA

The NorthWestern-UCLA Multiview Action3D dataset [180] is a multi-view, multimodal dataset including video, depth and skeleton data of 10 action classes: *pick up with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw* and *carry*. The videos are captured from 3 RGBD Kinect sensors. Frames from the 3 sensors are shown in Figure 27 along with their corresponding view labels. Notice the background of the scene, the bookcase, which would lead to significant disturbance in methods that employ only spatial features unless the space of interest is localized using temporal cues. We characterize this as a cluttered environment. The viewpoints can be considered similar but, in fact, can have significant occlusion between them as in case of the sample shown. With these viewpoints and the distance from the actor, this dataset is a satisfactory representation of a human-human or a human-robot interaction with an observational motivation in consideration. The action sequences are often shorter than the 60 frames, and thus the sequences are looped for the training process.

It is interesting to note that in this dataset, some actions share significant similarities – e.g. actions that are actually sub-parts of other actions, as in the case of *carry* and *walk around* – while others can be interpreted as inverse pairs, eg. *donning* i.e. putting on a piece of clothing, and *doffing* i.e. taking off a piece of clothing or *sit down* and *stand up*. This dataset is primarily used in the multiple view learning scenario, wherein 2 of the 3 views are used for training and the remaining one for testing. Therefore, about 100 samples are available for training each class.



(a) View0: lateral (b) View1: ego (c) View2: frontal

Figure 28: Synchronized samples from the MoCA dataset.

MoCA

The Multimodal Cooking Actions Dataset [114], is a multiview dataset that we have acquired in-house with the motivation of understanding motion analysis skills and view-invariance properties of both biological and artificial perception systems. The dataset comprises of 20 upper body actions with an average of 25 clips per class available per view, though the exact number of clips per class vary among the classes. The environment indoors, controlled and uncluttered. The cameras are focused on the actions, thus limiting the influence of any background cues or cues stemming from body parts that are not involved in the action directly. The dataset also provides motion capture data of the dominant arm of the subject. The actions are cooking and food-prepping actions executed on a table-top. They are relatively fine-grained, *e.g.* mixing and whipping actions are separated only by the angle of the motion and speed at which they are performed. Some actions are periodic, like mixing or grating in contrast to others, like transporting or pouring, ensuring diversity.

The primary and most important reason for addition of this dataset into our study is the opportunity to study the egocentric view, allowing us to build an understanding of the how different is it from an allocentric (non-egocentric) viewpoint in terms of a learnt representation. The 3 viewpoints of the dataset are shown in Figure 28. Notice how different the egocentric view is from the allocentric ones. When the action is seen from these viewpoints, the motion of the actions is perceived in drastically differently making this a very interesting case to study. This dataset is used for both single and multiple view learning scenarios, allowing a more complete analysis of the method as well as the dataset.

NTU RGB+D

The NTU RGBD+ action recognition dataset [147] consists of action videos of 60 actions by 40 distinct subjects. The actions include 40 regular single-actor

Figure 29: Samples of the different viewpoints in the NTU dataset for a variety of actions.

actions, 9 health-related actions and 11 interaction-related action involving two actors. 3 cameras are used for the recording, placed at 0°, 45° and 90° as shown in Figure 29. A subject performs each action twice, once facing the first camera and the other time facing the third camera. Hence 6 video clips are available for each action by a subject, 2 for the 0°, or front facing, and one each for -90°, -45°, +45° and +90°. This is the largest dataset incorporated in our work, a new benchmark employed by the latest cross-view action works. Its popularity stems from two reasons: (i) it is the first multiview dataset large enough to train a complex action recognition deep network from scratch, and (ii) it provides a variety of data modalities, other than videos: body-joints, depth maps and infra red data. This has paved way for a large influx of action recognition methods leveraging these modalities. On the other hand, our own method was conceived for small datasets, but this dataset allows us to compare the capability of our method with a much larger number of instances and classes, even when used used out-of-the-box and a comparison with some of the state of the art methods.

A single standard training and test protocol is available for this dataset for the cross view scenario, wherein samples from 0° and ±90° are used for training and samples from ±45° are used for testing. This is similar to a one-view-out protocol but has the advantage of using the extreme viewpoints for training and the intermediate view for testing. This would fall in the category of multiple view training. In addition to this, we propose an analysis wherein we use a single angle (0°, 45° or 90°) for training and evaluate the trained model on the other two individually, a protocol more similar to the one-one scenario but still with the advantage of the data from two sides when training on 45° or 90°. In this case, the training on 0° would be considered in the single view training scenario.

5.3 Pre-trained features transferability

As the baseline experiment, we test the transferability of a pre-trained representation to a new dataset and new classes. We use the representation learnt on

Table 7: Baseline evaluation of the pre-trained features on the IXMAS dataset (training and testing on the same view, the *one-subject-out* protocol).

Method	Cam0	Cam1	Cam2	Cam3	Cam4	μ	σ
CBP features + MMM-SVM [192]	82.02	85.74	85.54	**89.10**	69.50	82.38	6.82
I3D features + SLP	**88.49**	**87.28**	**87.22**	83.67	**83.89**	**86.11**	**1.96**

Table 8: Performance evaluation (in %) on the MoCA dataset. Views - 0: Lateral, 1: Egocentric, 2: Frontal

Source|Target	0|0	1|1	2|2	Mean
I3D + SLP	93.25	91.11	92.70	92.35
I3D + Inc.	96.25	96.35	96.43	96.34
I3D + CV-3	**98.65**	**99.21**	**99.13**	**99.00**

the base dataset, Kinetics, and test it on 2 different multi-view target datasets, IXMAS and MoCA, within an action recognition scenario. To this end, with IXMAS we follow a very simple *one-subject-out* protocol. A classifier is trained and tested on one view at a time, such that with n subjects in the training set, $n-1$ are used for training and the remaining for testing. The experiment is repeated n times, once for each subject as test samples and the average over all tests is presented. Limiting the scenario to a single view at a time allows us to compartmentalize such that we do not consider the cross-view challenge and simply derive a baseline evaluation of our method on a dataset as small as IXMAS. Our aim is to focus on the features and their representative power, thus we use a simple SLP for classification. The classifier would be trained on 297 samples in this protocol for each experiment. The small size of the training set allows for a limiting case in terms of the smallest training set.

To produce a comparative analysis, we were only able to identify one paper in the literature that used the dataset with the same one-subject-out protocol [192]. The obtained results, reported in Table 7, show the appropriateness of pre-trained features in comparison with a representation based on Correlogram of Body Poses (CBP) which was specific for the purpose. Our average results are superior and more stable even with a simple linear classifier, with a higher mean accuracy and a lower variance across views.

With MoCA we take a slightly different approach. The dataset has a separate training and test set for data from each view. This allows for a simple experimentation protocol with samples from all actions, and only one view in an experiment, following the training and test data splits of each action a directed by the dataset. We used three deep classifiers, the SLP, the remaining part

of the inception model, which we are calling the inception classifier and the cross-view 3D convolutional (CV-3) classifier. This is the first experiment that shows a comparison between the effectiveness of the three classifiers, though in a simple action recognition scenario. The results are shown in Table 8. Note the high values and consistency of each of the classifiers.

With these experiments, we demonstrate that the pipeline and the three deep classifiers are reasonably effective in a action recognition scenario. For these experiment, we used the representation from the 'mixed_5b' layer and only the Optical Flow (OF) stream. The choices here can be considered arbitrary since the aim is to establish that the method functions, while optimality is not a focus. The comparison between the different parameters involved in the design of the model are presented in the next section.

5.4 Experiments guiding the architecture design

The main choice we need to make in terms of the feature extractor is the most appropriate feature extraction point. In addition to this, although we already found sufficient proof in the literature in favor of the optical flow stream [16, 151, 170] since we prefer to choose only one of the two available streams, we still present an empirical study to add more consolidation based on results from some of the specific datasets we have employed in this work. We also need to demonstrate the advantage of the modified BN with respect to the regular BN transformation. For all of these, we carry out a set of comparative analysis on the IXMAS and NUCLA datasets with the various parameters.

Table 9 reports a comparison of raw RGB and OF performances, with regular and adaptive BN, considering classifiers of different complexity. We fix the feature extraction point to 'mixed_5b' for this set of experiments. In order to improve performance in the regular BN case (without modified-BN), each sample is processed separately, and completely, instead of being randomly cropped. Instead, since the model was trained on a fixed size of clips in terms of feature frames, hence during evaluation, a moving window was used, with a stride of one feature-frame. The outcome is the label with the maximum sum of scores, values after the Softmax operation, across the sample.

A few observations can be made from this table:

- OF provides consistently superior performances with respect to RGB.
- Modified Batch normalization usually leads to significant performance improvements and confirms its importance as a tool to support transferability. It is particularly useful on IXMAS, where a *one-to-one* protocol is used with more different viewpoints, compared to NUCLA where *one-*

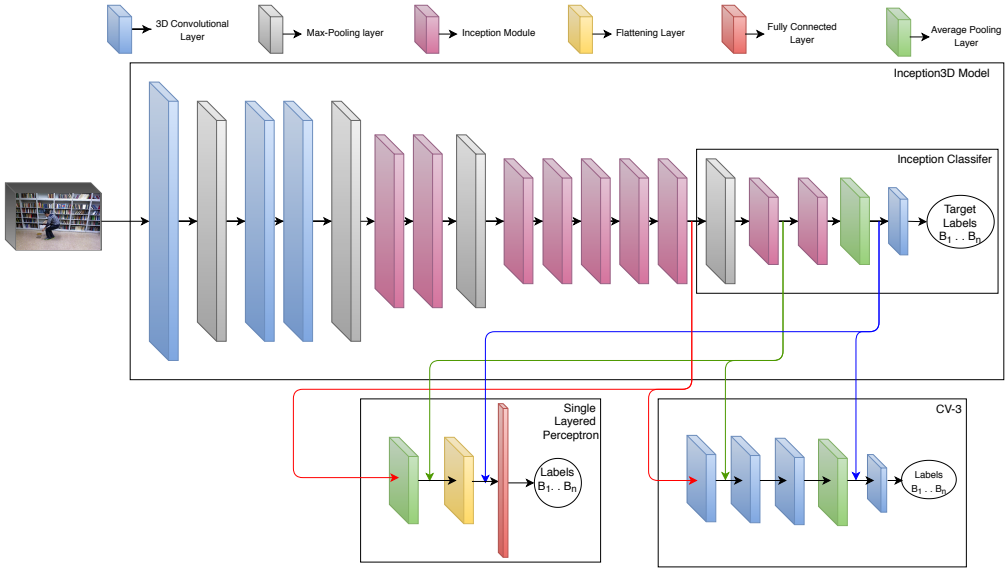


Figure 30: Feature extraction comparative analysis: features from 3 different points are fed into 3 different classifier(s)

view-out protocol is used over more similar viewpoints, allowing classifiers to accumulate more information about the actions from the training set.

- The performances of the more complex classifier are not consistently higher than the simpler one, in particular if modified batch norm is used; it appears that the combination of pre-trained features and the modified-BN is more crucial to achieve good results, especially when a single view is used in training the classifiers, in case of IXMAS dataset.

- The gap between regular and modified BN is most noticeable for the CV-3 classifier which is a simple classifier by design considering deep classifiers but leverages the higher number of BN transformations to improved recognition performances when incorporating modified BN.

Based on these experimental findings, we use OF input representation and the modified BN method for our pipeline.

Table 10 report a comparison of the three different feature extraction points listed in Table 5. The parameters of the comparison remain similar to the previous case in terms of the datasets. OF stream and the modified BN are used, for all results presented in this table. We remind the reader that the inception classifier here refers to the remaining layers of the inception architecture beyond the respective points of extraction and are marked in Figure 30. These layers leverage the pre-training on Imagenet and Kinetics for the inception modules while the other two classifiers, the SLP and CV-3 are trained from scratch, though their sizes also change according to the input stream as discussed in detail in the previous chapter. Note that the in case of the average

Table 9: Performance evaluation on the IXMAS and NUCLA datasets considering the *one-one* and *one-view-out* protocols respectively with regular and modified Batch Normalization methods used during evaluation, with RGB and OF streams. The table reports average percentage accuracies (in %) over the different training and test combinations within the respective protocols.

Classifier	Stream	IXMAS		NUCLA	
		Reg BN	Mod BN	Reg BN	Mod BN
I3D + SLP	Raw RGB	46.1	62.5	13.75	36.91
	OF	48.1	**69.4**	66.06	**67.84**
I3D + Inc.	Raw RGB	48.2	58.5	15.76	30.16
	OF	65.4	**68.5**	**78.38**	75.17
I3D + CV-3	Raw RGB	33.8	66.2	19.18	40.42
	OF	45.1	**78.4**	55.7	**79.12**

pooling layer, the inception classifier is the same as the CV-3 classifier and thus the presented results are identical as well. Major observations are as follows:

- Overall, CV-3 has the best performances for both datasets (marked by the red box). Within the results from each classifier, though, the performances associated with the different extraction points are more conflicting.

- *Avg_Pool* rarely provides the best results under any variables implying that the representation at this point is too specific to be generalized well on the new datasets. The highest performance with the SLP classifier can be due to this high level of the information.

- The high performances of the *mixed_4e* features in two of the 6 cases is quite easily explained by the increased number of classification layers involved by Inception and CV-3. The SLP performed poorest with the *mixed_4e* features because of the increased length of the flattened feature vector with a reduced level of the information.

- The middle layer, *mixed_5b* performed the best in 3 of the 6 cases, once with each classifier. In the remaining cases, with the modified BN, the result is not significantly lower than the best performance of each case.

Considering the trade-off between the size of the feature (recall Table 5) which can directly affects the number of parameters in the classifiers and the data requirement and the resources used in training, vs the difference in accuracies, the intermediate extraction point $\texttt{mixed}_5\texttt{b}$ is the optimal candidate, which will be adopted in the remaining experiments.

Table 10: Comparative analysis of the features extracted from 3 different extraction points in I3D, using the IXMAS and NUCLA datasets considering the *single view learning* and *multiple view learning* scenarios respectively. Presented accuracies (in %) represent the average of all combinations of training-evaluation splits within the respective scenarios. The red box outlines the best performance within results from each dataset.

Classifier	Extraction Point	IXMAS		NUCLA	
		Reg BN	Mod BN	Reg BN	Mod BN
I3D + SLP	mixed_4e	22.67	59.2	27.67	62.93
	mixed_5b	48.1	**69.4**	66.06	67.84
	Avg_Pool	45.45	61.3	**69.50**	65.98
I3D + Inc.	mixed_4e	64.18	**70.0**	16.83	78.32
	mixed_5b	65.4	68.5	**78.38**	75.17
	Avg_Pool	61.76	62.2	77.78	77.18
I3D + CV-3	mixed_4e	19.56	68.6	19.46	**79.43**
	mixed_5b	45.1	**78.4**	55.7	79.12
	Avg_Pool	61.76	62.2	77.78	77.18

Table 11: Performance evaluation on the IXMAS dataset considering the *single view learning* scenario. Each column refers to a different *Source*| *Target* pair. In brackets we report the reference to the paper from which we extracted the performance of the corresponding method, when the original publication of the method does not report results on IXMAS with the relevant protocol.

S|T	0|1	0|2	0|3	0|4	1|0	1|2	1|3	1|4	2|0	2|1	2|3	2|4	3|0	3|1	3|2	3|4	4|0	4|1	4|2	4|3	Mean
DT [178]	93.9	64.2	81.8	27.6	87.6	66.4	75.2	22.4	70.0	83.0	73.9	53.3	75.5	77.0	67.0	34.8	42.1	25.8	63.3	48.8	61.7
Hank. [87]	83.7	59.2	57.4	33.6	84.3	61.6	62.8	26.9	62.5	65.2	72.0	60.1	57.1	61.5	71.0	31.2	39.6	32.8	68.1	37.4	56.4
DVV [90]	72.4	13.3	53.0	28.8	64.9	27.9	53.6	21.8	36.4	40.6	41.8	37.3	58.2	58.5	24.2	22.4	30.6	24.9	27.9	24.6	38.2
CVP [208]	78.5	19.5	60.4	33.4	67.9	29.8	55.5	27.0	41.0	44.9	47.0	41.0	64.3	62.2	24.3	26.1	34.9	28.2	29.8	27.6	42.2
I3D + SVM-Lin	94.2	84.5	70.0	43.3	94.2	83.6	65.5	47.3	82.7	77.0	90.0	60.0	35.5	25.8	76.7	26.1	43.6	38.2	69.4	53.0	63.0
I3D + SVM-RBF	91.8	81.2	71.8	42.7	87.0	72.4	47.3	38.5	78.8	66.7	77.6	54.5	38.3	33.9	71.2	33.3	43.9	40.0	61.2	47.9	59.0
I3D + SLP	84.4	80.3	79.2	48.6	87.4	77.6	72.1	47.0	79.6	78.6	83.0	65.9	72.1	72.0	98.3	45.0	**53.3**	**56.6**	69.3	53.5	69.4
I3D + Inc.	88.6	76.7	84.0	44.4	87.4	70.6	79.4	42.5	78.9	77.9	80.8	61.2	86.0	85.4	77.3	45.4	48.6	49.1	57.9	46.7	68.5
I3D + **CV-3**	**97.1**	**92.7**	**94.6**	**50.3**	**95.4**	**85.6**	**92.8**	**47.5**	**88.9**	**86.8**	**95.3**	**77.5**	**91.6**	**90.5**	**94.9**	**54.4**	49.4	52.3	**73.7**	**56.5**	**78.4**

5.5 Single View Learning Problem

This section considers the cross-view recognition problem with single source view to train the classifiers. The scenario of single view learning is important to consider since gathering or searching for data from a single viewpoint is substantially simpler than from multiple view, for training. Add the requirement of synchronized multiple view data and the task of data acquisition and annotation becomes many times more resource consuming. We propose the Single View Learning Problem to boost efforts in the direction of building systems that can be trained with minimal amount of effort investment in gathering training data for a specific system. Such a scenario would also be relevant in a robotic learning scenario where a robot could effectively learn an action despite viewing it from a single viewpoint.

To this end, we employ 3 datasets, IXMAS, MoCA and NTU RGBD datasets for the analysis. IXMAS has been used by other methods in a similar way and allows a comparison with the other methods. The scenario of training on a single viewpoint and testing on another has been termed as the *one-to-one* training protocol but the single view training scenario is a complex learning scenario that facilitates an interesting paradigm to study the relationships between specific viewpoints.

The classifiers have access to samples from a single viewpoint during training. Thus, the classifiers have no opportunity to discriminate between the information provided by the features based on view dependence or invariance. In case of our pipeline, this allows for a gauge of the view-independent information in the features and the ability of the classifiers to adapt to the target or test viewpoint based solely on these features. The single view learning also motivates a better understanding between viewpoints and different factors that can affect the relationship between them at the feature level.

The performance of the different classifiers on IXMAS, MoCA and NTU, along with a comparison with previous works in case of IXMAS, are reported in Tables 11, 12 and 13. The last row in each table reports the results obtained by our proposed architecture, demonstrating superior performances.

For comparison with the performances of the IXMAS dataset, we particularly focused on methods that only employed video sequences (neither alternative nor additional sources), and were specifically meant to address cross view action recognition. The lower part of Table 11 reports results obtained by our chosen features paired with a variety of classifiers. As it can be observed, the pre-trained features provide on average better performance than other state of the art methods. Note that all the deep classifiers, i.e. the SLP, the inception classifier (remaining layers of the inception model after the feature extractor) and the CV-3, employ the modified BN, thus boosting performance. The two

Table 12: Performance evaluation (in %) on the MoCA dataset with the *single view learning* scenario. Views - 0: Lateral, 1: Egocentric, 2: Frontal

Source|Target	0|1	0|2	1|0	1|2	2|0	2|1
I3D + SLP	47.38	68.33	47.38	32.86	66.27	34.84
I3D + Inc.	50.63	64.84	33.10	36.35	61.67	54.92
I3D + CV-3	**54.84**	**79.52**	**65.71**	**69.52**	**88.20**	**61.83**

Table 13: Performance evaluation (in %) on the NTU dataset with the *single view learning* task.

Source|Target	0° | 45°	0° | 90°	45° | 0°	45° | 90°	90° | 0°	90° | 45°	Mean
I3D + SLP	58.94	48.61	58.39	57.92	50.08	59.41	55.56
I3D + Inc.	67.19	54.87	69.00	68.81	55.10	66.94	63.65
I3D + CV-3	**74.08**	**57.94**	**76.78**	**75.82**	**60.27**	**75.39**	**70.05**

SVMs, on the other hand, do not. Yet the SVMs are at par with the best classical method we found. This analysis speaks in favour of the fact the some amount of view-invariance capability is embedded in the representation itself. Similar trends are followed by MoCA and NTU datasets as well, with CV-3 performing better than the inception classifier which, in turn, performs better than SLP.

Next we focus on the relationship between the viewpoints. The fact that the representation and the method are not strictly view-invariant becomes evident with these tables. Association is higher between closer viewpoints, with low relative occlusion. This association falls when the angle decreases. In Table 12, it can be noted that the recognition capability of all the classifiers are higher for the lateral and frontal cameras. However, this association is relatively weak between them and the egocentric view. This is explained by the fact that at the video level, actions look very different from the egocentric view when compared to the allocentric counterparts.

The trend is similar for the NTU dataset, as shown in Table 13. The relationship between 45° and the other two angles is stronger than between 0° and 90°. The performance is relatively more consistent in this case than in the egocentric view of MoCA where the action can look relatively more different. We would take this opportunity to point out to the reader that this is not the standard split that is employed on the NTU dataset. This training and test scenario has not been implemented by previous works either and we propose this more challenging scenario as a tool to improve the estimate of the capability of a system that targets view-invariance in action recognition. While IXMAS allowed the method to be tested for a small dataset, NTU demonstrated the

Table 14: Average performances of Table 11 grouped per view, i.e. considering all the percentages obtained when a certain view was either in training or test.

Method	C_0	C_1	C_2	C_3	C_4
DT [178]	67.8	66.4	67.6	66.8	39.8
Hankelets [87]	59.7	59.9	65.0	56.3	41.2
DVV [90]	44.7	45.6	31.2	42.0	27.3
CVP [208]	50.0	49.3	34.7	45.9	31.0
I3D + SVM Lin.	68.5	65.7	78.0	55.3	47.6
I3D + SVM Gauss.	66.9	59.7	70.5	55.7	45.3
I3D + SLP	73.0	72.0	79.1	71.9	54.9
I3D + Inc.	74.3	72.6	72.7	73.1	49.5
I3D + **CV-3**	**82.5**	**81**	**86.9**	**83.8**	**57.7**

effectiveness of the overall pipeline even though no changes were made in the kernel sizes of the hidden layers of the classifiers, despite a large change in the number of classes from 11 to 60. The resulting recognition accuracies are still relatively high. Results from the standard protocol of NTU are presented and discussed in the next section.

A view-centric analysis quantitatively also demonstrates the uneven quality of the different views. The percentages obtained when a certain view of the IXMAS dataset was involved either in training or test, are averaged and presented in Table 14. Thus, this table gives a gauge of the strength of the relationships between a viewpoint with the rest of the viewpoints. Cam4, the camera viewpoint from top, looking down at the actor (see Figure 26), shows lowest accuracy, which can be due to multiple reasons.

i Cam4 has a significantly different angle from all the other viewpoints, leading to very different way in which the action is perceived.

ii The particular position of the Cam4 also inducing major occlusions, which, in some cases, can lead to very small amount of information to be available for recognition purposes.

iii The feature extractor is trained on Kinetics which consists of user generated actions videos, and tends to be biased towards human-human interaction viewpoints. The content from viewpoints similar to cam4 would be limited. Thus the feature itself may be relatively poorer in terms of high level information.

Moving to an action-centric perspective, we propose in Figure 31 (a) a comparison of the average recognition rates each action achieves across all the views

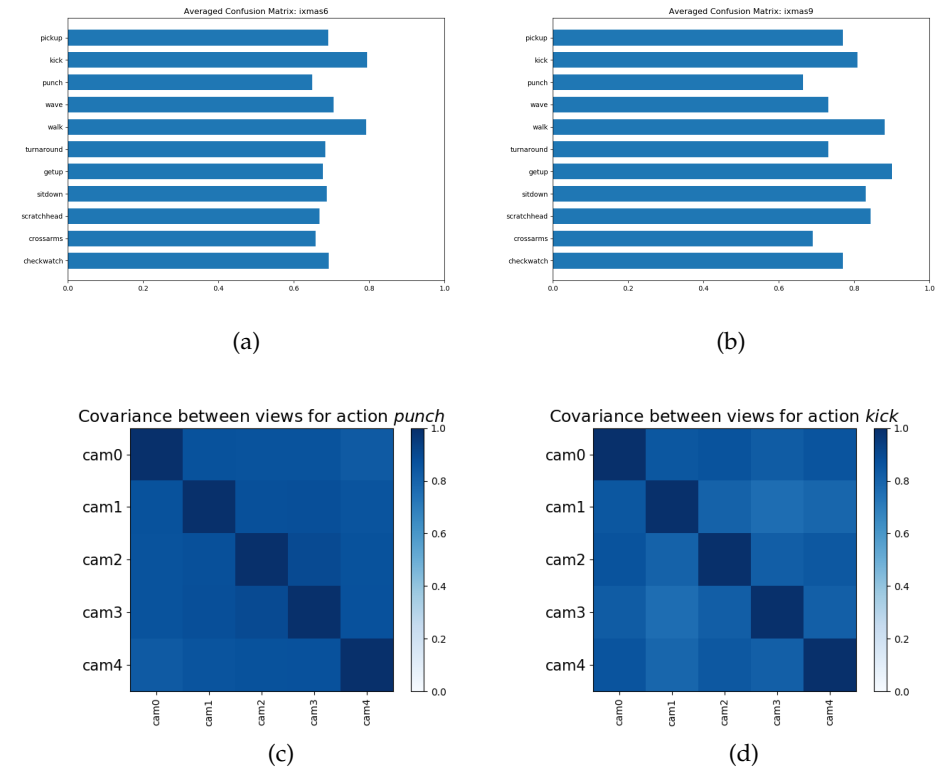


Figure 31: Above: average recognition accuracy of each class with the (a) SLP and (b) CNN classifiers respectively on the IXMAS dataset. Below: covariance matrix between the SLP's weight vectors of the same action trained on different training sets (viewpoints), for (c) the best and (d) worst performing on average (from the plot above).

by the simplest classifier, SLP and our proposed classifier, 3D CNN. Not all the actions perform similarly. The relationship between the trained SLP models, from different viewpoints are visually represented in Figure 31 (c) and (d). We propose here a similarity matrix computed between the representation of the deep model – i.e. the vector of weights – for a single action across all the views as covariance between weights. The weights of the SLP (which does not employ a non-linearity by design) are trained on the training set. Thus, the weights associated with each neuron can be considered a representative vector of coefficients pertaining to the training data for a label. Therefore a study of their relationship with each other and across views can be used to quantify their relationship.

In the Figure 31 we show the matrix for the two extreme cases, i.e. for the best performing action (i.e. *punch*), and for the worst one (i.e. *kick*). The first figure demonstrates high values and thus, stronger relationship between the actions themselves as expected, though the relationship with cam4 is still low. This is probably because the action is restricted to a small contained area in the scene. Figure 31(d) shows low values, as expected but displays the weaker relationship between cam1 and cam3 with others. This can be attributed to the large angular separation between them. From Figure 26, it can be noted that viewpoints cam1 and cam3 differ by a very high degree, not only in the horizontal angle, where they are approximately orthogonal, but also in height.

5.6 Multiple View training: Incorporating view-invariant information

This section reports on the ability of the classifiers to incorporate information from multiple[1] views during training, boosting view-invariance of the pipeline where such data is available. With multiple views available in the training data, classifiers have the resources available to incorporate the information form these viewpoints to build a relatively more view-invariant representation internally, discriminating between the view-invariant and view-dependent components of the learnt features. This is demonstrated with the help of the three datasets, NUCLA, NTURGBD and MoCA. Within multiple-view learning, for a dataset with n viewpoints available any umber of views upto $n-1$ can be used for training. We focus on specifically on training from $n-1$ views, which is also known as the *one-view-out protocol*.

Table 15 reports the comparative analysis on the NUCLA dataset, including performances of methods, in spirit, to ours as well as other interesting state of the art methods. The table also includes other methods that complement

1 Multiple view training here refers to use of multiple views in training, excluding the target view as opposed to multi-view action recognition in which target view is also used in training.

Table 15: Comparison of different methods and the architectures considered in this work on the NUCLA dataset. The analysis is based on the *multiple view learning*. Mod refers to the modalities of data that is involved in training the models, either as input or supervision. Sk: Skeleton. MoCap: motion capture.

Methods [cite]	Mod	$\{1,2\}\|3$	$\{1,3\}\|2$	$\{2,3\}\|1$	**Mean**
Hankelets [87, 180]	RGB	-	-	-	45.2
DVV [90]	RGB	58.5	55.2	39.3	51.0
CVP [208]	RGB	60.6	55.8	39.5	52
DA-Net [175]	RGB	86.5	82.7	83.1	84.2
MST-AOG [180]	RGB + Sk	-	-	-	73.3
nCTE[50]	MoCap	68.6	68.3	52.1	63.0
Zhang *et al* '16 [205]	MoCap	67.3	74.2	61.8	67.8
Zhang *et al* '18 [206]	MoCap	69.1	74.4	61.8	68.5
NKTM [132]	MoCap	75.8	73.3	59.1	69.4
R-NKTM [133]	MoCap	78.1	-	-	-
Li *et all* [88]	RGBD	62.5	-	-	-
Glimpse Clouds [10]	RGBD	90.1	89.5	83.4	87.6
I3D + SLP	RGB	69.09	69.91	64.51	67.84
I3D + Inc.	RGB	80.30	73.41	71.81	75.17
I3D + CV-3	RGB	**86.18**	**78.54**	**72.65**	**79.12**

the video data with other multi-modal information either during training or also during test. Wang et al. [180] uses skeletal information. Zhang et al. [205] and Zhang et al. [206] use depth-maps and motion capture information respectively to synthesize data for pre-training. Gupta et al. [50], Rahmani and Mian [132] and Rahmani, Mian and Shah [133] all use MoCap data during training, either as pose supervision or to learn 3D canonical views. Acquiring motion capture data for training can be a substantially expensive investment. Our method uses only video information throughout the entire process. Despite the complexity of the dataset, our proposed methodology effectively highlights the richness of the learnt features and their capability of characterizing actions with a strong tolerance to the variability of the specific viewpoint.

For the results on NTURGBD dataset, refer to Table 16. The table also contains results from a number of other state of the art methods. The NTU dataset provides depth maps, 3D joint information, RGB frames, and IR sequences and thus most works using the dataset employ more than one of these modalities during training and/or during testing. Some methods require data from the target views during training [88, 175]. DA-Net [175] requires extensive multi-view data to train. Others use pose information either as input to the model [9, 98] or for supervision during training [10]. For these reasons, we do not consider these methods comparable to ours. Note that many of these methods are

Table 16: Performance evaluation on NTU-RGBD Dataset [147] on the cross view standard protocol. D: Depth, Sk: Skeleton, OF: Optical Flow,

Method	Training Modality	Test Modality	Accuracy(%)
Part-LSTM [147]	Sk	Sk	70.3
Hands Attention [9]	RGB, 3D Pose	RGB, 3D Pose	90.6
Pose Evolution [98]	RGBD, 3D Pose	RGBD, 3D Pose	95.3
Glimpse Clouds [10]	RGB, 3D Pose	RGB	93.2
DA-Net [175]	Multi-view RGB	RGB	92.0
Li et al [88]	RGBD	RGBD	83.4
Pose Evolution [98]	RGB	RGB	84.2
Hands Attention [9]	RGB	RGB	80.5
I3D + SLP	RGB	RGB	65.6
I3D + Inc.	RGB	RGB	76.0
I3D + CV-3	RGB	RGB	84.9
I3D + CV-3-widened	RGB	RGB	86.1

Table 17: Performance evaluation (in %) on the MoCA dataset. Views - 0: Lateral, 1: Egocentric, 2: Frontal

Source |Target	0,1|2	0,2|1	1,2|0	Mean
I3D + SLP	67.46	46.03	68.10	60.53
I3D + Inc.	62.30	61.67	62.70	62.22
I3D + CV-3	**87.29**	**69.53**	**84.34**	**80.38**

training or extensively fine-tuning multiple deep networks streams, leading to unsustainable data and computational requirements.

Within the framework of methods using only RGB data through out the entire process, our method has reasonably good results. Considering the higher number of classes in NTU dataset, we also test an inflated version of the CV-3 classifier which has 4 times as may filters in each layer. The results are shown in Table 16 labelled *CV-3-widened*. We notice a small increase in the accuracy, though considering the increase in size of the classifier, this is a steep trade-off.

Finally, Table 17 reports the performances obtained on our MoCA dataset. This analysis has the main purpose of reasoning on the appropriateness of the proposed model with respect to egocentric vision. The trend of a weaker relationship between allocentric and egocentric views continues even when multiple allocentric views are used in training, though it is interesting to note that the recognition accuracy for the egocentric view increases by +14.69% and +7.70% in case of $0, 2|1$ with respect to $0|1$ and $2|1$ respectively, which

shows that the CV-3 classifier is able to learn a relatively more view invariant representation using multiple input views. In all above cases with our CV-3 classifier we can reach a good tolerance to view-point changes.

5.7 Resources Usage

In order to obtain a tangible understanding of the training efficiency of our method, we ran an experiment to calculate the time taken to train a classifier. To this end, we calculate the optical flow, extract the features and then train the classifier. To compare our results, we estimate the training time of DA-Net [175] on the same system. For this experiment we use a 16 GB NVIDIA Quadro P5000. We ran the training program of DA-Net for 24 hours, acquiring the data to extrapolate the overall training time. Since optical flow is necessary for both methods, we have removed the time taken to calculate it. Additionally, though our pipeline runs more efficiently if the optical flow is saved as video (.avi) files, but for a fare comparison, we used the same image files that are used by DA-Net. The DA-Net is run for 60 epochs by the authors. Our own pipelines converges sooner but here we show the time taken for the same number of epochs. The resulting times are shown in Table 18. DA-Net takes approximately 700% more time to train than our model. Our time to convergence is half of this value making the comparison even more startling. Note, that this value is an average of the training times of the 3 classifiers, since the difference between them is too small to be of significance in this table.

Classifier	Training time
DA-Net [175]	450 hours
Our Pipeline	56 hours

Table 18: Time taken by the two methods to train for 60 epochs on the standard split of the NTU datasets with a 16GB GPU NVIDIA Quadro P5000 after the optical flow calculation. The values are rounded to the closest hour.

5.8 Conclusion

In this chapter we presented an array of experimental results demonstrating the effectiveness of our methodological pipeline. The very first set of results proved the functionality of the method for an action recognition task, even with limited training data. The next set of results ensured the optimality of the pipeline's design choices, enabling an effective action recognition pipeline. The optimal extraction point is chosen based on the resulting accuracies as well as the size of the feature extracted from each point, since that has a direct

Table 19: Overall best performances of our pipelines with CV-3 classifier on the different datasets.

Training	IXMAS	MoCA	NUCLA	NTU
Single view	78.4	61.83	~	70.05
One-view-out	~	80.38	79.12	84.9

effect on the resources required for the training process. We chose to use a single stream between the two streams of the Inception3D model to reduce the resources involved in the feature extraction. We chose the Optical Flow stream over the raw RGB due to both, proof in the literature as well as experimental proof of better recognition accuracy based on features extracted from OF. We also demonstrated the improvement achieved by the modified Batch Normalization with respect to the regular Batch Normalization. Another comparative analysis that we address in this chapter is between a set of classifiers, both classical (linear SVM, gaussian SVM), and deep learning based (a single layered perceptron, the remaining layers of I3D, and the cross-view 3D convolutional classifier). Most experiments in the chapter employed 3 or more classifiers and the results showed that the CV-3 classifier was the most effective for most of the scenarios.

With the components of the pipeline finalized, we moved to a more detailed study of the datasets. We proposed the Single View Learning task, where all actions of a dataset were learnt by a classifier using samples from a single view. This task has the potential of being able to establish and analyze the relationship between the viewpoints. Learning from a single viewpoint requires the representation itself to be as robust to view-point change, and promotes design of systems that require raining data that can be easily acquired. As the CV-3 classifiers continues to outperform the other classifiers, this set of experiments are focused on the analysis of the features themselves. It is clearly demonstrated that the features contain a viable amount of information about the action that is independent of the viewpoint. Viewpoints that are closer in angular separation to the training view tend to be easier to recognize. Large angular separations lead to weaker relationships between the viewpoints in the feature space. Additionally, egocentric view also has a weak relationship with allocentric (second and third person) viewpoints. But the major cause of miss-classification is occlusion. These occlusions are present both in cases of simple human-human action viewpoint – where the major dynamics of the action may be covered by the actor's body – or due to difficult viewpoints where the camera views the actor from a point that leads to loss of information in most cases – like an over-the-head camera looking vertically downwards at the actor. Other than loss of information, these difficult viewpoints may also fail due to a weak relationship between the viewpoints in the feature space caused by a scarcity of samples from similar viewpoints in the original pre-training of the feature extracting component.

The next set of experiments analyze the ability of the classifiers to integrate information from multiple views during training. Training samples include each action from multiple views and the test is still on a view that is not included in the training data. The results from NTU RGBD and MoCA demonstrated that using multiple views improved the recognition ability of the datasets, as shown here in Table 19. Combination of allocentric views were able to provide sufficient information to the classifiers in order to bring the recognition accuracy of the egocentric to approximately the same values as the other training-test combinations.

The methodological pipeline has proven to be not only effective in its capacity to recognize actions from unseen viewpoints but to do so using a fraction of resources that other methods require. Moreover, this method has also been proven to be a powerful tool to study the relationship between viewpoints, enabling us to build a better understanding of features extracted from deep networks, and of the dynamics of actions when viewed from different viewpoints.

6

General Discussion and Conclusion

The recognition of human action is a crucial component of a system designed to understand a human-centric scenario, be it for the purpose of interaction or observation. As technology is getting out of our laboratories and into the phase of mass deployment, the systems need to be capable of adapting to new scenarios and variations in the visual information. While we can adopt methods to tackle some of those variations from well researched image-based scenarios, action recognition poses its own unique challenges. One of these challenges is posed by changes in viewpoint which leads to substantially different observed visual data.

To deal with view invariance in action recognition, we propose a novel pipeline applying it to a cross view scenario. The first step of this pipeline was to determine a suitable representation of video input with view-invariance properties. To this end we explored a classical representation based on the Shearlet transform. Although promising in its view-invariance, the representation was limited in its possibility towards the criterion of limited resources. Thereafter we explored some deeply learned representations, pin-pointing a specific representation for the task. This process involved examining existing models for the most appropriate features, done by relying on the results of existing studies in conjunction with a number of comparative analyses, the process and results of these are presented and discussed in Chapter 4 and Chapter 5. Our pipeline for cross-view action recognition leverages the Inception3D (I3D) [16] network pre-trained on ImageNet and Kinetics datasets for feature extraction considerably reducing the resource requirement of the method, in terms of required training data and computation requirements, while simultaneously obtaining an informative representation. We paired an array of classifiers with a variety of features, to study the optimal representation. We also propose the CV-3 for the most effective cross view recognition. Additionally, we propose the use of a modified batch normalization technique which, by evidence, promotes the capability of the different classifiers in adapting to new viewpoints.

We provide evidence that cross-view action recognition can be performed considerably well using only video input, obtaining results comparable to meth-

ods that use data from other sensors like RGBD sensors or Motion Capture systems. This has been shown in studies presented by Baradel, Wolf and Mille [9] and Liu and Yuan [98] very recently, using diverse methods, corroborating our results. Moreover, we have shown that our methodology taps into this capability even when the model is trained on *a single view*, with only a slightly reduced accuracy. And pre-trained networks can be leveraged to do so with limited resources.

Another method of gauging the effectiveness of the pipeline is the improvement that is makes over the pre-trained model, part of which is used as the feature classifier. Although 3 of the 4 datasets we consider in our work would not be large enough to finetune I3D end-to-end, the final one would easily take over 2 months to do so. Our pipeline leads to a trained model in a little over 2 days on the same system. Moreover to examine a scenario where I3D is partially fine-tuned on the smaller dataset, we can consider the result of the Inception classifier, with the regular batch normalization to be the effective resulting accuracy on the small scale datasets. Therefore, any improvement made on top of those values can be considered due to the overall methodology in dealing with unseen viewpoints. This gap is very high in case of single view learning scenario, referring to the results of the IXMAS dataset using the optical flow stream of the network and *mixed_5b* extraction point, accuracy improved by 13%. In multiple-view learning, this gap is less, for example in NUCLA, with the same experimental parameters, the accuracy increased by 1.05%. All in all, the difference between training resources and recognition power is appreciable.

We proposed a single view learning task, wherein a model learns samples of action videos captured from only one viewpoint. This scenario pushes the limits of cross-view action recognition, promoting minimal resource usage. We found that our method not only functions reasonably well within this scenario, it also outperformed all existing methods that can be trained on a single viewpoint for which results are obtainable. Combining this expertise with the multiple-view training scenario boosts the accuracy of the pipeline, especially when testing on more challenging viewpoints like the egocentric.

Our analysis also allowed us to build an understanding of the problem, the representation or feature space, as well as the relationship between viewpoints.

- The pre-trained I3D layers used out-of-the-box with no fine tuning of the new dataset, performed action recognition acceptably well, even with simple single layer perceptron classifier.

- In terms of cross view action recognition, the simple classifiers still perform reasonably well, implying that the representation they are using as input inherently contains some view invariant information. Our proposed classifier, when coupled with this representation, leverages the full potential of the pre-training to obtain higher accuracies than all the

methods found in the literature for the single view learning protocol and competitive results for cross-view recognition from training using multiple views.

- Modified batch normalization, using the local batch of the target data, promotes the view-invariant capability of the classifier.

- The system performed substantially better on viewpoints that are common within the scenario of human-human interaction, demonstrating that this dataset bias from pre-training dataset continues to affect the performance of classifiers, even with the improved performance of the CV-3 classifier.

Since the pre-trained model is used as-is, one possibility for a future work can be to minimize the resource requirement even farther by simplifying the feature extractor. On a different direction, a natural extension would be to explore Generative Adversarial Networks (GANs) to generate multi-view video data with a lower view-point bias with respect to the currently existing datasets, and to extend the validity of the methods applied to the cross-view action recognition scenario towards a view-invariance setting, as much as action recognition allows.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. Vol. 16. 2016, pp. 265–283.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Apostol Natsev, George Toderici, Balakrishnan Varadarajan and Sudheendra Vijayanarasimhan. "YouTube-8M: A Large-Scale Video Classification Benchmark". In: *ArXiv* abs/1609.08675 (2016).

[3] Catherine Achard, Xingtai Qu, Arash Mokhber and Maurice Milgram. "A novel approach for recognition of human actions with semi-global features". In: *Machine Vision and Applications* 19.1 (2008), pp. 27–34.

[4] Jake K Aggarwal and Lu Xia. "Human activity recognition from 3d data: A review". In: *Pattern Recognition Letters* 48 (2014), pp. 70–80.

[5] Maryam Asadi-Aghbolaghi, Albert Clapés, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei and Sergio Escalera. "A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences". In: (2017), pp. 476–483. DOI: 10.1109/FG.2017.150. URL: https://doi.org/10.1109/FG.2017.150.

[6] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia and Atilla Baskurt. "Sequential Deep Learning for Human Action Recognition". In: *Human Behavior Unterstanding - Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings*. 2011, pp. 29–39. DOI: 10.1007/978-3-642-25446-8_4. URL: https://doi.org/10.1007/978-3-642-25446-8_4.

[7] Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[8] Gioia Ballin, Matteo Munaro and Emanuele Menegatti. "Human action recognition from rgb-d frames based on real-time 3d optical flow estimation". In: *Biologically Inspired Cognitive Architectures 2012*. Springer, 2013, pp. 65–74.

[9] Fabien Baradel, Christian Wolf and Julien Mille. "Human action recognition: Pose-based attention draws focus to hands". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 604–613.

[10] Fabien Baradel, Christian Wolf, Julien Mille and Graham W Taylor. "Glimpse clouds: Human activity recognition from unstructured feature points". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 469–478.

[11] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[12] Robert Bodor, Bennett Jackson and Nikolaos Papanikolopoulos. "Vision-based human tracking and activity recognition". In: *Proc. of the 11th Mediterranean Conf. on Control and Automation*. Vol. 1. 2003.

[13] Paulo Vinicius Koerich Borges, Nicola Conci and Andrea Cavallaro. "Video-based human behavior understanding: A survey". In: *IEEE transactions on circuits and systems for video technology* 23.11 (2013), pp. 1993–2008.

[14] Allah Bux, Plamen Angelov and Zulfiqar Habib. "Vision based human activity recognition: a review". In: *Advances in Computational Intelligence Systems*. Springer, 2017, pp. 341–371.

[15] Linqin Cai, Xiaolin Liu, Fuli Chen and Min Xiang. "Robust human action recognition based on depth motion maps and improved convolutional neural network". In: *Journal of Electronic Imaging* 27.5 (2018), p. 051218.

[16] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *CVPR, 2017*. IEEE. 2017, pp. 4724–4733.

[17] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi and Andrew Zisserman. "The devil is in the details: an evaluation of recent feature encoding methods." In: *BMVC*. Vol. 2. 4. 2011, p. 8.

[18] Liang-Chieh Chen, George Papandreou, Florian Schroff and Hartwig Adam. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[19] Zhe Chen, Xueli Hao and Zhaoyun Sun. "Image denoising in shearlet domain by adaptive thresholding". In: *Journal of Information & Computational Science* 10.12 (2013), pp. 3741–3749.

[20] Srikanth Cherla, Kaustubh Kulkarni, Amit Kale and Viswanathan Ramasubramanian. "Towards fast, view-invariant human action recognition". In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2008, pp. 1–8.

[21] Chun-Te Chu, Jenq-Neng Hwang, Shen-Zheng Wang and Yi-Yuan Chen. "Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients". In: *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*. IEEE. 2011, pp. 1–6.

[22] Lovish Chum, Anbumani Subramanian, Vineeth N Balasubramanian and CV Jawahar. "Beyond Supervised Learning: A Computer Vision Perspective". In: *Journal of the Indian Institute of Science* (2019), pp. 1–23.

[23] Gabriela Csurka. "Domain adaptation for visual applications: A comprehensive survey". In: *arXiv preprint arXiv:1702.05374* (2017).

[24] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski and Cédric Bray. "Visual categorization with bags of keypoints". In: *ECCV-W*. Vol. 1. 1-22. 2004.

[25] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang and Yong Yu. "Translated Learning: Transfer Learning across Different Feature Spaces". In: *NIPS, 2008*. 2008, pp. 353–360.

[26] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: 2005.

[27] Navneet Dalal, Bill Triggs and Cordelia Schmid. "Human detection using oriented histograms of flow and appearance". In: *European conference on computer vision*. Springer. 2006, pp. 428–441.

[28] Yonghao Dang, Fuxing Yang and Jianqin Yin. "DWnet: Deep-Wide Network for 3D Action Recognition". In: *arXiv preprint arXiv:1908.11036* (2019).

[29] Aras Dargazany and Mircea Nicolescu. "Human body parts tracking using torso tracking: applications to activity recognition". In: *2012 Ninth International Conference on Information Technology-New Generations*. IEEE. 2012, pp. 646–651.

[30] Debapratim Das Dawn and Soharab Hossain Shaikh. "A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector". In: *The Visual Computer* 32.3 (2016), pp. 289–306.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *CVPR 2009*. 2009, pp. 248–255.

[32] Piotr Dollár, Vincent Rabaud, Garrison Cottrell and Serge Belongie. "Behavior recognition via sparse spatio-temporal features". In: VS-PETS Beijing, China. 2005.

[33] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers and Thomas Brox. "Flownet: Learning optical flow with convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766.

[34] Ionut Cosmin Duta, Bogdan Ionescu, Kiyoharu Aizawa and Nicu Sebe. "Spatio-Temporal Vector of Locally Max Pooled Features for Action Recognition in Videos". In: (2017), pp. 3205–3214. DOI: 10.1109/CVPR.2017.341. URL: https://doi.org/10.1109/CVPR.2017.341.

[35] M. A. Duval-Poo, F. Odone and E. De Vito. "Edges and Corners With Shearlets". In: *IEEE T. Image Proc.* 24.11 (2015), pp. 3768–3780.

[36] M. A. Duval-Poo, N. Noceti, F. Odone and E. De Vito. "Scale Invariant and Noise Robust Interest Points with Shearlets". In: *IEEE T. Image Proc.* (2017). DOI: 10.1109/TIP.2017.2687122.

[37] Glenn R Easley, Demetrio Labate and Flavia Colonna. "Shearlet-based total variation diffusion for denoising". In: *TIP* 18.2 (2009), pp. 260–268.

[38] Claudia Elsner, Terje Falck-Ytter and Gustaf Gredebäck. "Humans anticipate the goal of other people?s point-light actions". In: *Front. in Psychology* 3 (2012).

[39] Bernard Ghanem Fabian Caba Heilbron Victor Escorcia and Juan Carlos Niebles. "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.

[40] Christoph Feichtenhofer, Axel Pinz and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1933–1941.

[41] Adrien Gaidon, Zaid Harchaoui and Cordelia Schmid. "Activity representation with motion hierarchies". In: *International journal of computer vision* 107.3 (2014), pp. 219–238.

[42] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand and Victor Lempitsky. "Domain-adversarial training of neural networks". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.

[43] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic and Bryan Russell. "ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 3165–3174. DOI: 10.1109/CVPR.2017.337. URL: https://doi.org/10.1109/CVPR.2017.337.

[44] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CVPR 2014*. 2014, pp. 580–587.

[45] Nikolaos Gkalelis, Hansung Kim, Adrian Hilton, Nikos Nikolaidis and Ioannis Pitas. "The i3dpost multi-view and 3d human action/interaction database". In: *2009 Conference for Visual Media Production*. IEEE. 2009, pp. 159–168.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[47] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani and Ronen Basri. "Actions as space-time shapes". In: *IEEE transactions on pattern analysis and machine intelligence* 29.12 (2007), pp. 2247–2253.

[48] Guodong Guo and Alice Lai. "A survey on still image based human action recognition". In: *Pattern Recognition 2014* 47.10 (2014), pp. 3343–3361.

[49] Kanghui Guo, Demetrio Labate and Wang-Q Lim. "Edge analysis and identification using the continuous shearlet transform". In: *Applied and Computational Harmonic Analysis* 27.1 (2009), pp. 24–46.

[50] Ankur Gupta, Julieta Martinez, James J. Little and Robert J. Woodham. "3D Pose from Motion for Cross-View Action Recognition via Non-linear Circulant Temporal Encoding". In: *CVPR*. IEEE Computer Society, 2014.

[51] Fei Han, Brian Reily, William Hoff and Hao Zhang. "Space-time representation of people based on 3D skeletal data: A review". In: *Computer Vision and Image Understanding* 158 (2017), pp. 85–105.

[52] Kensho Hara, Hirokatsu Kataoka and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *CoRR* abs/1711.09577 (2017). arXiv: 1711.09577. URL: http://arxiv.org/abs/1711.09577.

[53] Christopher G Harris, Mike Stephens et al. "A combined corner and edge detector." In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[55] Samitha Herath, Mehrtash Harandi and Fatih Porikli. "Going deeper into action recognition: A survey". In: *Image and vision computing 2017* 60 (2017), pp. 4–21.

[56] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[57] Berthold KP Horn and Brian G Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[58] Chun-Hao Huang, Yi-Ren Yeh and Yu-Chiang Frank Wang. "Recognizing actions across cameras by exploring the correlated subspace". In: *ECCV 2012*. Springer. 2012.

[59] Gao Huang, Zhuang Liu, Laurens Van Der Maaten and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[60] Kaiqi Huang, Yeying Zhang and Tieniu Tan. "A discriminative model of motion and cross ratio for view-invariant action recognition". In: *IEEE TIP 2012* 21.4 ().

[61] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[62] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez and Daniel Cremers. "A primal-dual framework for real-time dense RGB-D scene flow". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 98–104.

[63] Hervé Jégou, Matthijs Douze, Cordelia Schmid and Patrick Pérez. "Aggregating local descriptors into a compact image representation". In: *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*. IEEE Computer Society. 2010, pp. 3304–3311.

[64] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez and Cordelia Schmid. "Aggregating local image descriptors into compact codes". In: *IEEE transactions on pattern analysis and machine intelligence* 34.9 (2011), pp. 1704–1716.

[65] Shuiwang Ji, Wei Xu, Ming Yang and Kai Yu. "3D Convolutional Neural Networks for Human Action Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.1 (2013), pp. 221–231. DOI: 10.1109/TPAMI.2012.59. URL: https://doi.org/10.1109/TPAMI.2012.59.

[66] Yanli Ji, Feixiang Xu, Yang Yang, Fumin Shen, Heng Tao Shen and Wei-Shi Zheng. "A large-scale rgb-d database for arbitrary-view human action recognition". In: *2018 ACM Multimedia Conference on Multimedia Conference*. ACM. 2018, pp. 1510–1518.

[67] Imran Junejo, Emilie Dexter, Ivan Laptev and Patrick Perez. "View-independent action recognition from temporal self-similarities". In: *IEEE PAMI 2011* (2011).

[68] A KLASER. "A spatiotemporal descriptor based on 3D-gradients". In: *19th British Machine Vision Conference, September 2008*. 2008, pp. 995–1004.

[69] Sahak Kaghyan and Hakob Sarukhanyan. "Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer". In: *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria* 1 (2012), pp. 146–156.

[70] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar and Fei-Fei Li. "Large-Scale Video Classification with Convolutional Neural Networks". In: (2014), pp. 1725–1732. DOI: 10.1109/CVPR.2014.223. URL: https://doi.org/10.1109/CVPR.2014.223.

[71] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green et al. "The kinetics human action video dataset". In: *arXiv preprint* (2017).

[72] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel and Farid Boussaid. "A new representation of skeleton sequences for 3d action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3288–3297.

[73] Yu Kong, Zhengming Ding, Jun Li and Yun Fu. "Deeply Learned View-Invariant Features for Cross-View Action Recognition". In: *IEEE Trans. Image Processing 2017* 26.6 (2017), pp. 3028–3037. DOI: 10.1109/TIP.2017.2696786. URL: https://doi.org/10.1109/TIP.2017.2696786.

[74] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *NIPS 2012*. 2012, pp. 1097–1105.

[75] Hilde Kuehne, Ali Arslan and Thomas Serre. "The language of actions: Recovering the syntax and semantics of goal-directed human activities". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 780–787.

[76] Sonal Kumari and Suman K Mitra. "Human action recognition using DFT". In: *2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*. IEEE. 2011, pp. 239–242.

[77] G. Kutyniok, W.Q. Lim and R. Reisenhofer. "ShearLab 3D: Faithful Digital Shearlet Transforms Based on Compactly Supported Shearlets". In: *ACM Trans. Math. Softw.* (2016), 5:1–5:42.

[78] Gitta Kutyniok and Demetrio Labate. *Shearlets*. Appl. Numer. Harmon. Anal. Birkhäuser/Springer, New York, 2012.

[79] Gitta Kutyniok and Philipp Petersen. "Classification of edges using compactly supported shearlets". In: *Applied and Computational Harmonic Analysis* (2015).

[80] Demetrio Labate, Wang-Q Lim, Gitta Kutyniok and Guido Weiss. "Sparse multidimensional representation using shearlets". In: *Optics & Photonics*. 2005.

[81] I. Laptev. "On space-time interest points". In: *Int. J. Computer Vision* 64.2 (2005), pp. 107–123.

[82] Ivan Laptev. "On Space-Time Interest Points". In: *International Journal of Computer Vision* 64.2-3 (2005), pp. 107–123. DOI: 10.1007/s11263-005-1838-7. URL: https://doi.org/10.1007/s11263-005-1838-7.

[83] Ivan Laptev, Marcin Marszalek, Cordelia Schmid and Benjamin Rozenfeld. "Learning realistic human actions from movies". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[84] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[85] Iat-Fai Leong, Jing-Jing Fang and Ming-June Tsai. "Automatic body feature extraction from a marker-less scanned human body". In: *Computer-Aided Design* 39.7 (2007), pp. 568–582.

[86] Maylor K. Leung and Yee-Hong Yang. "First sight: A human body outline labeling system". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.4 (1995), pp. 359–377.

[87] Binlong Li, Octavia I Camps and Mario Sznaier. "Cross-view activity recognition using hankelets". In: *IEEE CVPR 2012*. IEEE. 2012.

[88] Junnan Li, Yongkang Wong, Qi Zhao and Mohan Kankanhalli. "Unsupervised learning of view-invariant action representations". In: *Advances in Neural Information Processing Systems*. 2018, pp. 1254–1264.

[89] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui and Jiebo Luo. "Action Recognition by Learning Deep Multi-Granular Spatio-Temporal Video Representation". In: (2016), pp. 159–166. DOI: 10.1145/2911996.2912001. URL: http://doi.acm.org/10.1145/2911996.2912001.

[90] Ruonan Li and Todd Zickler. "Discriminative virtual views for cross-view action recognition". In: *IEEE CVPR 2012*. IEEE. 2012.

[91] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu and Xiaodi Hou. "Revisiting batch normalization for practical domain adaptation". In: *arXiv preprint arXiv:1603.04779* (2016).

[92] Bin Liang and Lihong Zheng. "A survey on human action recognition using depth sensors". In: *2015 International conference on digital image computing: techniques and applications (DICTA)*. IEEE. 2015, pp. 1–8.

[93] Duohan Liang, Guoliang Fan, Guangfeng Lin, Wanjun Chen, Xiaorong Pan and Hong Zhu. "Three-Stream Convolutional Neural Network With Multi-Task and Ensemble Learning for 3D Action Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[94] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *ECCV 2014*. Springer. 2014, pp. 740–755.

[95] Zachary C Lipton, John Berkowitz and Charles Elkan. "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019* (2015).

[96] Jun Liu, Amir Shahroudy, Dong Xu and Gang Wang. "Spatio-temporal lstm with trust gates for 3d human action recognition". In: *European Conference on Computer Vision*. Springer. 2016, pp. 816–833.

[97] Li Liu, Ling Shao, Xuelong Li and Ke Lu. "Learning spatio-temporal representations for action recognition: A genetic programming approach". In: *IEEE transactions on cybernetics* 46.1 (2015), pp. 158–170.

[98] Mengyuan Liu and Junsong Yuan. "Recognizing human actions as the evolution of pose estimation maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1159–1168.

[99] Ming-Yu Liu and Oncel Tuzel. "Coupled generative adversarial networks". In: *Advances in neural information processing systems*. 2016, pp. 469–477.

[100] Pengpeng Liu, Michael Lyu, Irwin King and Jia Xu. "SelFlow: Self-Supervised Learning of Optical Flow". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4571–4580.

[101] Jonathan Long, Evan Shelhamer and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[102] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[103] David G Lowe et al. "Object recognition from local scale-invariant features." In: *iccv*. Vol. 99. 2. 1999, pp. 1150–1157.

[104] Wei-Lwun Lu and James J Little. "Simultaneous tracking and action recognition using the pca-hog descriptor". In: *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE. 2006, pp. 6–6.

[105] Xin Lu, Qiong Liu and Shunichiro Oe. "Recognizing non-rigid human actions using joints tracking in space-time". In: *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.* Vol. 1. IEEE. 2004, pp. 620–624.

[106] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[107] Damiano Malafronte, Francesca Odone and Ernesto De Vito. "Detecting Spatio-temporally Interest Points using the Shearlet Transform". In: *IBPRIA*. 2017.

[108] Damiano Malafronte, Francesca Odone and Ernesto De Vito. "Local Spatio-Temporal Representation using the 3D Shearlet Transform". In: *SAMPTA*. 2017.

[109] CD Manning, R PRABHAKAR and S HINRICH. "Introduction to information retrieval, volume 1 Cambridge University Press". In: *Cambridge, UK* (2008).

[110] Hiroaki Mikami, Hisahiro Suganuma, Yoshiki Tanaka, Yuichi Kageyama et al. "Massively Distributed SGD: ImageNet/ResNet-50 Training in a Flash". In: *arXiv preprint arXiv:1811.05233* (2018).

[111] Thomas B Moeslund, Adrian Hilton and Volker Krüger. "A survey of advances in vision-based human motion capture and analysis". In: *Computer vision and image understanding* 104.2-3 (2006), pp. 90–126.

[112] Pradeep Natarajan and Ramakant Nevatia. "View and scale invariant action recognition using multiview shape-flow models". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[113] Cisco Visual Networking Index. "Forecast and methodology, 2016-2021, white paper". In: *San Jose, CA, USA* 1 (2016), p. 1.

[114] Elena Nicora, Gaurvi Goyal, Nicoletta Noceti and Francesca Odone. "The Effects of Data Sources: A Baseline Evaluation of the MoCA Dataset". In: *International Conference on Image Analysis and Processing*. Springer. 2019, pp. 544–555.

[115] Akitsugu Noguchi and Keiji Yanai. "A surf-based spatio-temporal feature for feature-fusion-based action recognition". In: *European Conference on Computer Vision*. Springer. 2010, pp. 153–167.

[116] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal and Ruzena Bajcsy. "Berkeley mhad: A comprehensive multimodal human action database". In: *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE. 2013, pp. 53–60.

[117] Maxime Oquab, Leon Bottou, Ivan Laptev and Josef Sivic. "Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks". In: *CVPR 2014*. 2014.

[118] O Oshin, A Gilbert, J Illingworth and R Bowden. "Spatio-Temporal Feature Recognition using Randomised Ferns". In: *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis (MVLMA'08)*. 2008.

[119] Edouard Oyallon, Eugene Belilovsky and Sergey Zagoruyko. "Scaling the Scattering Transform: Deep Hybrid Networks". In: (2017), pp. 5619–5628. DOI: 10.1109/ICCV.2017.599. URL: https://doi.org/10.1109/ICCV.2017.599.

[120] Mustafa Ozuysal, Pascal Fua and Vincent Lepetit. "Fast keypoint recognition in ten lines of code". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2007, pp. 1–8.

[121] Eunbyung Park, Xufeng Han, Tamara L. Berg and Alexander C. Berg. "Combining multiple sources of knowledge in deep CNNs for action recognition". In: *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7-10, 2016*. 2016, pp. 1–8. DOI: 10.1109/WACV.2016.7477589. URL: https://doi.org/10.1109/WACV.2016.7477589.

[122] Vishal M Patel, Raghuraman Gopalan, Ruonan Li and Rama Chellappa. "Visual domain adaptation: A survey of recent advances". In: *IEEE signal processing magazine* 32.3 (2015), pp. 53–69.

[123] Xiaojiang Peng, Changqing Zou, Yu Qiao and Qiang Peng. "Action recognition with stacked fisher vectors". In: *European Conference on Computer Vision*. Springer. 2014, pp. 581–595.

[124] Xiaojiang Peng, Limin Wang, Xingxing Wang and Yu Qiao. "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice". In: *Computer Vision and Image Understanding* 150 (2016), pp. 109–125.

[125] Javier Sánchez Pérez, Enric Meinhardt-Llopis and Gabriele Facciolo. "TV-L1 optical flow estimation". In: *Image Processing On Line* 2013 (2013), pp. 137–150.

[126] Florent Perronnin, Jorge Sánchez and Thomas Mensink. "Improving the fisher kernel for large-scale image classification". In: *European conference on computer vision*. Springer. 2010, pp. 143–156.

[127] Ronald Poppe. "A survey on vision-based human action recognition". In: *Image Vision Comput.* 28.6 (2010), pp. 976–990. DOI: 10.1016/j.imavis.2009.11.014. URL: https://doi.org/10.1016/j.imavis.2009.11.014.

[128] Huimin Qian, Yaobin Mao, Wenbo Xiang and Zhiquan Wang. "Recognition of human activities using SVM multi-class classifier". In: *Pattern Recognition Letters* 31.2 (2010), pp. 100–111.

[129] Zhaofan Qiu, Ting Yao and Tao Mei. "Learning spatio-temporal representation with pseudo-3d residual networks". In: *proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5533–5541.

[130] Lawrence R Rabiner and Biing-Hwang Juang. "An introduction to hidden Markov models". In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[131] Lawrence Rabiner and Biing-Hwang Juang. "Fundamentals of speech processing". In: *Prantice Hall* (1993).

[132] Hossein Rahmani and Ajmal S. Mian. "Learning a non-linear knowledge transfer model for cross-view action recognition". In: *IEEE CVPR 2015*. 2015, pp. 2458–2466. DOI: 10.1109/CVPR.2015.7298860. URL: https://doi.org/10.1109/CVPR.2015.7298860.

[133] Hossein Rahmani, Ajmal Mian and Mubarak Shah. "Learning a deep model for human action recognition from novel viewpoints". In: *IEEE PAMI 2018* 40.3 (2018), pp. 667–681.

[134] Hossein Rahmani, Arif Mahmood, Du Q. Huynh and Ajmal S. Mian. "Histogram of Oriented Principal Components for Cross-View Action Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38.12 (2016), pp. 2430–2443. DOI: 10.1109/TPAMI.2016.2533389. URL: https://doi.org/10.1109/TPAMI.2016.2533389.

[135] Cen Rao, Alper Yilmaz and Mubarak Shah. "View-invariant representation and recognition of actions". In: *International Journal of Computer Vision* 50.2 (2002), pp. 203–226.

[136] Kishore K Reddy and Mubarak Shah. "Recognizing 50 human action categories of web videos". In: *Machine Vision and Applications* 24.5 (2013), pp. 971–981.

[137] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[138] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[139] Alexander Richard, Hilde Kuehne and Juergen Gall. "Weakly Supervised Action Learning with RNN Based Fine-to-Coarse Modeling". In: (2017), pp. 1273–1282. DOI: 10.1109/CVPR.2017.140. URL: https://doi.org/10.1109/CVPR.2017.140.

[140] Neil Robertson and Ian Reid. "A general method for human activity recognition in video". In: *Computer Vision and Image Understanding* 104.2-3 (2006), pp. 232–248.

[141] Grégory Rogez, José Jesús Guerrero and Carlos Orrite. "View-invariant human feature extraction for video-surveillance applications". In: *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE. 2007, pp. 324–329.

[142] Myung-Cheol Roh, Ho-Keun Shin and Seong-Whan Lee. "View-independent human action recognition with volume motion template on single stereo camera". In: *Pattern Recognition Letters* 31.7 (2010), pp. 639–647.

[143] Allah Bux Sargano, Plamen Angelov and Zulfiqar Habib. "A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition". In: *applied sciences* 7.1 (2017), p. 110.

[144] C. Schuldt, I. Laptev and B. Caputo. "Recognizing Human Actions: A Local SVM Approach". In: *ICPR*. Vol. 3. 2004.

[145] Christian Schuldt, Ivan Laptev and Barbara Caputo. "Recognizing human actions: a local SVM approach". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. IEEE. 2004, pp. 32–36.

[146] Paul Scovanner, Saad Ali and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition". In: *Proceedings of the 15th ACM international conference on Multimedia*. ACM. 2007, pp. 357–360.

[147] Amir Shahroudy, Jun Liu, Tian-Tsong Ng and Gang Wang. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[148] Eli Shechtman and Michal Irani. "Space-time behavior based correlation". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 405–412.

[149] Lei Shi, Yifan Zhang, Jian Cheng and Hanqing Lu. "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12026–12035.

[150] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang and Russell Webb. "Learning from simulated and unsupervised images through adversarial training". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2107–2116.

[151] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[152] Bharat Singh, Tim K. Marks, Michael J. Jones, Oncel Tuzel and Ming Shao. "A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection". In: (2016), pp. 1961–1970. DOI: 10.1109/CVPR.2016.216. URL: https://doi.org/10.1109/CVPR.2016.216.

[153] Sanchit Singh, Sergio A Velastin and Hossein Ragheb. "Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods". In: *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE. 2010, pp. 48–55.

[154] J Sivic and A Zisserman. "Video Google: a text retrieval approach to object matching in videos". In: *Proceedings Ninth IEEE International Conference on Computer Vision*.

[155] Andrews Sobral and Antoine Vacavant. "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos". In: *Computer Vision and Image Understanding* 122 (2014), pp. 4–21.

[156] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *CoRR* (2012).

[157] César Roberto de Souza12, Adrien Gaidon, Yohann Cabon and Antonio Manuel López. "Procedural generation of videos to train deep action recognition networks". In: (2017).

[158] Emma Strubell, Ananya Ganesh and Andrew McCallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *arXiv preprint arXiv:1906.02243* (2019).

[159] Lin Sun, Kui Jia, Dit-Yan Yeung and Bertram E Shi. "Human action recognition using factorized spatio-temporal convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4597–4605.

[160] Tanveer Syeda-Mahmood, A Vasilescu and Saratendu Sethi. "Recognizing action events from multiple viewpoints". In: *Detection and Recognition of Events in Video, 2001*. IEEE. 2001.

[161] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. "Going deeper with convolutions". In: *CVPR 2015*. 2015.

[162] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision". In: (2016), pp. 2818–2826. DOI: 10.1109/CVPR.2016.308. URL: https://doi.org/10.1109/CVPR.2016.308.

[163] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke and Alexander A. Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* 2017, pp. 4278–4284. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806.

[164] Hoang Le Uyen Thuc, Shian-Ru Ke, Jenq-Neng Hwang, Pham Van Tuan and Truong Ngoc Chau. "Quasi-periodic action recognition from monocular videos via 3D human models and cyclic HMMs". In: *The 2012 International Conference on Advanced Technologies for Communications*. IEEE. 2012, pp. 110–113.

[165] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne and Henrike Moll. "Understanding and sharing intentions: The origins of cultural cognition". In: *Behavioral and brain sciences* 28.05 (2005), pp. 675–691.

[166] Alexander Toshev and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1653–1660.

[167] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani and Manohar Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *IEEE ICCV 2015*. 2015, pp. 4489–4497. DOI: 10.1109/ICCV.2015.510. URL: https://doi.org/10.1109/ICCV.2015.510.

[168] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*. Vol. 201. Prentice Hall Englewood Cliffs, 1998.

[169] Zhigang Tu, Wei Xie, Dejun Zhang, Ronald Poppe, Remco C Veltkamp, Baoxin Li and Junsong Yuan. "A survey of variational and CNN-based optical flow techniques". In: *Signal Processing: Image Communication* 72 (2019), pp. 9–24.

[170] Gül Varol, Ivan Laptev and Cordelia Schmid. "Long-term Temporal Convolutions for Action Recognition". In: *CoRR* abs/1604.04494 (2016). arXiv: 1604.04494. URL: http://arxiv.org/abs/1604.04494.

[171] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins and Takeo Kanade. "Three-dimensional scene flow". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. IEEE. 1999, pp. 722–729.

[172] Ashok Veeraraghavan, A Roy Chowdhury and Rama Chellappa. "Role of shape and kinematics in human movement analysis". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. IEEE. 2004, pp. I–I.

[173] Ashok Veeraraghavan, Amit K Roy-Chowdhury and Rama Chellappa. "Matching shape sequences in video with applications in human movement analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.12 (2005), pp. 1896–1909.

[174] Michalis Vrigkas, Christophoros Nikou and Ioannis A Kakadiaris. "A review of human activity recognition methods". In: *Frontiers in Robotics and AI* 2 (2015), p. 28.

[175] Dongang Wang, Wanli Ouyang, Wen Li and Dong Xu. "Dividing and aggregating network for multi-view action recognition". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 451–467.

[176] Haoran Wang, Chunfeng Yuan, Weiming Hu and Changyin Sun. "Supervised class-specific dictionary learning for sparse modeling in action recognition". In: *Pattern Recognition* 45.11 (2012), pp. 3902–3911.

[177] Heng Wang and Cordelia Schmid. "Action Recognition with Improved Trajectories". In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2013, pp. 3551–3558. DOI: 10.1109/ICCV.2013.441. URL: https://doi.org/10.1109/ICCV.2013.441.

[178] Heng Wang, Alexander Kläser, Cordelia Schmid and Cheng-Lin Liu. "Action recognition by dense trajectories". In: *CVPR*. IEEE. 2011.

[179] Heng Wang, Dan Oneata, Jakob Verbeek and Cordelia Schmid. "A robust and efficient video representation for action recognition". In: *International Journal of Computer Vision* 119.3 (2016), pp. 219–238.

[180] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu and Song Zhu. "Cross-View Action Modeling, Learning, and Recognition". In: *CVPR* (2014).

[181] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang and Yihong Gong. "Locality-constrained linear coding for image classification". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. Citeseer. 2010, pp. 3360–3367.

[182] Limin Wang, Yu Qiao and Xiaoou Tang. "Action recognition with trajectory-pooled deep-convolutional descriptors". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 4305–4314. DOI: 10.1109/CVPR.2015.7299059. URL: https://doi.org/10.1109/CVPR.2015.7299059.

[183] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang and Luc Van Gool. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition". In: *ECCV 2016*. 2016, pp. 20–36. DOI: 10.1007/978-3-319-46484-8_2. URL: https://doi.org/10.1007/978-3-319-46484-8_2.

[184] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey". In: *Neurocomputing* 312 (2018), pp. 135–153.

[185] Xingxing Wang, LiMin Wang and Yu Qiao. "A comparative study of encoding, pooling and normalization methods for action recognition". In: *Asian Conference on Computer Vision*. Springer. 2012, pp. 572–585.

[186] Yifan Wang, Jie Song, Limin Wang, Luc Van Gool and Otmar Hilliges. "Two-Stream SR-CNNs for Action Recognition in Videos." In: *BMVC*. 2016.

[187] Daniel Weinland, Remi Ronfard and Edmond Boyer. "Free viewpoint action recognition using motion history volumes". In: *Computer vision and image understanding 2006* 104.2-3 (2006), pp. 249–257.

[188] Daniel Weinland, Remi Ronfard and Edmond Boyer. "A survey of vision-based methods for action representation, segmentation and recognition". In: *Computer vision and image understanding* 115.2 (2011), pp. 224–241.

[189] Karl R. Weiss, Taghi M. Khoshgoftaar and Dingding Wang. "A survey of transfer learning". In: *J. Big Data* 3 (2016).

[190] Geert Willems, Tinne Tuytelaars and Luc Van Gool. "An efficient dense and scale-invariant spatio-temporal interest point detector". In: *European conference on computer vision*. Springer. 2008, pp. 650–663.

[191] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell and Alex Paul Pentland. "Pfinder: Real-time tracking of the human body". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 780–785.

[192] Di Wu and Ling Shao. "Multi-max-margin support vector machine for multi-source human action recognition". In: *Neurocomputing* 127 (2014), pp. 98–103.

[193] Di Wu, Nabin Sharma and Michael Blumenstein. "Recent advances in video-based human action recognition using deep learning: a review". In: *IJCNN 2017*. IEEE. 2017, pp. 2865–2872.

[194] Xinxiao Wu and Yunde Jia. "View-invariant action recognition using latent kernelized structural SVM". In: *ECCV 2012*. Springer. 2012.

[195] Guangyou Xu and Feiyue Huang. "Viewpoint insensitive action recognition using envelop shape". In: *Asian Conference on Computer Vision*. Springer. 2007, pp. 477–486.

[196] Jianchao Yang, Kai Yu, Yihong Gong and Thomas Huang. "Linear spatial pyramid matching using sparse coding for image classification". In: *2009 IEEE Conference on computer vision and pattern recognition*. IEEE. 2009, pp. 1794–1801.

[197] Guangle Yao, Tao Lei and Jiandan Zhong. "A review of Convolutional-Neural-Network-based action recognition". In: *Pattern Recognition Letters* 118 (2019), pp. 14–22.

[198] Alper Yilmaz and Mubarak Shah. "Recognizing human actions in videos acquired by uncalibrated moving cameras". In: *ICCV 2005*. IEEE. 2005.

[199] Donghyun Yoo, Haoqi Fan, Vishnu Naresh Boddeti and Kris M Kitani. "Efficient k-shot learning with regularized deep networks". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[200] Jason Yosinski, Jeff Clune, Yoshua Bengio and Hod Lipson. "How transferable are features in deep neural networks?" In: *NIPS 2014*. 2014, pp. 3320–3328.

[201] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga and George Toderici. "Beyond short snippets: Deep networks for video classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4694–4702.

[202] Christopher Zach, Thomas Pock and Horst Bischof. "A duality based approach for realtime TV-L1 optical flow". In: *Joint Pattern Recognition Symposium 2017*. Springer. 2007, pp. 214–223.

[203] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao and Hanli Wang. "Real-Time Action Recognition with Enhanced Motion Vector CNNs". In: *IEEE CVPR 2016*. 2016, pp. 2718–2726. DOI: 10.1109/CVPR.2016.297. URL: https://doi.org/10.1109/CVPR.2016.297.

[204] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du and Duan-Sheng Chen. "A comprehensive survey of vision-based human action recognition methods". In: *Sensors* 19.5 (2019), p. 1005.

[205] J. Zhang, L. Zhang, H. P. H. Shum and L. Shao. "Arbitrary view action recognition via transfer dictionary learning on synthetic training data". In: *ICRA*. 2016.

[206] Jingtian Zhang, Hubert P. H. Shum, Jungong Han and Ling Shao. "Action Recognition From Arbitrary Views Using Transferable Dictionary Learning". In: *IEEE Trans. Image Processing* (2018).

[207] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang and Zhen Li. "A review on human activity recognition using vision-based method". In: *Journal of healthcare engineering* 2017 (2017).

[208] Zhong Zhang, Chunheng Wang, Baihua Xiao, Wen Zhou, Shuang Liu and Cunzhao Shi. "Cross-view action recognition via a continuous virtual path". In: *IEEE CVPR 2013*. 2013.

[209] Xiantong Zhen and Ling Shao. "Action recognition via spatio-temporal local features: A comprehensive study". In: *Image and Vision Computing* 50 (2016), pp. 1–13.

[210] Jingjing Zheng and Zhuolin Jiang. "Learning view-invariant sparse representations for cross-view action recognition". In: *IEEE ICCV 2013*. 2013.

[211] Jingjing Zheng, Zhuolin Jiang, P Jonathon Phillips and Rama Chellappa. "Cross-View Action Recognition via a Transferable Dictionary Pair." In: *BMVC 2012*. 2012.

[212] Fan Zhu and Ling Shao. "Correspondence-free dictionary learning for cross-view action recognition". In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 4525–4530.

[213] Fan Zhu and Ling Shao. "Weakly-supervised cross-domain dictionary learning for visual recognition". In: *International Journal of Computer Vision* 109.1-2 (2014), pp. 42–59.

[214] Maryam Ziaeefard and Robert Bergevin. "Semantic human activity recognition: A literature review". In: *Pattern Recognition* 48.8 (2015), pp. 2329–2345.