Miguel António Figueiredo Moura Diogo

Skeleton Fusion for Gestures Recognition in Augmented Reality Environments

Master's Dissertation in MIEEC, supervised by

Prof.Dr.Paulo Jose Monteiro Peixoto and presented to the

Faculty of Science and Technology of the University of Coimbra

January of 2020

Universidade de Coimbra

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Skeleton Fusion for Gestures Recognition in Augmented Reality Environments

Miguel António Figueiredo Moura Diogo

January of 2020

Skeleton Fusion for Gestures Recognition in Augmented Reality Environments

Supervisor:

Prof.Dr.Paulo Jose Monteiro Peixoto

Co-Supervisor:

Dr.João Luis Ruivo Carvalho Paulo

Jury:

Jorge Manuel Moreira de Campos Pereira Batista

João Pedro de Almeida Barreto

Paulo José Monteiro Peixoto

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering.

January of 2020

Acknowledgements

Primeiramente deixo o meu agredimento ao Professor Peixoto por ter sido o meu orientador, e por todas suas as indicações que no decorrer da dissertação que desenvolveram a minha capacidade pesquisar informação científica.

A toda a equipa do laboratório, em especial ao Dr.João Paulo por todo o tempo me disponibilizado assim como por toda a orientação e conhecimento partilhado que foram vitais para a concretização deste trabalho.

Quero também agradecer imenso à minha mãe por todo o apoio incondicional que meu deu e por todo o esforço que fez para garantir o meu sucesso académico.

Por último, deixo um obrigado a todos amigos que fiz no departamento, e vizinhos na residência universitária, pois sem a vossa amizade o tempo na universiade não teria passado tão rápido.

Resumo

Inteligência artificial (IA) é uma área da computação responsável por criar algoritmos capazes de realizar tarefas que requerem inteligência humana. Uma destas tarefas é reconhecimento de gestos humanos, que tem como objectivo analisar os movimentos do corpo humano ao longo do tempo por forma a discriminar/distinguir diferentes gestos. Reconhecimento de gestos implica capacidade de sentir a pose desse humano ao longo do tempo, o que geralmente é feito com câmaras e recorrendo outra área de IA chamada visão por computador.

Esta dissertação propõe um pipeline que reconhece gestos humanos a partir de 4 câmaras *Microsoft Kinect V2*. O *pipeline* proposto pode ser divido em 3 partes: fusão de *skeleton data* gerada por 4 câmaras RGB-D, codificação numa imagem da informação fundida e reconhecimento de gestos a partir dessas imagens através de algoritmos de aprendizagem de máquina. De cada câmara é obtida uma série temporal de posições 3D de juntas. Para obter posições tridimensionais, duas das coordenadas são calculadas por *OpenPose*, e a restante provém da informação de profundidade lida pelas câmaras. As quatro séries temporais são fundidas com um filtro de Kalman. Na segunda parte do pipeline, a série temporal é codificada numa imagem. Dois métodos diferentes são testados para a codificação da série temporal numa imagem: *gramian angular fields* e *recurrence plots*. Por último uma rede neural convolucional (CNN) é usada para distinguir sequências de gestos codificadas nas imagens.

O nosso *pipeline* consegui obter uma precisão de 87.8% no nosso dataset usando a codificação *recurrence plot*. No entanto, o nosso algoritmo de codificação de *skeleton data* em imagens e alimentação de uma CNN com essas imagens foi testado não só com um *dataset* nosso, mas também com outros 2 públicos.

Abstract

Artificial Intelligence (AI) is a field of computer science responsible for creating algorithms capable of executing tasks that have traditionally required human intelligence. One of these tasks is Human Action Recognition (HAR), whose purpose is to analyze human body movements through time and differentiate between different actions. HAR algorithms rely on the capacity to sense a human body's pose through time, which is generally done with cameras through another field in AI called computer vision.

This thesis proposes a pipeline that recognizes human actions from 4 cameras Microsoft Kinect V2. The proposed pipeline can be divided into three parts: the fusion of skeleton data attained from 4 RGB-D cameras, the conversion of the fused data into an image, and action recognition from those images through machine learning algorithms. A time series of 3D joints is extracted from each one of the four cameras. Two of the joint coordinates are computed by the OpenPose algorithm, and the remaining one comes from depth information measured by the cameras.The four time series are fused with a Kalman filter. On the second part of the pipeline, the time series is converted into an image. Two different methods are tested to convert a time series into an image: the gramian angular fields and recurrence plots. Finally, the image that encodes skeleton data is feed into a convolutional neuronal network (CNN) to recognize the action sequence being performed.

Our pipeline manages to attain an accuracy of 87.8% on our dataset while recurrence plots to encode time series into an image. Nevertheless, our algorithm to convert time series into images and feed those images into a CNN was tested with our dataset and two other public datasets.

The purpose of (scientific) computing is insight, not numbers.

— Richard Hamming,

Contents

List of Acronyms

AAL Ambient Assisted Living.

AI Artificial Intelligence.

CCTV Closed-circuit television.

CNN convolutional neuronal network.

DPRL Deep Progressive Reinforce-ment Learning.

GAF Gramian angular Fields.

HAR Human Action Recognition.

RGB Red-Blue-Green.

RGB-D Red-Blue-Green-Depth.

RNN Recurrent Neural Network.

RP Recurrence plot.

ToF Time of Flight.

List of Figures

List of Tables

1 Introduction

1.1 Motivation and context

Researchers have explored different compact representations of human actions in the past few decades [1]. One of the most influential works was done by Johansson [2] in the field of psychology. His experiment consisted in studying 3D human motion perception from 2D patterns. Johansson placed several bright spots distributed on the human body against a homogeneous contrasting background. The experiment demonstrated that human vision detects motion directions and different limb motion patterns from those bright spots and the velocity in which those patterns were being performed. He also noticed that the number of light spots and their distribution on the human body might affect motion perception [2], "The geometric structures of body motion patterns in man [...] are determined by the construction of their skeletons."

Since the earliest works in Human Action Recognition (HAR) algorithms, three decades ago, [3],[4], the interest in automatic HAR has grown considerably in the last few years, greatly due to advances in deep learning-based methods. Deep learning has become a reference methodology for obtaining state-of-the-art performance in HAR. The literature in human action recognition is already extensive in several fields, including computer vision, machine learning, pattern recognition, signal processing, and many more [5],[6],[7].

Automatic recognition of human actions from video footage has a vast number of applications in various fields, such as behavior analysis [8], surveillance where suspicious or violent human activities can be automatically identified from video [9],[10]. Action recognition also has many applications in healthcare. Such applications include developing patient monitoring systems [11], [12] to track patients' daily activities and give real-time feedback about their progress. To help patients suffering from the declined mental ability or mental disorders, which must be monitored continuously to identify unusual actions in time and thus prevent unwanted consequences [13]. It can also be used in Ambient Assisted Living (AAL)

[14],[15], or to analyze lower limb locomotion [16] and to an accurate assessment of physical activity and analyze the daily energy expenditure of a patient [17],[18]. Another field whose applications are vast is in improving human-computer interfaces [19], [20], [21], or complementing existent ones since hand movements that accompany speech are an integral part of communication and very often influence the meaning taken from speech. Action recognition can also be used to interpret and translate sing languages.There are Gaming applications as well since it was the demand for creating more immersive videogames that led to the development of the RBG-D Microsoft Kinect camera, making depth imaging technology available at a consumer price point and allowing for drastic advances in the conception of depth maps. Autonomous driving vehicles use HAR to provide a more user-friendly and safer interaction between human and automated vehicles [22], [23]. Lastly, HAR algorithms can also be used to automatically index human activities in a video, facilitating the search of specific events.

1.2 Problem Formulation

There are several limiting factors when it comes to developing a system that can recognize human action. Firstly, depending on the type of sensor technology employed, di□erent problems emerge. Sensor-based approaches found in Wearable devices use accelerometers, gyroscopes, and magnetometers to measure the individual pose of each joint of a human body. For that reason, to model all body parts, several wearable devices are needs making the setup process slow and very unpractical. Alternatively, HAR can be vision-based, relying on scanners or cameras to sense depth in scenes. The most accurate and precise technology in this category is motion capture. Nevertheless, motion capture systems are usually costly, and it is a marker-based technology which implies having long setup times.

Notwithstanding, several vision-based technologies do not require markers. Such technologies include stereo vision(passive or active), time of flight, and structure light cameras, which are relatively cheap and increasingly more common in general quotidian devices such as smartphones and CCTV. The most significant limitation of these technologies, when compared with motion capture systems, is their noisy images measurements, forcing the adoption of more computational complex algorithms in order to mitigate the noisy readings. Body occlusions are another limiting factor for these technologies, and however, it can be eased by increasing the number of views of the scene.

The type of sensors employed might also be a limiting factor in the system's accuracy. RGB cameras produce rich texture data of the subjects and their background and are usu-

ally cheap. However, systems that only rely on RGB cameras are susceptible to illumination variations in the scene, and in some situations, the subject's texture blend with the background texture. For that reason, RGB-D cameras were developed, they work better than conventional RGB cameras in low light environments, and they are robust against lighting conditions and illumination changes. Furthermore, depth data is provided directly without the need for extra calculations. On the other hand, RGB-D cameras usually have low resolution and low sensitivity introducing even more noise into the measurements/images. Also, RGB-D cameras can be easily a□ected by some materials, such as light-absorbing and transparent materials.

Additionally, the significant di□culty in body-pose estimation is an enormous range of poses that the human body is capable of, which are di□cult to simulate and account for, non-including the extra steps necessary to account for biometric di□erences, di□erent viewing angles, and data normalization.

Even with a reliable skeleton estimation, 3D skeleton-based action classification is not that simple as it may appear. In this sense, "one of the biggest challenges of using posed-based features is that semantically similar motions may not necessarily be numerically similar" [24]. Moreover, motion is ambiguous, and action classes can share movements [25].

Another very challenging issue is the extra (temporal) dimension in sequences typically turned action recognition into a challenging problem in terms of both amounts of data to be processed and model complexity. Therefore, there is always a trade-o□ between computational e□ciency and accuracy. Some approaches using local temporal-spatial features limit their ability to recognize long and complex actions, and others struggle to deal with actions sequences with varying temporal duration.

1.3 Objectives and main contributions

The main objective of this master's dissertation was to build a system able to recognize human actions from video sequences composed of depth images. Depicted in Figure 1.1is an overview of the proposed pipeline. The proposed system architecture is divided into five steps:

- Extract data from file: Extract the 3D skeleton data for each one of the four Microsoft Kinect v2 in the multi-view RGB-D camera system previously calibrated.
- Create fused 3D skeleton: Define an algorithm to fuse the Define an algorithm to

fuse the multiple multi-sensorial information, then use it to fuse the 3D skeleton data. Encode the data: Encode the fused 3D skeleton data into a sequence of images. This encoding describes the space-time behavior of person movement.

- Classification of human actions: Propose an algorithm to recognize and classify human actions, which uses as input the previous sequence of images previously calculated. Test the HAR algorithm: Train and test the HAR algorithms with internal and public datasets of 3D skeleton data.

The main contribution of this work is to introduce a four-camera multi-view setup to prevent body occlusion and to encode the segmented time series composed of the fused 3D skeleton into a sequence of images using gramian angular fields and recurrence plots to encode that time series.

The structure/organization of this dissertation and the content of each chapter are the following:

- Chapter 2 reviews state of the art on HAR;
- Chapter 3 presents and briefly describes methods used to pre-processing and encode used in this work;
- Chapter 4 describes the entire architecture used;
- Chapter 5, the results obtained are presented and discussed;
- Chapter 6 draws some conclusions, as well as some proposals for future work;

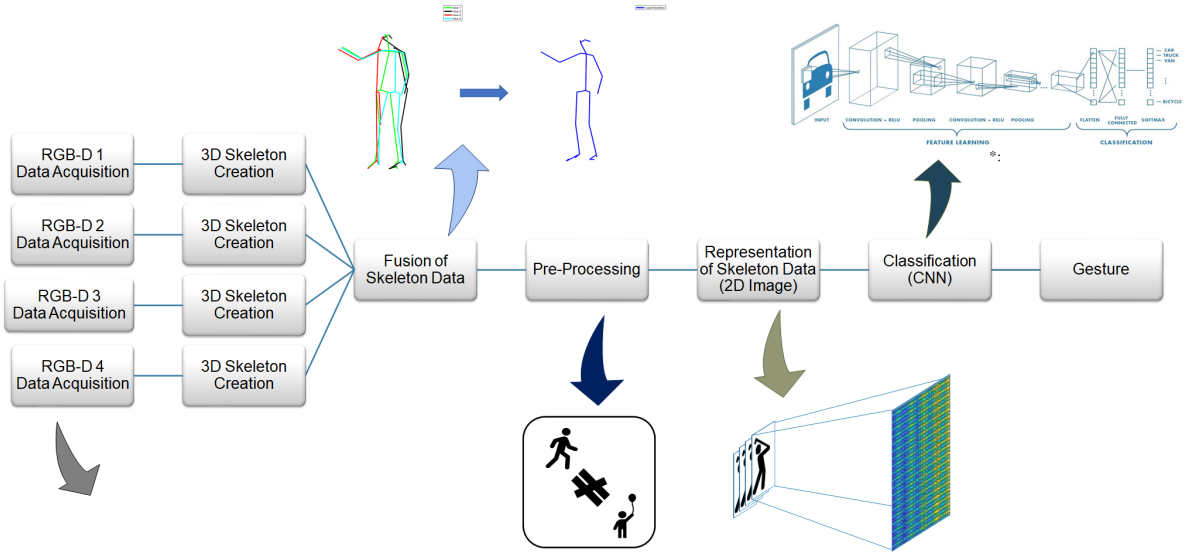

Figure 1.1: The proposed system architecture ,(*: Convolutional Neural Networks[26])

2 State of the art

2.1 Human activities

According to [6] and [7], human activities can be categorized into four levels: gestures, actions, interactions, and group activities. A gesture is the most straightforward activity of the four, and it is defined as an elementary movement representing a specific meaning or idea. The definition of action is a single-person activity that may include several gestures. An interaction is an activity that involves two agents, one being a person and the other an object (human-object interaction) or a human (human-human interaction). Furthermore, a group activity is a type of activity that requires multiple individuals and may include interactions with one or more objects. These definitions will be the ones used throughout this document. Nonetheless, some authors have slightly di□erent definitions of actions and gestures. According to [27], action is a body movement produced to achieve a goal, while a gesture is a type of action described as being a simulated action. In [28], a gesture is also defined as a type of action with an essential and unique role in human communication and whose presence a□ects how the information is received.

2.2 Skeleton data

As first introduced in [2], the computer vision community defines a skeleton as a model of an articulated human body formed by a hierarchy of joints connected by bones. The motion of such a skeleton can be used as a representation of gestures/actions. Therefore, the human body pose is defined by the relative locations of the joints in the skeleton.

A person detector [29], [30], [31],[32], [33] extracts skeleton data from 2D images. Person detectors generally comprise two steps: Part detectors and pictorial structures (PS) models. Part detectors are are algorithms that can find specific object in this case specific body part [34],[35] [36], [37], [38], [39], [40], [41], [42], [43], [44], [45]. The pictorial structures

model [46], [47], [48] is a deformable model that knows where each body part belongs to in a human body in order to build skeleton data reliably enough to infer the actual human body pose based on constraints among body parts. In general, such constraints are meant to represent the actual human body articulations. Inferring the pose of multiple people in images is another very challenging problem, especially with socially engaged individuals, not just due to the unknown number of people that can occur at any position or scale but also due to the interactions between each other, which can result in body occlusion and complex spatial interference due to physical contact. According to [49]], there are two di□erent approaches to solving the multi-person problem: top-down and bottom-up approaches. Top-down approaches employ a person detector and perform single-person pose estimation for each detection. According to [49], the computational cost of this approach is proportional to the number of people in the scene, and if the person detector fails, as it is prone to, the information related to the body pose is lost. In contrast, bottom-up approaches are more robust and very often are easier to compute. One of the most popular 2D multi-person pose estimators is OpenPose, a bottom-up approach that uses deep learning to detect body parts and then Part A□nity Fields (PAFs) to learn to associate them [49].

The wide di□usion of cheap cameras capable of generating depth maps combined with the work done by [50] allowed for 3D skeleton data to be extracted directly from depth maps alone. Depth maps proved to be extremely useful in providing data for an easy and fast human body estimation. Estimating the 3D joints from RGB imagery is subject to errors and high computational cost. However, with the use of Kinect, we can acquire the 3D locations of the body parts in real-time with better accuracy [51].

2.2.1 Pre-processing of skeleton data

Pre-processing of skeletal data is often used to cope with biometric di□erences among subjects and varying temporal duration of the sequences due to di□erent velocities of the actions and inter-subjects style variations.

Data normalization and biometric differences

When skeletal data is employed to recognize human activities, it is crucial to account for biometric di□erences between individuals, adjust for the variety of coordinate systems of those individuals, and consider that people in the scene may be scaled di□erently. The lack of these kinds of data normalization will force the increase of complexity of the classifier to

cope with these variations instead of just analyzing the joints' movement in the scene.

To tackle the issue of random viewpoints or multiple coordinate system changes in the case of various people in the scene, the work in [52] initially registers the data into a common coordinate system to make the joint coordinates comparable to each other. In [53],[54], all the 3D joint coordinates are transformed from the world coordinate system into a person-centric coordinate system by placing the hip center at the origin. In [55], skeletons are aligned based on the head location. All joint locations are associated with body parts normalized by the head length to eliminate the influence of scale and translation. Similarly, in [56], human poses are normalized by aligning torsos and shoulders.

Works such as [57] consider the joint angles between any two connected limbs and represent an action as a time series of joint angles, not just to ignore biometric differences but also to select the most informative joints in the sequence.

To make it scale-invariant in [58], the coordinates of the 3D joints are normalized to the interval between 0 and 1 in all the dimensions over the sequence. Whereas [59] uses Spherical Coordinates of Histogram, placing any 3D joint into spatial histogram bins. The pitch angle is divided into x_1 bins, and similarly, the yaw angle is divided into x_2 equal bins that result in the x_1 times x_2 bins that form the histogram. The radial distance is not used in this representation to make the method scale-invariant.

Representing time and accounting for time-varying sequences

Analyzing the space-time volume of a video is changeling mainly due to the temporal dimension. Even though sometimes it is possible to recognize some actions using only one frame for most actions, a sequence of frames or key poses is needed. Hence one of the main issues is dealing with the sequences with varying lengths and durations. These differences are typically caused by the vast types of actions and the velocity and style with which a certain action is performed. The additional step of encoding the order of the sequence may be required, for instance, to distinguish between the action of pushing and pulling.

Works such as [59] adopt global feature representation of the entire sequence sacrificing, in general, the information about the temporal structure of the sequence. In the specific work case [59], a key pose is inferred from every frame and then fed into a hidden Markov model. In [60],[58],[61], time representation was done by adopting a temporal pyramidal approach [62]. Approaches adopting some distance among the trajectories of 3D joints use the dynamic time warping (DTW) algorithm. This type of approach was used by [63],

which from a bag of key poses, found a similar training sequence by considering a temporal alignment of the involved key poses. Another way to create representations of the same length is to divide the sequence into a prefixed number of temporal segments. However, the problem of finding the most appropriate number of segments to use across all the classes and sequences is not trivial. Therefore, [57],[25] propose to divide the action sequence into a varying number of temporal segments, each of the same temporal duration. In particular, [25] adopts a sliding window approach where the temporal segments are partially overlapping. The sliding window approach makes the method more robust to the temporal warping of the sequence, and for that reason, it is usually used along with deep learning algorithms.

2.3 3D human action representation and classification

Human action recognition can be thought of as the union of di□erent fields of artificial intelligence like computer vision and machine learning. Computer vision is used to analyze the space-time volume of video footage and recognize the di□erent objects in the scene and their movement through time. At the same time, machine learning is used to learn the patterns in those objects' trajectories to distinguish between di□erent classes of human gestures or actions. In this chapter, HAR will be divided into two di□erent categories, HAR from skeleton data or model-based and non-model-based.

2.3.1 HAR non-model based

Analyzing the space-time volume of a video without having a model of the object (human)in the scene implies analyzing each frame to extract patterns in these frames, consequently achieving HAR. There are various computer-vision techniques to analyze images, and these can be decomposed into two groups: Deep-learning methods and traditional hand-crafted methods.

Traditional Hand-crafted involves finding a way to match images that inscribe the same human action. For that, a bag of visual words [64][65], (figure 2.1) is the most popular hand-crafted method. To find visual words in an image and construct a visual dictionary, images have to be sampled. Dense sampling outperforms sparse sampling [66],[67]. Later, dense trajectories [68] and improved dense trajectories were introduced as improved variants of dense sampling. They take into consideration the movement of the dense samples through time, once taking into consideration the temporal dimension in HAR algorithms is essential

to have good accuracy, as earlier work using histogram of oriented gradients (HOG) [69] and histogram of optical flow (HOF) [70],[71] as shown.

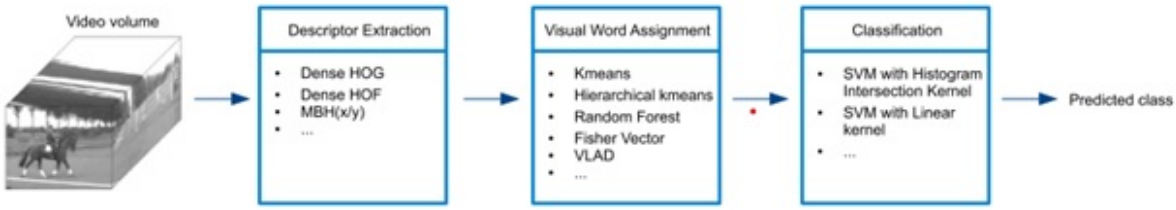

Figure 2.1: General framework for video classification using Bag-of-Visual-Words [65]

In deep learning approaches, the temporal dimension is equally important. The extra dimension in sequences typically turned action/gesture recognition into a challenging problem regarding both amounts of data to be processed and model complexity, which are crucial aspects for training large parametric deep learning networks. Based on how it is dealt with, deep learning approaches can be categorized into four non-mutually exclusive groups.

The first group consists of 2D CNNs, which can exploit appearance(spatial) information. These approaches [72],[73] sample one or more frames from the whole video and then apply pre-trained 2D models on each frame separately. They finally label the actions by averaging the result of the sampled frames. The main advantage of this kind of model is its possibility to use pre-trained models on larger image datasets, such as ImageNet [74]. Gesture recognition methods mainly fall into this category [75],[76],[77].

Methods in the second group first extract 2D motion features like optical flow and then utilize these features as a di□erent input channel of 2D convolutional networks [78],[79],[80] [81]. In other words, these methods consider the temporal information from the pre-computed motion features. Third group uses 3D filters in the convolutional layers [82], [83], [84]. The 3D convolution and 3D pooling allow capturing discriminative features along both spatial and temporal dimensions while maintaining the temporal structure in contrast to 2D convolutional layers. The spatiotemporal features extracted by this model have proven to surpass 2D models trained on the same video frames.

Finally, the fourth group combines 2D (or 3D) convolutional nets, which are applied at individual (or stacks of) frames, with a temporal sequence modeling. Recurrent NeuralNetwork (RNN) [85] is one of the most used networks for this task, considering the temporal data using recurrent connections in hidden layers. The drawback of this network is its short memory which is insu□cient for real-world actions. LongShort-Term Memory (LSTM) networks [86] were proposed to solve this problem, as they are a variant of RNN. Bidirectional RNN

(B-RRN) [87], Hierarchical RNN (H-RNN) [88], and Di□erential RNN (D-RNN) [89] are some successful extensions of RNN in recognizing human actions. Other temporal modeling tools like HMM are also applied [90] in this context. For all methods in the four groups, their performance can be boosted by combining its output with additional hand-crafted features [83], improved dense trajectories (IDT) [79].

2.3.2 HAR model based

The use of skeleton data reduces the complexity of the HAR algorithm, as conjecture in [91]. Using skeleton data alone for action recognition can perform better than using other low level image data.

According to [92] 3D human action representation with skeleton data can be grouped in three distinct categories: joint-based representations, mined joint-based descriptors and dynamics-based descriptors.

Mined joint-based descriptors

Detection of the activated subsets of joints can help to discriminate among di□erent action classes. Methods such as [57],[55],[56] focus on mining the subsets of most discriminative joints or consider the correlation of subsets of joints.

The method in [60] uses a spherical coordinate system to model each joint by its location and velocity. An action sequence is modeled as histograms, each computed on a specific feature and joint. A partial least square (PLS) [93] is used to weigh the importance of the joints, and kernel-PLS SVM [94] is adopted for classification purposes. The approach in [95] employs a genetic algorithm to select the joints that represent an action class. The method in [96] adopts multi-part modeling of the body. The coordinates of each joint are expressed in a local reference system, which is defined at the preceding joint in the chain. Body sub-parts are aligned separately, and a modified nearest-neighbor classifier is used to perform action classification by learning the most informative body parts.

Dynamics-based descriptors

Skeleton-based action methods in this category focus on modeling the dynamics of either subsets or all the joints in the skeleton. This can be accomplished by considering linear dynamical systems (LDS) [54],[97] or hidden Markov models (HMM) or mixed approaches [25]. Like in [54], the skeleton sequence is represented as a set of time series (one for each

body part) of features such as position, tangents, and shape context. Each feature time series is modeled using an LDS, and the method learns the corresponding system parameters by performing system identification. The estimated parameters are used to represent the action sequence. Multiple kernel learning (MKL) [98]] is used to learn a set of optimal weights for each part configuration and temporal extent.

In [25],[99], autoregressive models are used to represent the 3D joint trajectories, represented by an Hankelet [100]. A subspace distance to compare Hankel matrices is approximated through a dissimilarity score [[101]]. Using a sliding window approach, an action is represented as a sequence of Hankelets. An HMM allows modeling the transition from one LTI system to another, yielding a model for switching dynamical systems.

Joint-based representations

The methods in this category analyze locations of joints and their variation through time. This category can be organized into three sub-categories: spatial descriptors, geometric descriptors, and key-pose-based descriptors.

Spatial descriptors represent the body pose through the correlation of the 3D body joints. In [102], that correlating three pieces of information: all the pairwise distance of 3D joints in the current frame, those distances between the current frame and the previous ones, and between the current frame and a neutral pose. A similar approach [103] uses pairwise distances between joints as a feature. Principal component analysis (PCA) is employed to reduce the dimensionality of the feature space, and a Naive-Bayes-nearest-neighbor classifier is used for action classification. Pairwise distances were also used in [53], the main difference is that HMM was used to classify actions, and a deep neuronal network emitted the probabilities. Furthermore, later works attempt to capture the correlation between joints by representing a skeleton sequence through its covariance [58].

Hand-crafted feature-based methods require an active engagement along with many efforts to extract spatial and temporal features from skeleton sequences. Sometimes, it becomes more complicated to design discriminative features from the 3D skeleton videos, which degrades the system's performance. For that reason, and due to its excellent performance, deep learning has become the most popular approach for HAR.

According to [104], deep learning methods using skeleton data can be grouped into three non-mutually exclusive groups.

The RNN [105] based methods are naturally suitable for sequence data. However, the

well-known gradient vanishing problems are inevitable, and therefore LSTM [106] and GRU may more relent to those problems to some extent.

The CNN-based methods differ from RNNs because CNN models can efficiently and quickly learn high-level semantic cues with their naturally equipped excellent ability to extract high-level information. However, CNNs generally focus on image-based tasks, and the action recognition tasks based on skeleton sequence are unquestionable a heavy time-dependent problem. Therefore, balancing and more fully utilizing spatial and temporal information in CNN-based architecture is still challenging. In [107], the correlation among joints' locations is done through convolutional neural networks.

The graph and convolutional neural network (GCN) based methods [108], like the two-stream adaptive GCN [109], and graph convolutional networks [108], according to [104]] achieve high accuracy on some popular public datasets. Likewise, according to [104], mixed-methods like 3S-CNN+Multi-Task Ensemble Learning [110], Richly Activated GCN [111], and Semantics Guided GCN [112], , performed very well on same those public datasets. Moreover, view adaptive neuronal networks such as view adaptive neural networks based on RNN (VA-RNN) and based on CNN (VA-CNN) presented outstanding experimental results [113].

Spatial descriptors are a representation that lacks any temporal information and may result in ambiguous descriptions of the action sequence.

Geometrical descriptors attempt to represent a skeleton utilizing the geometric relations between different body parts. The work done in [114] consists of a set of boolean features, each associated with a quadruple of joints, where three of them are used to identify a plane, and the other boolean feature indicate if the point is in front or behind the plane. This kind of feature allows representing the geometric relations among sets of joints and is robust to spatial variations, global orientation changes, and the size of the skeleton.

Similarly, in [115], joints are considered in quadruples. In this case, two out of the four joints in the quadruple are used to set a coordinate system where one of the points is used as origin, while the most distant point in the quadruple is represented in the new coordinate system as [1,1,1]. A skeleton is then represented as a set of skeletal quads. Then for each class, a Gaussian mixture model is trained, and finally, a multi-class linear SVM and perform action classification.

The representation introduced in [52] explicitly estimated the relative 3D geometry between different body parts. In a nutshell, given two rigid body parts, their relative geometry can be described by considering a rigid-body transformation [6] to align one body part to

the other. Methods [116],[60] estimate the geometric relations between the whole skeleton in a sequence rather than its body parts.

Key-pose based descriptors describe methods that learn a dictionary/codebook of key-poses and represent an action sequence in terms of these key-poses. The most common form of key-poses-based descriptors uses a histogram of motion words [57], where the concatenated 3D joint locations are clustered into K distinctive using K-means. Each action sequence is represented by counting the number of detected motion words. Motion words are detected by assigning each skeleton representation to its closest distinctive pose. The method in [117] divides the body into four regions. Each body region is represented by a feature vector of 21 dimensions comprising the line-to-line angles between body joints and the six line-to-plane angles. A dictionary of body poses is obtained by a standard K-means clustering algorithm. A hierarchical model is trained to represent complex activities as a mixture of simpler actions. In [118], a decision tree forest recognizes action from key poses.

3 Methods

3.1 Kalman Filter

Kalman filters [119] are used to estimate states variable of linear and dynamic systems. This recursive estimation technique was developed around 1960, most notably by R.E. Kalman [120]. In practice, measurements and processes are inherently nosy. For that reason, algorithms like Kalman filters are implemented to decrease the overall uncertainty of the system. A Kalman filter works by combining the estimated state variables with their correspondent measurements in an optimal way, which increases the estimated state's prediction accuracy. Another great advantage of the Kalman filter is its ability to link more than one measurement/observation into a single state variable concurrently and optimally, making this filter great not just to mitigate the overall noise of the system but also to fuse multiple sensory data, with a relatively smaller footprint in terms of computational power.

Prior to implementing the Kalman filter, a non-deterministic model of the system must be designed, describing the autocorrelation sequence of the process signals. In equation3.1, the signal is modeled by a simple first-order recursive filter driven by zero-mean white noise.

$$\hat{x}\,(k) = a\,(k)\,\hat{x}\,(k-1) + w\,(k-1) \tag{3.1}$$

$$E[w(k)w(j)] = \begin{cases} \sigma_w^2, & k = j \\ 0, & k = j \end{cases} \tag{3.2}$$

The equation 3.2 represents the white noise. The observation model is also non-deterministic, and it must be assumed linear, as described below (equation 3.3).

$$y(k) = c.x(k) + v(k) \tag{3.3}$$

The time-varying random signal is described in eq.3.3, and factor c represents an ob-

servation (or measurement) parameter, $v(k)$ is an independent additive white noise with zero-mean and variance σ_v^2.

The Kalman filter is a recursive estimator defining the evolution of the state from time k-1 to time k.

$$\hat{x}(k) = a(k)\hat{x}(k-1) + b(k)y(k) \tag{3.4}$$

The first term represents the previous weighted estimate (at time k-1), and the second term is the weighted present (time k) measure/ data sample. In this case, we have two parameters, $a(k)$ and $b(k)$, to be determined from the minimization of the mean-square error

$$p(k) = E[e^2(k)] \tag{3.5}$$

where $e(k) = \hat{x}(k) - x(k)$ is the error.

This minimization of the mean-square error is why the Kalman filter is an optimum recursive filter. However if neither the process model nor the observation model follows a gaussian distribution, the solution may not be optimal.

The equation 3.6, was obtained by replacing $\hat{x}(k)$ in 3.5, and by differentiating 3.6 concerning $a(k)$ and $b(k)$ obtain the expression 3.7. More detailed information can be found in [121].

$$p(k) = E[a(k)\hat{x}(k-1) + b(k)y(k) - x(k)]^2 \tag{3.6}$$

$$a(k) = a[1 - c.b(k)] \tag{3.7}$$

Subsequently by applying equation 3.7, we attain the optimum recursive estimator equation 3.8.

$$\hat{x}(k) = a\hat{x}(k-1) + b(k)[y(k) - ac\hat{x}(k-1)] \tag{3.8}$$

The Kalman filter algorithm is known for being composed of two stages. Therefore by decomposing equation 3.8, in two terms, we can obtain those stages. The first term, $\hat{x}(k-1)$, represents the best estimate without any additional information, and it is a prediction based on past observations that are often called the prediction or propagation stage. The second term is a correction/update term involving the difference between the new data sample and

the observation estimate, $\hat{y}(k) = c\hat{x}(k)$, weighted by a variable gain factor $b(k)$, known as Kalman filter gain.

$$b(k) = \frac{c[a^2 p(k-1) + \sigma_w^2]}{\sigma_v^2 + c^2\sigma_w^2 + c^2 a^2 p(k-1)} \tag{3.9}$$

As described above, the Kalman filter can process multidimensional signals from either state variables or observations, and this is achieved simply by writing multidimensional signals as vectors.

The vectors $x(k)$ and $w(K)$ are q-dimensional vectors of the q signals and q white noise driving processes, respectively.

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_q(k) \end{bmatrix} \qquad w(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_q(k) \end{bmatrix}$$

Hence, equation 3.10 defines the first-order dynamic's vector, where A is a $(q \times q)$ matrix.

$$x(k) = Ax(k-1) + w(k-1) \tag{3.10}$$

While in the observation model, y(k) and v(k) are $(r \times 1)$vectors, and where C is an $(r \times q)$ observation matrix, which in this case, assuming $r < q$, is given by

$$C = \begin{bmatrix} c_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & c_2 & & \vdots & & \vdots \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \dots & c_r & \dots & 0 \end{bmatrix}_{r \times q} \qquad y(k) = Cx(k) + v(k)$$

The transition from the observation noise variance to the noise covariance matrix is written as

$$\sigma_v^2 = E[v_1^2(k)] \rightarrow R(k) = E[v(k)v^T(k)] \tag{3.11}$$

Similarly, for the system noise, we have

$$\sigma_w^2 = E[w_1^2(k)] \rightarrow Q(k) = E[w(k)w^T(k)] \tag{3.12}$$

Recursive filter estimator:
$\hat{x}(k) = A\hat{x}(k-1) + K(k)[y(k) - CA\hat{x}(k-1)]$
Filter gain:
$K(k) = P_1(k)C^T[CP_1(k)C^T + R(k)]^{-1}$
where, $P_1(k) = AP(k-1)A^T + Q(k-1)]$
Mean-square error covariance matrix:
$P(K) = P_1(k) - K(k)C(k)P_1(k)$

Table 3.1: Vectorial Kalman filter equations

3.2 Encoding time series into images

3.2.1 Gramian angular fields

Gramian angular Fields (GAF) [122] is a framework used for encoding time series as a 2D image. It works by changing the coordinate system of the times series from cartesian coordinates into polar coordinates, using various operations to convert these angles into a symmetry matrix called Gramian Angular Field.

The first step to construct a GAF matrix is to normalize a given time series $X = x_1, x_2, x_3, \ldots, x_n$ of n real-valued observations, rescaling it so that all values fall in the interval $[-1, 1]$ or $[0, 1]$. The following equation shows the simple linear normalization method used, where $\widetilde{X}$ represents the normalized data.

$$\widetilde{X} = \frac{(x_i - max(X)) + (x_i - min(X))}{max(X) - min(X)} \quad , \widetilde{X} \quad [-1, 1] \tag{3.13}$$

$$\widetilde{X} = \frac{x_i - min(X)}{max(X) - min(X)} \quad , \widetilde{X} \quad [0, 1] \tag{3.14}$$

The rescaled times series $\widetilde{X}$ will be represent in polar coordinates by encoding the value as the angular cosine and time stamp as the radius as showed in eq.3.15.

$$\begin{cases} \phi = arccos(\tilde{x}_i) & , -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \quad \tilde{X} \\ r = \frac{t_i}{N} & , t_i \quad \mathbf{N} \end{cases} \tag{3.15}$$

In the equation above, t_i is the time stamp and N is a constant factor to regularize the span of the polar coordinate system. The encoding map of equation 3.15 has two import properties. First, it is bijective as $cos(\phi)$ is monotonic when ϕ [0, π].Moreover, unlike Cartesian coordinates, the polar coordinates preserve absolute temporal relations through the radius coordinate.

The rescaled data in di□erent intervals have di□erent angular bounds. The [0, 1] interval corresponds to the cosine function in [0, $\pi/2$], while cosine values in the interval [−1, 1] fall into the angular bounds [0, π]. They provide di□erent information granularity in the Gramian Angular Field for classification tasks.

After transforming the rescaled time series into the polar coordinate system, we can easily exploit the angular perspective by considering each point's trigonometric sum/di□erence to identify the temporal correlation within di□erent time intervals. The Gramian Summation Angular Field (GASF) and Gramian Di□erence Angular Field (GADF) are defined as follows:

$$GASF = [cos(\phi_i + \phi_j)] = \tilde{X} .\tilde{X} - \sqrt{I - (\tilde{X})^2}.\sqrt{I - \tilde{X}^2} \tag{3.16}$$

$$GASF = [sin(\phi_i - \phi_j)] = \sqrt{I - (\tilde{X})^2}.\tilde{X} - \tilde{X} .\sqrt{I - \tilde{X}^2} \tag{3.17}$$

I is the unit row vector [1, 1, . . . , 1]. After transforming to the polar coordinate system, we take time series at each time step as a 1-D metric space. By defining the inner product $< x, y >= x.y - \sqrt{1 - x^2}.\sqrt{1 - y^2}$ and $< x, y >= \sqrt{1 - x^2}.y - x.\sqrt{1 - y^2}$, two types of Gramian Angular Fields (GAFs).

G is a Gramian matrix:

$$G = \begin{bmatrix} cos(\phi_1 + \phi_1) & \dots & cos(\phi_1 + \phi_n) \\ cos(\phi_2 + \phi_1) & \dots & cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ cos(\phi_n + \phi_1) & \dots & cos(\phi_n + \phi_n) \end{bmatrix}$$
$$= \tilde{X} .\tilde{X} - \sqrt{I - (\tilde{X})^2}.\sqrt{I - \tilde{X}^2}$$

The GAFs has several advantages. First, they provide a way to preserve temporal dependency since time increases as the position moves from top-left to bottom-right. The GAFs contain temporal correlations because G (i,j ||j-j|=k) represents the relative correlation by superposition/di□erence of directions with respect to time interval k. The main diagonal

G(i,j) is when k = 0, which contains the original value/angular information. We can reconstruct the time series from the high-level features learned by the deep neural network from the main diagonal. However, the GAFs are large because the size of the Gramian matrix is n x n when the length of the raw time series is n.

3.2.2 Recurrence Plots

Using Recurrence plot (RP) [123] to analyze time series allows to visualize and quantify structures hidden in the data. RPs visualize the behavior of trajectories in phase space [1,2]. They are a graphical representation of the matrix described by the equation. 3.18

$$R_{i,j} = \Theta(\varepsilon - \left\|\vec{x}_i - \vec{x}_j\right\|), \qquad i,j = 1,\ldots,N \tag{3.18}$$

Heaviside function represented by Θ() in eq.3.18 defines a threshold value. One assigns a “black” dot to the value one and a “white” dot to the value zero. The two-dimensional graphical representation of $R_{i,j}$ then is called a RP. An unthresholded RP is not binary, but its matrix Rui,j is given by the (real-valued) distances of the vectors x_i and x_j. The matrix is then usually represented as a two-dimensional colored plot. It has been shown [124] that from an unthresholded RP, it is possible to reconstruct the time series. Nevertheless, unthresholded RPs are more challenging to quantify than binary RPs. For this reason, in data analysis usually, binary RPs are used.

4 Implementation

4.1 RGB-D Data Acquisition and 3D Skeleton Creation

As depicted in Fig.1.1, each of the four Microsoft Kinect V2 acquires a frame of RGB-D data with a frequency of 25Hz. A 2D Skeleton is extracted for each frame by applying the OpenPose [49] algorithm over the RGB values. The depth information of each pixel is projected into the 2D Skeleton to create the 3D Skeleton data.

4.2 Fusion of skeleton data

A Kalman filter performs the Fusion of 3D skeleton data from the four cameras. The Kalman filter uses the confidence values of the joint outputted by the OpenPose algorithm to optimally combine the joint positions from the four RGB-D cameras with a first-order process model. Dynamic models of the human musculoskeletal systems can be very complex. However, [125],[126] shows that a simple first-order model to represent the motion of each human body limb segment is sufficient for most motion tracking applications. It is assumed that each limb segment is independent of the others.

The input to the linear system is the 3D coordinates of joints added with white noise, and the output is the combined 3D coordinated optimally filtered. The most crucial parameter in this model is the time constant, which determines how fast a limb segment joints can move in typical human motion conditions. The state vector is 3-D, composed of the x, y, and z coordinates of the joint:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{4.1}$$

The state equations of the discrete-time model can be seen at eq.4.2 and eq.4.3.

$$x_{k+1} = \Phi_k . x_K + \omega_k \tag{4.2}$$

$$\Phi_k = \begin{bmatrix} e^{-\frac{\delta}{\tau_1}} & 0 & 0 \\ 0 & e^{-\frac{\delta}{\tau_2}} & 0 \\ 0 & 0 & e^{-\frac{\delta}{\tau_3}} \end{bmatrix} \tag{4.3}$$

Where δ = 0.04 and all the τ values were set to 1.2566(values attained experimentally).

The measurement data of the kalman filter is a 12 × 1 vector, as showed below.

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \\ Z_7 \\ Z_8 \\ Z_9 \\ Z_{10} \\ Z_{11} \\ Z_{12} \end{bmatrix} = \begin{bmatrix} \text{x component of joint view from camera 1} \\ \text{y component of joint view from camera 1} \\ \text{z component of joint view from camera 1} \\ \text{x component of joint view from camera 2} \\ \text{y component of joint view from camera 2} \\ \text{z component of joint view from camera 2} \\ \text{x component of joint view from camera 3} \\ \text{y component of joint view from camera 3} \\ \text{z component of joint view from camera 3} \\ \text{x component of joint view from camera 4} \\ \text{y component of joint view from camera 4} \\ \text{z component of joint view from camera 4} \end{bmatrix} \tag{4.4}$$

The corresponding discrete measurement equation is given by eq.4.5. The matrix H_x is the 12 × 3 matrix that maps each element of the 12 × 1 measurements vector with the states vector with size 3 × 1.

$$Z_k = H_k . x_k + v_k \tag{4.5}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

The process noise covariance matrix Q_k is defined by

$$Q_k = \begin{bmatrix} \frac{D_1}{2\tau_1}(1 - e^{-2\frac{\delta}{\tau_1}}) & 0 & 0 \\ 0 & \frac{D_2}{2\tau_2}(1 - e^{-2\frac{\delta}{\tau_2}}) & 0 \\ 0 & 0 & \frac{D_3}{2\tau_3}(1 - e^{-2\frac{\delta}{\tau_3}}) \end{bmatrix} \tag{4.7}$$

$$D_1 = 0.7817$$
$$D_2 = 0.7817$$
$$D_3 = 0.7817$$

The measurement noise covariance matrix R_k represents the level of confidence placed in the accuracy of the measurements and is given by:

$$R_k = E[v_k v_k^T] \tag{4.8}$$

Where R_k is a (12 × 12) diagonal matrix, and each non-zero element is given by the equation 4.9

$$diag\_element(a) = 0.42 + 10.7 \quad (1 - acc(a)) \tag{4.9}$$

acc(a) is the accuracy value output by the OpenPose algorithm, associated with its respective measurement a. The constant values in this equation were obtain experimentally.

4.3 Encoding skeleton data

4.3.1 Preprocessing skeleton data

The preprocessing stage comprises two sequential steps: Data normalization and sliding window creation. Some datasets with an insu□cient number of action sequences per action require additional obtained by data augmentation. This data augmentation process consisted in rotating the original sequences on the z-axis 30 degrees ten times ($[0, 360[$ degrees).

Data is normalized by referencing all the joints in the sequence to a specific neck joint position of a specif frame. The neck joint position chosen to normalize was the one on the first frame of the sequence.

A constant non-overlapping sliding window size of 15 frames of a single skeleton joint (e.g15x15x60, a matrix size 15x15 composed of poses of 3 skeletons with 20 joints each) is employed to encode the temporal dimension. We try to have a good performance in our dataset with the least amount of frames per sliding window and experimentally arrive at the value of 15 frames.

4.3.2 Encoding skeleton data

The previously mentioned sliding window contains encoded data of skeleton joints throughout a sequence of frames. Two types of skeleton data encoding were implemented separately to evaluate how di□erent types of encoding a□ect our system performance and, consequently, decide which one best suits our system.

Gramian Angular Fields

We used the previously normalized joints times-series to implement the gramian angular fields that composed a single skeleton during 15 frames of data skeleton data $X = x_1, .., x_n$ and scale it onto $[-1, 1]$ with a Min-Max scaler. Initially, computing a Gram matrix involves extracting 15 consecutive frames from the scale time-series. The 15 frames extracted are composed of 3D joint positions, but only one of the three position components is selected to create a vector size of 15×1. The newly created vector is then converted into "polar coordinates" according to the equation 3.15. The radius coordinate was discarded to compute the inner product needed to create the Gram matrix, including the radius coordinate would adjust the position for the time dependency, which would be biased in favor of the most

recent one. Next, the vector in polar coordinates with size 15 × 1 becomes the first row and the first column of a matrix with size 15 × 15. The rest of the elements in the matrix are given by interception between the first row and the first column. This interception is done by the inner product operation, according to equation 4.10.

$$Gram_{i,j} = cos(x\_scaled_i + x\_scaled_j) \qquad , i = 1, .., 15, j = 1, .., 15 \tag{4.10}$$

Thus, achieving the encoded 15 × 15 sliding window of single joint component over 15 frames.

Recurrence plot

To create a sliding window encoded through an unthresholded recurrence plot. Procedures similar to the ones used to encode time series in a sliding window through GAF were used. The time-series composed of 3D joint coordinates and normalized to the neck joint are scaled into [-1, 1] with a Min-Max scaler. Then, 15 consecutive frames of one component of the three that form the 3D position are selected to create a vector of size 15. Next, the vector of size 15 × 1 is used as the first column and as the first row of a new 15 × 15 matrix. Lastly, the rest of the matrix elements are attained by calculating the pairwise euclidean distance between the values first row of the matrix with its first fist column, following the equation 3.18. The final matrix is the 15 by 15 sliding window that encodes the joint's time series into an image through the recurrence plot representation.

15 × 1 15 × 15 3.18.

4.4 Classification

The classifier chosen to solve our human dynamic gestures recognition problem was a CNN. Deep learning approaches tend to perform much better than hand craft-method. Furthermore, according to [113], within deep-learning, CNN tends to be more e□cient than RNN, and CNNs do not have the vanishing gradients problem that RNN typically have. Most The proposed CNN architecture, inspired by the work done in [127], can be seen in Fig.4.1. The CNN was implemented in python using Pytorch, an open-source machine learning framework.

The network, as depicted in Fig.4.1 is composed of three convolutional layers and three fully-connected layers. Each layer is followed by a Rectified Linear Unit (ReLU). ReLU is typically chosen in classification problems due to the network's increased performance

during training the network. The network is trained with a cross-entropy loss function and uses the Adam optimizer [128] with a learning rate of 0.0001. Between each convolutional layer and fully-connected layer, a dropout of 25% is applied. A kernel of 5 × 5 is used in each convolutional layer with 1 stride and 2 padding. Data are normalized between each convolutional layer, and max-pooling of 2 × 2 with two strides is applied.

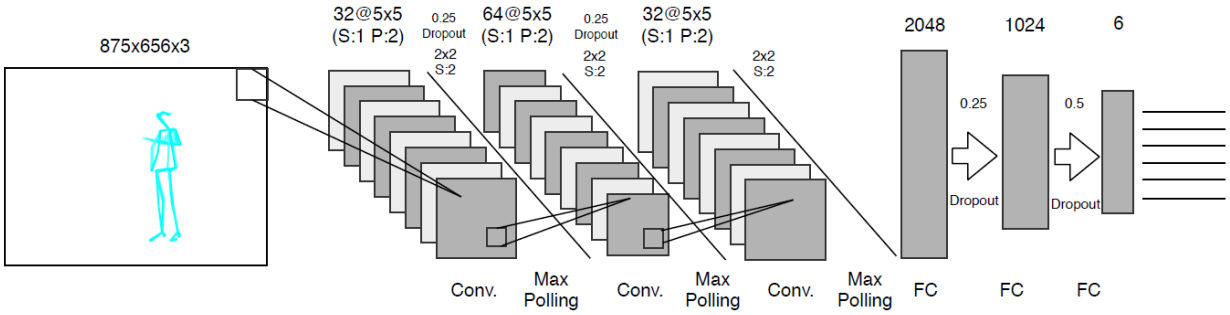

Figure 4.1: The proposed CNN architecture for HAR

5 Results and Discussion

In this chapter we present the results of the performance evaluation of the data fusion component , as well as the evaluation of the performance of HAR algorithm of our system including its encoding components. Both results are exhibited in this chapter along with their respective analysis. To study the combined performance of the HAR algorithm and encoding components several datasets were used, a private previous mentioned on chapter 3, in section 5.1 and four public, this analyse starts in the section 5.3.

5.1 Dataset

We evaluated our proposed method using our dataset developed for 3D human activity analysis. This dataset consists of 6 di□erent actions classes captured from 11 distinct human subjects, eight males and three females with an average age of 23 with a standard deviation of 7. In order to obtain a meaningful and randomly distributed dataset, none of these volunteers were aware of the research being conducted nor had been previously involved in similar tasks. Moreover, Only an intuitive and semantic description of how gestures should be performed was provided to the volunteers, encouraging them to naturally express their freedom of movement, allowing for variability in the dataset collection process.

The volunteers were asked to continuously perform each gesture for one minute, resulting in a sequence with 45000 samples of 3D joints. The skeleton data(3D locations of 25 major body joints) inferred from collected RGB frames, and depth maps were attained from Microsoft Kinect v2. The six di□erent action classes are clap, push with left arm, push with the right arm, wave, swipe with left arm, and swipe with the right arm. These actions were captured in 4 di□erent cameras simultaneously, creating four di□erent viewpoints of each sequence.

5.2 Skeleton Fusion

As described in the previous chapter, a Kalman filter was used to fuse the skeleton data from the four views in our own dataset. The performance of the Kalman filter was evaluated by calculating the standard deviation from skeleton data of each action sequence and performed by each subject. The final standard deviation values are present in table 5.1, consisting of standard deviation values averaged over the 6 actions classes and 11 subjects.

Evaluating our Kalman filter's performance is challenging due to the lack of ground truth in our skeleton data. Visually the skeleton data filtered by the Kalman filter is less noisy than the skeleton data attained from only one camera. Nonetheless, we quantify our Kalman filter performance. That was possible because we only applied the Kalman filter to our dataset. We notice that some joints ideally were supposed to be static during all the action sequences in our dataset, so the lower the variation, the more realistic the prediction was. We selected four joints that ideally should the static and computed their standard deviation. The four joints selected were the neck, right hip, right knee, left hip joints represented in table 5.1 as the numbers 2,10,11,13 respectively.

As theoretically expected, the Kalman filter optimally combined the four skeletons, outputting a skeleton whose variance was relatively lower in all the four "static" joint components when compared to any of the initial unfiltered skeleton data as depicted in table 5.1. (The concept of "static" joints is explained in the paragraph above)

5.3 Gesture Recognition

To test and analyse the performance of our HAR algorithm, several dataset were use to train the CNN

To evaluate the performance of our HAR system each subject in a given dataset is grouped into either the training group or the system's validation group. The criteria for placing the subjects through the groups changes based on the number of subject in that dataset and on the type of training-validation approach usually applied to test that dataset, which means that each dataset will its own unique distribution of subjects through the two groups. This segregation of subject in two groups, will divide the images containing the action sequences sliced in time windows in this two categories. The way accuracies are obtain from this images does not change, no matter the dataset, after having our CNN trained with the images from the training group , the images from the validation group are feed into our system which

Table 5.1: Standard deviation of skeleton data before and after being filtered by kalman filter

Method	Joint	σ_x (mm)	σ_y (mm)	σ_z (mm)
Raw View 1	9	81.7	78.8	218.1
Raw View 2	9	37.4	39.7	209.6
Raw View 3	9	60.4	62.4	216.8
Raw View 4	9	65.2	63.6	225.7
Kalman	**9**	**47.8**	**39.7**	**204.5**
Raw View 1	10	85.7	83.7	219.2
Raw View 2	10	41.2	42.9	209.9
Raw View 3	10	60.4	63.0	216.6
Raw View 4	10	66.5	64.8	225.9
Kalman	**10**	**47.9**	**39.7**	**204.5**
Raw View 1	13	83.8	80.9	219.1
Raw View 2	13	38.0	40.6	210.1
Raw View 3	13	64.6	67.7	217.9
Raw View 4	13	66.0	64.4	226.3
Kalman	**13**	**47.8**	**39.8**	**205.1**
Raw View 1	2	83.7	81.3	218.8
Raw View 2	2	37.2	39.5	210.0
Raw View 3	2	66.0	69.7	218.1
Raw View 4	2	65.7	64.0	226.2
Kalman	**2**	**47.8**	**39.6**	**205.4**

outputs the confusion matrix. Then the eq.5.1 is applied.

$$\text{accuracy} = \frac{\text{success cases}}{\text{total number of cases}} \tag{5.1}$$

Inside of eq.5.1, the number of success cases is the sum of the diagonal values of the confusion matrix, and the total number of cases is the sum of all values in the matrix.

Table 5.2: Accuracy (%) comparisons on di□erent datasets of our proposed HAR architecture

Dataset	No. actions classes	Gramian angular fields Accuracy	Recurrence plot Accuracy
Our own dataset	6	46.58	87.78
UTD-MHAD	27	13.62	33.92
UTKinect	10	43.73	64.06

5.3.1 Private Dataset

We tested and validated our own dataset through cross-validation, where 9 subjects were used in training, and 1 subject was use for validation. It was achieved an accuracy of 46.58% and 87.78% for the GAF encoding and recurrence plot encoding, respectively, the results are shown in tab. 5.2. We also performed training and validation by selecting half of the subjects for training, and the other half was used for validation. Initially, we obtained the accuracy values of 36.09% with the GAF and 69.15% with the recurrence plot encoding while selecting the subjects number 6 to 11 to training, leaving the ones from 1 to 5 for validation. Later the inverse was performed using subjects 1 to 5 for training and the rest for validation. We obtained the accuracy values 38.89% with the GAF encoding and 86.19% with the recurrence plot encoding. With all the results obtained in this dataset and shown in tab. 5.2, it is clear that the recurrence plot performs better than GAF in our system.

5.3.2 Public Datasets

UT-Kinect

The UTKinect-Action3D Dataset (UT-Kinect) [59] is a dataset for action recognition from depth sequences, captured using a single stationary Kinect. There are 10 action types: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands. Ten subjects perform each action twice. The sequences were skeleton joint locations of the 20 joints that constitute a body in this dataset. Initially, the framerate is 30f/s, but because the only recorded frames were the ones where the skeleton was tracked, the frame number of the files has jumped, setting the de facto frame rate to about 15f/sec.

The procedure of data augmentation mentioned in the previous chapter was performed with this dataset. As mentioned above, this dataset was first presented in [59] in 2012, with an overall mean accuracy of 90.92 %, more recently in 2018, a Deep Progressive Reinforcement Learning (DPRL) [129] for Skeleton-Based Action Recognition report an accuracy of 98.5%. The training and validation for this dataset were performed through cross-validation, where 9 subjects were used in training, leaving 1 for validation. The average accuracy was 43.73 % with the GAF encoding and 64.06 with the recurrence plot encoding, as displayed in table 5.2 . These results show that the recurrence plot performs far better with this CNN architecture in this dataset. Moreover, our algorithm's performance in this dataset might have been severely penalized for only utilizing 15 frames to create the sliding window.

UTD-MHAD

The UTD multimodal human action dataset (UTD-MHAD) [130] was collected using a Microsoft Kinect sensor and a wearable inertial sensor in an indoor environment. It is a dataset that contains 27 actions performed by 8 subjects (4 females and 4 males). Each subject repeated each action 4 times.

A deep neural network-based multi-modal algorithm called HAMLET [131], published in 2020, retains the best performance in the dataset with an accuracy of 95.12%. We implement the procedure of data augmentation mentioned in the previous chapter in this dataset. The validation for this dataset was performed as described in [130], where the actions performed by subjects 2, 4, 6, 8 were used to validate the results, and subjects 1, 3, 5, 7 were used to train the machine learning algorithm. In order to achieve better performances, another two convolutional layers were added to the CNN, resulting in accuracy values of 43.73% for the GAF encoding and 64.06% for the recurrence plot encoding as shown in table 5.2. The two extra convolutional layers were added in an attempt to increase the accuracy, culminating in a rise of about 8pp(percentual points) in its accuracy. It is also clear from table 5.2that the system when recurrence plot encoding was used outperformed the gramian angular fields. Moreover, our algorithm's performance in this dataset might have been severely penalized for only utilizing 15 frames to create the sliding window.

5.3.3 HAR algorithm

Our HAR algorithm achieved a good accuracy of about 88% in our dataset while using the recurrence plot encoding, as exhibited in tab. 5.2. Moreover, the 15 frames used to create the sliding window in our dataset might have severely penalized the performance on the public dataset. Nevertheless, an inverse correlation is observable between the number of human actions classes that comprise a dataset and the accuracy of the HAR algorithm, which indicates that the current CNN architecture can not deal with the increased complexity of having a more considerable number of classes. Also, by analyzing the results in the table 5.2, it is evident that our system reaches better accuracy when using the recurrence plot to encode data skeleton into images than when it uses the GAF encoding, showing better accuracy in every single dataset.

6 Conclusion

6.1 Work Done

This dissertation aimed to develop a pipeline capable of fusion skeleton data, encoding that fused data into an image using two di□erent approaches, the recurrence plot and the GAF, and feed those images into a CNN to recognize sequences of human gestures. First, we applied the algorithms to our dataset were with fused skeleton data, encoding it into images and, then the same methods for encoding into imagens CNN on two public datasets.

Using the results obtained in this experiment, it was possible to conclude that our system has relatively good accuracy in datasets with few human action classes. Consequently, it starts to lose accuracy when the number of action classes in those datasets increases. We also noticed a superior performance in our system when using recurrence plot encoding over the GAF encoding approach. However, the number of frames used to create the sliding window may not be enough to achieve high accuracy on public datasets. The Kinect sensor has 25 Hz, even though having a sliding window of 15 frames allows us to perform well in our dataset. Creating a window that takes less than a second of the body movement certainly harshly penalized the performance on public datasets, whose action sequences may be slower than on ours.

6.2 Future Work

Considering the work developed and the results obtained, there are several possibilities for improving the current work. The Microsoft Kinect V2 has a relatively noisy ToF technology, which consequently generates noisy depth maps. Therefore to mitigate the noise in these depth maps, improvements in the fusion algorithm can be made. To improve the Kalman accuracy without changing the sensor implies having a physical model of the dynamics of the human body more capable of producing more accurate predictions of the movement of the

body's behavior. Hence, the system's overall uncertainty could be reduced by placing more trust in this improved model in our Kalman filter algorithm. A software suited for modeling, simulating, controlling, and analyzing the mechanics of a neuromusculoskeletal system like a human body is OpenSim [132], a freely available software.

The performance of our HAR algorithm can be highly increased in public datasets by forming a sliding window from more frames of skeleton data. The current 15 frames used to achieve a good performance in our dataset with the lowest number of frames possible indisputably punishes other datasets.

As mentioned previously, our system struggles to cope with added complexity created by increasing the number of human actions classes. Since recurrence plots perform better than GAF, future work could explore the potential of the recurrence plot in creating images to feed to HAR algorithms. However, a more suitable CNN architecture could be designed and tested to solve better the problem of HAR for a large number of action classes.

Bibliography

[1] V. M Zatsiorsky, *Kinematics of human motion*. Human Kinetics, 1998.

[2] G. Johansson, “Visual perception of biological motion and a model for its analysis”, *Perception & Psychophysics*, vol. 14, pp. 201–211, 1973.

[3] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using hidden markov model”, *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. J76-D-II, pp. 379–385, Jul. 1992. DOI: 10.1109/CVPR.1992.223161.

[4] Y. Kuniyoshi, H. Inoue, and M. Inaba, “Design and implementation of a system that generates assembly programs from visual recognition of human action sequences”, 567–574 vol.2, 1990. DOI: 10.1109/IROS.1990.262444.

[5] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey”, *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, pp. 1473–1488, Dec. 2008. DOI: 10.1109/TCSVT.2008.2005594.

[6] J. Aggarwal and M. Ryoo, “Human activity analysis: A review”, *ACM Comput. Surv.*, vol. 43, no. 3, Apr. 2011, ISSN: 0360-0300. DOI: 10.1145/1922649.1922653. [Online]. Available: https://doi.org/10.1145/1922649.1922653.

[7] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, “Sensor-based and vision-based human activity recognition: A comprehensive survey”, *Pattern Recognition*, vol. 108, p. 107 561, 2020, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107561. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320320303642.

[8] G. Batchuluun, J. H. Kim, H. G. Hong, J. K. Kang, and K. R. Park, “Fuzzy system based human behavior recognition by combining behavior prediction and recognition”, *Expert Systems with Applications*, vol. 81, pp. 108–133, 2017, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.03.052. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417302051.

[9] X. Ji, J. Cheng, W. Feng, and D. Tao, “Skeleton embedded motion body partition for human action recognition using depth sequences”, *Signal Processing*, vol. 143, pp. 56–68, 2018, ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2017.08.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168417302980.

[10] A. Jalal, “Robust human activity recognition from depth video using spatiotemporal multi-fused features”, *Pattern Recognition*, vol. 61, pp. 295–308, Jan. 2017.

[11] B. N. Capela NA Lemaire ED, “Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients”, May 2015. DOI: 10.1371/journal.pone.0124414.

[12] “Sensors, vision and networks: From video surveillance to activity recognition and health monitoring”, Jan. 2019. DOI: 10.3233/AIS-180510.

[13] R. Varatharajan and G. Manogaran, “Wearable sensor devices for early detection of alzheimer disease using dynamic time warping algorithm”, *Cluster Computing*, vol. 21, Mar. 2018. DOI: 10.1007/s10586-017-0977-2.

[14] S. Sennan, “Internet of things based ambient assisted living for elderly people health monitoring”, *Research Journal of Pharmacy and Technology*, vol. 11, pp. 1–5, Oct. 2018.

[15] E. Zdravevski, P. Lameski, V. Trajkovik, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo, and N. Garcia, “Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering”, *IEEE Access*, vol. 5, pp. 5262–5280, 2017. DOI: 10.1109/ACCESS.2017.2684913.

[16] P. W. Li H Derrode S, “Lower limb locomotion activity recognition of healthy individuals using semi-markov model and single wearable inertial sensor”, 2019. DOI: 10.3390/s19194242.

[17] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, “A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors”, *IEEE Internet of Things Journal*, vol. 6, pp. 1384–1393, 2019.

[18] G. Plasqui, “Smart approaches for assessing free-living energy expenditure following identification of types of physical activity”, *International Association for the Study of Obesity*, 2017. DOI: 10.1111/obr.12506.

[19] C. Xu, L. N. Govindarajan, and L. Cheng, “Hand action detection from ego-centric depth sequences with error-correcting hough transform”, *Pattern Recognition*, vol. 72, pp. 494–503, 2017, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.08.009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317303114.

[20] O. Oyedotun and A. Khashman, “Deep learning in vision-based static hand gesture recognition”, *Neural Computing and Applications*, vol. 28, Dec. 2017. DOI: 10.1007/s00521-016-2294-8.

[21] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, “Beyond temporal pooling : Recurrence and temporal convolutions for gesture recognition in video”, eng, *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 126, no. 2-4, pp. 430–439, 2018, ISSN: 0920-5691. [Online]. Available: http://dx.doi.org/10.1007/s11263-016-0957-7.

[22] T. Billah, S. Rahman, M. O. Ahmad, and M. Swamy, “Recognizing distractions for assistive driving by tracking body parts”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, pp. 1–1, Mar. 2018. DOI: 10.1109/TCSVT.2018.2818407.

[23] M. M. T. E. Ohn-Bar, “Looking at humans in the age of self-driving and highly automated vehicles”, *IEEE Transactions on Intelligent Vehicles*, 2016.

[24] G. F. Angela Yao Juergen Gall and L. V. Gool, “Does human action recognition benefit from pose estimation?”, in *Proceedings of the British Machine Vision Conference*, http://dx.doi.org/10.5244/C.25.67, BMVA Press, 2011, pp. 67.1–67.11, ISBN: 1-901725-43-X.

[25] L. Lo Presti, M. La Cascia, S. Sclaro□, and O. Camps, “Gesture modeling by hanklet-based hidden markov model”, vol. 9005, Nov. 2014, ISBN: 978-3-319-16810-4. DOI: 10.1007/978-3-319-16811-1_35.

[26] S. Saha, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[27] A. Hostetter and M. Alibali, "Visible embodiment: Gestures as simulated action", *Psychonomic bulletin & review*, vol. 15, pp. 495–514, Jul. 2008. DOI: 10.3758/PBR.15.3.495.

[28] M. A. Novack and S. Goldin-Meadow, "Gesture as representational action: A paper about function", *Psychonomic Bulletin & Review*, vol. 24, pp. 652–665, 2017.

[29] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future", *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3178–3185, 2012.

[30] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "Using k-poselets for detecting people and localizing their keypoints", in *CVPR*, 2014.

[31] M. Sun and S. Savarese, "Articulated part-based model for joint object detection and pose estimation", in *2011 International Conference on Computer Vision*, 2011, pp. 723–730. DOI: 10.1109/ICCV.2011.6126309.

[32] U. Iqbal and J. Gall, "Multi-person pose estimation with local joint-to-person associations", in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds., Cham: Springer International Publishing, 2016, pp. 627–642, ISBN: 978-3-319-48881-3.

[33] G. T. Z. N. K. A. T. J. T. C. Bregler and K. Murphy, "Towards accurate multi-person pose estimation in the wild".

[34] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, 2006, pp. 26–36. DOI: 10.1109/CVPR.2006.202.

[35] Viola, Jones, and Snow, "Detecting pedestrians using patterns of motion and appearance", in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, 734–741 vol.2. DOI: 10.1109/ICCV.2003.1238422.

[36] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition", *International Journal of Computer Vision*, vol. 61, pp. 55–97, 2005, ISSN: 1573-1405. DOI: https://doi.org/10.1023/B:VISI.0000042934.15159.49.

[37] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3d pose estimation and tracking by detection”, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 623–630, 2010.

[38] ——, “Pictorial structures revisited: People detection and articulated pose estimation”, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1014–1021, 2009.

[39] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, “Poselet conditioned pictorial structures”, Jun. 2013.

[40] Y. Yang and D. Ramanan, “Articulated human detection with flexible mixtures of parts”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013. DOI: 10.1109/TPAMI.2012.261.

[41] S. Johnson and M. Everingham, “Clustered pose and nonlinear appearance models for human pose estimation”, in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2010, pp. 12.1–12.11, ISBN: 1-901725-40-5. DOI: 10.5244/C.24.12.

[42] D. Ramanan, D. Forsyth, and A. Zisserman, “Strike a pose: Tracking people by finding stylized poses”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, 271–278 vol. 1. DOI: 10.1109/CVPR.2005.335.

[43] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines”, Jan. 2016.

[44] A. Bulat and G. Tzimiropoulos, “Human pose estimation via convolutional part heatmap regression”, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, 2016, pp. 717–732, ISBN: 978-3-319-46478-7.

[45] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, “Pose machines: Articulated pose estimation via inference machines”, in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 33–47, ISBN: 978-3-319-10605-2.

[46] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition”, vol. 61, 2005, pp. 55–79. DOI: 10.1023/B:VISI.0000042934.15159.49. [Online]. Available: https://doi.org/10.1023/B:VISI.0000042934.15159.49.

[47] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Progressive search space reduction for human pose estimation”, English, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, United States: Institute of Electrical and Electronics Engineers (IEEE), 2008, pp. 1–8, ISBN: 978-1-4244-2242-5. DOI: 10.1109/CVPR.2008.4587468.

[48] D. Ramanan, “Learning to parse images of articulated bodies”, in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS’06, Canada: MIT Press, 2006, pp. 1129–1136.

[49] Z. C. T. S. S. E. Wei and Y. Sheikh, “Realtime multi-person 2d pose estimation using part a□nity fields”, 2017, arXiv:1611.08050.

[50] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images”, in *CVPR 2011*, 2011, pp. 1297–1304. DOI: 10.1109/CVPR.2011.5995316.

[51] L. Xia, C.-C. Chen, and J. Aggarwal, “Human detection using depth information by kinect”, *CVPR 2011 WORKSHOPS*, pp. 15–22, 2011.

[52] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group”, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588–595. DOI: 10.1109/CVPR.2014.82.

[53] D. Wu and L. Shao, “Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition”, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 724–731. DOI: 10.1109/CVPR.2014.98.

[54] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, “Bio-inspired dynamic 3d discriminative skeletal features for human action recognition”, in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 471–478. DOI: 10.1109/CVPRW.2013.153.

[55] C. Wang, Y. Wang, and A. Yuille, “An approach to pose-based action recognition”, Jun. 2013, pp. 915–922. DOI: 10.1109/CVPR.2013.123.

[56] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu, “Concurrent action detection with structural prediction”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013.

[57] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Sequence of the most informative joints (smij): A new representation for human skeletal action recognition”, in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 8–13. DOI: 10.1109/CVPRW.2012.6239231.

[58] M. Hussein, M. Torki, M. Gowayyed, and M. El Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations”, Aug. 2013.

[59] L. Xia, C.-C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3d joints”, in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 20–27. DOI: 10.1109/CVPRW.2012.6239233.

[60] A. Eweiwi, M. S. Cheema, C. Bauckhage, and J. Gall, “E□cient pose-based action recognition”, vol. 9007, Nov. 2014, ISBN: 978-3-319-16813-5. DOI: 10.1007/978-3-319-16814-2_28.

[61] T. Kerola, N. Inoue, and K. Shinoda, “Spectral graph skeletons for 3d action recognition”, Nov. 2014, pp. 417–432, ISBN: 978-3-319-16816-6. DOI: 10.1007/978-3-319-16817-3_27.

[62] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories”, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, 2006, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.

[63] A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta, “Fusion of skeletal and silhouette-based features for human action recognition with rgb-d devices”, in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 91–97. DOI: 10.1109/ICCVW.2013.19.

[64] I. C. Duta, J. Uijlings, B. Ionescu, K. Aizawa, A. Hauptmann, and N. Sebe, “E□cient human action recognition using histograms of motion gradients and vlad with descriptor shape information”, *Multimedia Tools and Applications*, vol. 76, pp. 22 445–22 472, 2017.

[65] J. Uijlings, I. Duta, E. Sangineto, and N. Sebe, “Video classification with densely extracted hog/hof/mbh features: An evaluation of the accuracy/computational video classification with densely extracted hog/hof/mbh features: An evaluation of the accuracy/computational e□ciency trade-o□”, vol. 4, Mar. 2015.

[66] F. Jurie and B. Triggs, “Creating e□cient codebooks for visual recognition”, in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, vol. 1, 2005, 604–610 Vol. 1. DOI: 10.1109/ICCV.2005.66.

[67] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition.”, Sep. 2009. DOI: 10.5244/C.23.124.

[68] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, “Dense trajectories and motion boundary descriptors for action recognition”, *International Journal of Computer Vision*, vol. 103, May 2013. DOI: 10.1007/s11263-012-0594-8.

[69] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[70] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance”, in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 428–441, ISBN: 978-3-540-33835-2.

[71] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies”, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587756.

[72] L. Sun, K. Jia, D.-Y. Yeung, and B. Shi, “Human action recognition using factorized spatio-temporal convolutional networks (fstcn)”, Dec. 2015. DOI: 10.1109/ICCV.2015.522.

[73] X. Wang, A. Farhadi, and A. Gupta, “Actions ~ transformations”, in *CVPR*, 2016.

[74] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks”, *Neural Information Processing Systems*, vol. 25, Jan. 2012. DOI: 10.1145/3065386.

[75] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, “Learning human pose estimation features with convolutional networks”, *CoRR*, vol. abs/1312.7302, 2014.

[76] S. Li, W. Zhang, and A. Chan, “Maximum-margin structured learning with deep networks for 3d human pose estimation”, Aug. 2015.

[77] C. Liang, Y. Song, and Y. Zhang, “Hand gesture recognition using view projection from point cloud”, *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 4413–4417, 2016.

[78] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos”, in *NIPS*, 2014.

[79] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors”, May 2015. DOI: 10.1109/CVPR.2015.7299059.

[80] G. Gkioxari and J. Malik, “Finding action tubes”, Nov. 2014.

[81] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “Learning to track for spatio-temporal action localization”, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3164–3172, 2015.

[82] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, “Sequential deep learning for human action recognition”, Nov. 2011.

[83] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013. DOI: 10.1109/TPAMI.2012.59.

[84] Z. Liu, C. Zhang, and Y. Tian, “3d-based deep convolutional neural network for action recognition with depth sequences”, *Image and Vision Computing*, vol. 55, pp. 93–100, 2016, Handcrafted vs. Learned Representations for Human Action Recognition, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2016.04.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885616300592.

[85] J. L. Elman, “Finding structure in time”, *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E.

[86] F. Gers, N. Schraudolph, and J. Schmidhuber, “Learning precise timing with lstm recurrent networks”, *Journal of Machine Learning Research*, vol. 3, pp. 115–143, Jan. 2002. DOI: 10.1162/153244303768966139.

[87] L. Pigou, A. van den Oord, S. Dieleman, M. V. Herreweghe, and J. Dambre, “Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video”, *International Journal of Computer Vision*, vol. 126, pp. 430–439, 2016.

[88] Y. Du, W. Wang, Wang, and Liang, “Hierarchical recurrent neural network for skeleton based action recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015.

[89] V. Veeriah, N. Zhuang, and G.-J. Qi, “Di erential recurrent neural networks for action recognition”, Dec. 2015, pp. 4041–4049. DOI: 10.1109/ICCV.2015.460.

[90] D. Wu, L. Pigou, P.-J. Kindermans, N. LE, L. Shao, J. Dambre, and J.-M. Odobez, “Deep dynamic neural networks for multimodal gesture segmentation and recognition”, eng, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 38, no. 8, pp. 1583–1597, 2016, ISSN: 0162-8828. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2016.2537340.

[91] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, “Hough forests for object detection, tracking, and action recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011. DOI: 10.1109/TPAMI.2011.70.

[92] L. Lo Presti and M. La Cascia, “3d skeleton-based human action classification: A survey”, *Pattern Recognition*, vol. 53, pp. 130–147, 2016, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2015.11.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320315004392.

[93] M. Barker and W. Rayens, “Partial least squares for discrimination, journal of chemometrics”, *Journal of Chemometrics*, vol. 17, pp. 166–173, Mar. 2003. DOI: 10.1002/cem.785.

[94] R. Rosipal and L. Trejo, “Kernel partial least squares regression in reproducing kernel hilbert space”, *Journal of Machine Learning Research*, vol. 2, pp. 97–123, Dec. 2001. DOI: 10.1162/15324430260185556.

[95] P.-L. J. F.-R. F. Climent-Pérez P. Chaaraoui A.A., “Optimal joint selection for skeletal data from rgb-d devices using a genetic algorithm”, vol. 7630, 2013. DOI: 10.1007/978-3-642-37798-3_15.

[96] B. S. D. B. A. P. P. Seidenari L. Varano V., “Weakly aligned multi-part bag-of-poses for action recognition from depth cameras”, vol. 8158, 2013. DOI: 10.1007/978-3-642-41190-8_48.

[97] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, “Accurate 3d action recognition using learning on the grassmann manifold”, *Pattern Recognition*, vol. 48, Aug. 2014. DOI: 10.1016/j.patcog.2014.08.011.

[98] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm”, in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML ’04, Ban□, Alberta, Canada: Association for Computing Machinery, 2004, p. 6, ISBN: 1581138385. DOI: 10.1145/1015330.1015424. [Online]. Available: https://doi.org/10.1145/1015330.1015424.

[99] L. Lo Presti, M. La Cascia, S. Sclaro□, and O. Camps, “Hankelet-based dynamical systems modeling for 3d action recognition”, *Image and Vision Computing*, Oct. 2015. DOI: 10.1016/j.imavis.2015.09.007.

[100] B. Li, O. I. Camps, and M. Sznaier, “Cross-view activity recognition using hankelets”, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1362–1369. DOI: 10.1109/CVPR.2012.6247822.

[101] B. Li, M. Ayazoglu, T. Mao, O. I. Camps, and M. Sznaier, “Activity recognition using dynamic subspace angles”, in *CVPR 2011*, 2011, pp. 3193–3200. DOI: 10.1109/CVPR.2011.5995672.

[102] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, and R. Sukthankar, “Exploring the trade-o□ between accuracy and observational latency in action recognition”, *International Journal of Computer Vision*, vol. 101, pp. 420–436, 2013. DOI: 10.1007/s11263-012-0550-7.

[103] X. Yang and Y. Tian, “Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor”, *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 14–19, 2012.

[104] B. Ren, M. Liu, R. Ding, and H. Liu, *A survey on 3d skeleton-based action recognition using learning method*, 2020. arXiv: 2002.05907 [cs.CV].

[105] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, *Adding attentiveness to the neurons in recurrent neural networks*, 2018. arXiv: 1807.04445 [cs.CV].

[106] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, *View adaptive neural networks for high performance skeleton-based human action recognition*, Apr. 2018. DOI: 10.1109/TPAMI.2019.2896631.

[107] E. P. Ijjina and C. K. Mohan, “Human action recognition based on mocap information using convolution neural networks”, in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 159–164. DOI: 10.1109/ICMLA.2014.30.

[108] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, *Actional-structural graph convolutional networks for skeleton-based action recognition*, 2019. arXiv: 1904.12659 [cs.CV].

[109] L. Shi, Y. Zhang, J. Cheng, and H. Lu, *Two-stream adaptive graph convolutional networks for skeleton-based action recognition*, 2019. arXiv: 1805.07694 [cs.CV].

[110] D. Liang, G. Fan, L. Guangfeng, W. Chen, X. Pan, and H. Zhu, “Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition”, Jun. 2019, pp. 934–940. DOI: 10.1109/CVPRW.2019.00123.

[111] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, “Richly activated graph convolutional network for robust skeleton-based action recognition”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1915–1925, May 2021, ISSN: 1558-2205. DOI: 10.1109/tcsvt.2020.3015051. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2020.3015051.

[112] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, *Semantics-guided neural networks for e cient skeleton-based human action recognition*, 2020. arXiv: 1904.01189 [cs.CV].

[113] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View adaptive neural networks for high performance skeleton-based human action recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[114] M. Müller, T. Röder, and M. Clausen, “E□cient content-based retrieval of motion capture data”, in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH ’05, Los Angeles, California: Association for Computing Machinery, 2005, pp. 677–685, ISBN: 9781450378253. DOI: 10.1145/1186822.1073247. [Online]. Available: https://doi.org/10.1145/1186822.1073247.

[115] G. Evangelidis, G. Singh, and R. Horaud, “Skeletal Quads: Human Action Recognition Using Joint Quadruples”, in *International Conference on Pattern Recognition*, Stockholm, Sweden: IEEE, Aug. 2014, pp. 4513–4518. DOI: 10.1109/ICPR.2014.772. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00989725.

[116] B. S. Devanne M. Wannous H., P. P., D. M., and D. B. A., “Space-time pose representation for 3d human action recognition”, *ICIAP 2013. Lecture Notes in Computer Science*, vol. 8158, pp. 14–19, 2013. DOI: 10.1007/978-3-642-41190-8_49.

[117] I. Lillo, A. Soto, and J. C. Niebles, “Discriminative hierarchical modeling of spatio-temporally composable human activities”, Jun. 2014. DOI: 10.1109/CVPR.2014.109.

[118] L. Miranda, T. Vieira, D. M. Morera, T. Lewiner, A. W. Vieira, and M. Campos, “Online gesture recognition from pose kernel learning and decision forests”, *Pattern Recognit. Lett.*, vol. 39, pp. 65–73, 2014.

[119] H. B. Youngjoo Kim, *Introduction to Kalman Filter and Its Applications*. Nov. 2018. DOI: 10.5772/intechopen.80600.

[120] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552. [Online]. Available: https://doi.org/10.1115/1.3662552.

[121] S. M. Bozic, *Digital And Kalman Filtering*. Courier Publishing Languages, 1994.

[122] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation”, in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI’15, Buenos Aires, Argentina: AAAI Press, 2015, pp. 3939–3945, ISBN: 9781577357384.

[123] M. Thiel, M. C. Romano, and J. Kurths, “How much information is contained in a recurrence plot?”, *Physics Letters A*, vol. 330, no. 5, pp. 343–349, 2004, ISSN: 0375-9601. DOI: https://doi.org/10.1016/j.physleta.2004.07.050. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0375960104009922.

[124] Y. Chen and H. Yang, “Multiscale recurrence analysis of long-term nonlinear and nonstationary time series”, *Chaos, Solitons & Fractals*, vol. 45, no. 7, pp. 978–987, 2012, ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2012.03.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077912000860.

[125] E. Bachmann, I. Duman, U. Usta, R. McGhee, X. Yun, and M. Zyda, “Orientation tracking for humans and robots using inertial sensors”, in *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA’99 (Cat. No.99EX375)*, 1999, pp. 187–194. DOI: 10.1109/CIRA.1999.810047.

[126] J. L. Marins, X. Yun, E. Bachmann, R. McGhee, and M. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors", *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 4, 2003–2011 vol.4, 2001.

[127] J. R. Paulo, L. Garrote, P. Peixoto, and U. J. Nunes, "Spatiotemporal 2d skeleton-based image for dynamic gesture recognition using convolutional neural networks", in *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 2021, pp. 1138–1144. DOI: 10.1109/RO-MAN50785.2021.9515418.

[128] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[129] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5323–5332. DOI: 10.1109/CVPR.2018.00558.

[130] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor", in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 168–172. DOI: 10.1109/ICIP.2015.7350781.

[131] M. M. Islam and T. Iqbal, "Hamlet: A hierarchical multimodal attention-based human activity recognition algorithm", *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 285–10 292, 2020.

[132] S. Delp, F. Anderson, A. Arnold, P. Loan, A. Habib, C. John, E. Guendelman, and D. Thelen, "Opensim: Open-source software to create and analyze dynamic simulations of movement", *Biomedical Engineering, IEEE Transactions on*, vol. 54, pp. 1940–1950, Dec. 2007. DOI: 10.1109/TBME.2007.901024.