FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Human Action and Facial Expressions Recognition in a VR game

Júlio Castro Lopes

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

International Master in Computer Vision

Supervisor: Luís Teixeira (PhD)

Supervisor: Rui Pedro Lopes (PhD)

November 9, 2022

© Júlio Castro Lopes, 2022

Resumo

Hoje em dia, a tecnologia de jogos está muito avançada e pode ser usada de várias maneiras, incluindo entretenimento, educação e até reabilitação. Os jogos de Realidade Virtual (VR) podem ser uma forma importante de realizar reabilitação, porque requerem muito mais envolvimento do utilizador, interagindo com movimentos e com emoções dos jogadores. Este trabalho apresenta o desenvolvimento de vários métodos para duas tarefas de visão por computador distintas, Reconhecimento de Ações Humanas (HAR) e Reconhecimento da Expressão Facial (FER), integradas como parâmetros de decisão de um módulo de Adaptação Dinâmica de Dificuldade (DDA) no futuro, ajustando assim dinamicamente a dificuldade do jogo com base no desempenho do jogador. Para executar tarefas de HAR, o trabalho descrito nesta dissertação recorreu a dois algoritmos distintos, OpenPose (2D) e BlazePose (3D), para extrair o esqueleto humano e depois normalizar e explorar esta informação, de forma a classificar que ação um humano está a realizar. A pontuação de F1 mais alta 0.745, utilizando o conjunto de dados N-UCLA, foi conseguida com o algoritmo de extração do esqueleto OpenPose, seguida da computação dos ângulos entre as articulações mais próximas.

No que diz respeito ao FER, este trabalho apresenta também o desenvolvimento de um módulo capaz de detetar expressões faciais, enquanto o utilizador está a usar óculos VR, escondendo assim, parte do rosto. Para executar esta tarefa, três Redes Neuronais Convolutionais (CNNs), uma ResNet-18, um VGG19 e a combinação de ambas, foram usadas. Concluiu-se que o melhor modelo de classificação foi a combinação de ambas as redes, que obteve uma eficácia de classificação de 0.649.

Abstract

Nowadays, games are very advanced being one of the top and fresh technological themes and can be used in a variety of ways, including entertainment, education, and even rehabilitation. Virtual Reality (VR) games can be an important way to perform rehabilitation because they require much more involvement from the user, interacting with movements and player's emotions. This work presents the development of several methods for two distinct computer vision tasks, Human Action Recognition (HAR) and Facial Expression Recognition (FER), integrated as decision parameters of a Dynamic Difficulty Adaptation (DDA) module in the future, dynamically adjusting the difficulty of the game, based on how the player performs. To perform HAR tasks, the work described in this dissertation, followed the usage of two distinct algorithms, OpenPose (2D) and BlazePose (3D), to extract the human skeleton and then normalize and explore this information, in order to classify which action a human is performing. The highest F1 score 0.745, using the N-UCLA dataset, was achieved using OpenPose skeleton extraction, followed by the computation of the angles between the closest joints.

Regarding FER, this work presents also the development of a module capable of detecting facial expressions, while the user is wearing VR glasses, occluding like this, part of the face. To perform this task, three Convolutional Neural Networks (CNNs), a ResNet-18, a VGG19 and the combination of both, were used. It was concluded that the best classifying model was the combination of both networks, which achieved an accuracy of 0.649.

Acknowledgements

I would like to thank to my supervisor, Prof. Luís Teixeira, for the availability and suggestions, that were a crucial part of the development of this work.

A special thank to my Co-Supervisor, Prof. Rui Lopes, that is also the coordinator of my current scholarship. The work described here would not have been possible without Prof. Rui's availability, assistance, guidance, and flexibility throughout the year. Thanks also for letting me follow this dual career between school and sports. During the development of this dissertation, I was also able, for the first time, to integrate the national team of the sport that I practice (canoeing), which fulfilled me and also motivated me even more, for this dissertation.

Finally, I could not forget about the support that my girlfriend and family gave to me, during this period. To reconcile both careers, it is really necessary to have the help and comprehension of several people that are around us.

Thank you for everything.

Júlio Castro Lopes

"Excellence is an art won by training and habituation. We do not act rightly because we have virtue or excellence, but we rather have those because we have acted rightly. We are what we repeatedly do. Excellence, then, is not an act but a habit."

Aristotle

Contents

List of Figures

List of Tables

Abbreviations

AR Augmented Reality. 9

BPPEM Binary Proximity Patches Ensemble Motion. 10

CLNF Conditional Local Neural Fields. 12, 13

CNN Convolutional Neural Network. 6–8, 11–14, 16, 18, 26, 34, 43

DAE Deep Autoencoder. 6

DDA Dynamic Difficulty Adjustment. vii, 1–3, 5, 43

DNN Deep Neural Network. 7, 8, 25

EEG Electroencephalography. 1, 5, 13

FER Facial Expression Recognition. 2–5, 11, 30, 34, 40, 43

GCNs Graph Convolutional Networks. 36, 37

HAR Human Action Recognition. vii, ix, 2–10, 18, 19, 25, 30, 33, 43

HOG Histogram of Oriented Gradients. 16, 17

KNN K-Nearest Neighbour. 7, 15, 17

LBP Local Binary Patterns. 16, 17

LGBHPS Local Gabor Binary Pattern Histogram Sequence. 15

LSTM Long Short-Term Memory. 6, 7, 13, 25, 31, 33, 35, 37

MHI Motion History Image. 9

MJD Moving Joints Descriptor. 8, 18, 20, 21, 23, 24, 33, 35, 36, 43

ML Machine Learning. 13

MTCNN Multi-task Cascade Convolutional Neural Networks. 27, 43

PCA Principal Component Analysis. 9, 17

PDM Point Distribution Model. 12, 13

RL Reinforcement Learning. 3, 5

RNN Recurrent Neural Network. 7, 25, 33

SOM Self-Organizing Mapping network. 10

SVM Suport Vector Machine. 6, 7, 9, 10, 13, 14, 16, 17

SVR Support Vector Regression. 13

VR Virtual Reality. vii, 1–6, 9, 10, 16, 17, 27, 30, 40, 44

Chapter 1

Introduction

1.1 Background

Games, whether in video or VR format, are usually accomplished through a series of levels of increasing difficulty. When playing games either offline or online, the intuitive approach is to divide them into three difficulty levels [85], commonly labeled as: easy, medium, and hard [113]. This division in three levels can be seen in several applications, such as educational games [98], virtual reality games [35], video games for entertainment purposes [121], and even on gamification systems for educational purposes [73]. Although this division into three levels can be seen as the standard approach, in some cases it can be extended to accommodate more levels or finer degrees of difficulty. The authors in [88] concluded that their game should have more than three levels of difficulty because the player would frequently get stuck on either the lower or higher difficulty.

Many games allow the user to select the desired level of difficulty at the beginning of the game, that will be kept until the player decides to change it. When the user is playing a game for the first time, deciding on his/her own can be difficult because they have no idea about the difficulty of each level and do not know the relative level of their skills [85]. Furthermore, requiring players to select difficulty levels on a regular basis may be inconvenient and cause game interruptions. When people engage in an activity that is neither boring nor frustrating, they become engaged in it, allowing them to work for longer periods of time and remain focused on the task. The flow channel (Figure 1.1) is a state of mind whose ideas can be applied to a video game [27]. Following the flow, the idea of using adaptive difficulty mechanisms is to keep the game's difficulty within a certain range to prevent the player from getting bored if the game is too easy, or frustrated if the game is too difficult. In recent years, there has been a lot of research done on Dynamic Difficulty Adjustment (DDA) applied to games [144].

DDA aims to adapt the difficulty of the game based on the players' performance, and can be accomplished by considering not only their in-game performance, but also the associated physiological responses. By measuring and integrating physiological data it is possible to map how difficult or easy it is for the player to perform a given task [67]. The study in [115], demonstrated that DDA based on Electroencephalography (EEG) signals improved the players' game

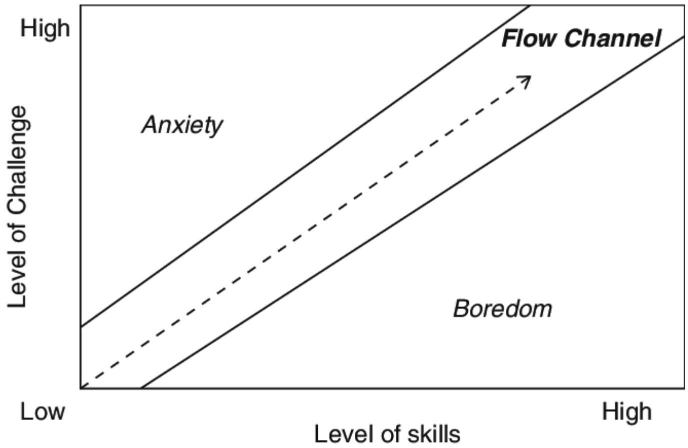


Figure 1.1: Flow channel [41].

experience.

VR games may necessitate more information than the actual gameplay; one method of obtaining information about how a player is performing in the game is to record the gameplay with a third-party camera. This can be integrated in real time in a VR game to provide a clear and grateful gameplay experience. Although substantial research has already been done to construct effective HAR systems from RGB videos, only a few have dedicated its application in a VR [33]. HAR is a constantly evolving topic that has received a lot of attention from researchers due to its importance in a variety of real-world applications, with recent technological advancements allowing for novel applications [10]. It is critical in unknown environments or situations to automatically detect what a person is doing. Along with the respective algorithm to detect these actions, data acquisition is also a critical step, because it is sometimes impossible to use whichever algorithm, even by pre-processing of the visual data, due to poor acquisition conditions.

Since HAR algorithms can be insufficient to determine if a person is performing well on the game, the inclusion of the classification of players' facial expression, can be a crucial factor, to build a complete and structured DDA system. The classification of human facial expressions indicate whether a given subject is smiling, angry, sad, or others, and may be used to determine whether or not, a person is in a stressful situation. Facial expressions are often expressed automatically, without the transmitter even realizing that he/she is executing them [127]. Based on these signals, we adjust our behaviour and make choices according to our perception of the emotional state of the other person. In other words, the interaction may be conditioned by the mutual interpretation of the emotional state, derived from reading the other's facial expression (and complemented with body language, voice tone, and others). Extending this possibility to computers, the automatic identification of human facial expressions introduces several benefits. It has the potential to develop better and more useful human-computer interaction, provide visually impaired with haptic clues regarding the expression of others [107], monitor the motivation of students in the classroom [112], among many other applications. Although difficult, Facial Expression

Recognition (FER) is constantly evolving with several proposed solutions by the scientific community [143].

1.2 Objectives

The work described in this dissertation aims at the development, testing and validation, of several methods for two distinct computer vision tasks (FER and HAR), that will be potentially integrated, in order to fulfill a future DDA module (Figure 1.2). Grounded by previous research [18], the future DDA module, will consist on the development of a Reinforcement Learning (RL) system, that will benefit from several inputs as basis for decision (FER, HAR, the gameplay, etc.).

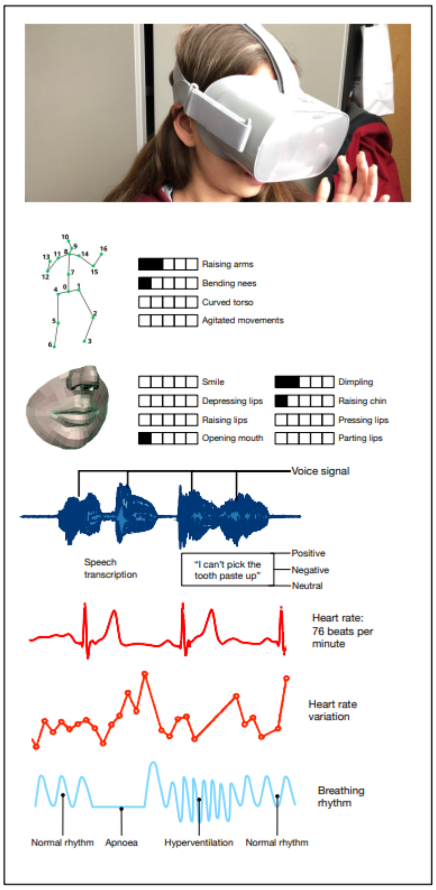


Figure 1.2: Biometric and physiological data sources for DDA [74].

Regarding the recognition of players' facial expression, two tasks are explored: face detection and expression recognition. The main objective of this module is to develop a system capable of evaluating humans facial expression in real-time and, eventually, use this information to assess the emotional state while playing a virtual reality game. It should be noted that since this work aims to integrate this module in a DDA system while playing a VR game, it introduces another challenge to this work: the classification of human facial expressions with a portion of the face occluded.

The HAR module, aims at quickly classifying the actions a player is performing in a VR environment. The classification of the task a player is performing may indicate whether he/she is performing the expected action, struggling with a difficulty or with no difficulty at all. For the time being, our methodology has only been tested on a public dataset; however, our dataset is being collected and will include a set of actions performed by a player while playing a VR game.

1.3 Structure

The remaining part of this dissertation is structured in 6 chapters, as follows: Chapter 2, discusses recent relevant approaches in the field of HAR and FER, detailing different methods and results; Chapter 3, describes the methodology used in this work to perform HAR; Chapter 4 shows the methodology used to solve the facial expression recognition task, with and without oclusion of part of the face; Chapter 5 introduces the datasets used in this work and also the development of our own dataset, based on several daily basis actions; Chapter 6 shows the results obtained using the proposed methodology to solve the two distinct tasks; and finally, conclusions are drawn and future work is proposed, in Chapter 7.

Chapter 2

Related Work

This chapter provides the most relevant state-of-the-art approaches in HAR, HAR in VR games, FER and also FER with part of the face occluded. Also, supported by our review paper [18], Table 2.1 provides an overview of the most relevant DDA approaches, showing the game type, the sensors/data sources used, the respective techniques and their achievements and limitations.

Table 2.1: Overview of Dynamic Difficulty Adjustment Games

Study	Game Type	Sensors/ Data Source	Techniques Used	Achievements	Limitations
Huber et al. [49]	Virtual Reality Entertainment Game	Head-Mounted Display (HMD); Electrocardiogram (ECG) sensor attached to patient's upper body	Deep Reinforcement Learning	Generation of levels based on player's skills.	Static user simulation. Generated levels to slow to allow for fast adaptation.
Tan et al. [123]	Entertainment Computer Game	Keyboard; AI algorithm	Reinforcement Learning; Adaptive Uni-Chromosome; Duo-chromosome Controller	Training is not required; adaptation takes place during the game session.	The adaptative controller is only good as its design; unable to adapt if a player exceeds its full potential.
Liu et al. [67]	Entertainment Computer Game	Wearable biofeedback sensors	Machine Learning Classifiers Highlited: Regression Tree	Dynamic difficulty adjustment based on players' affective state.	Unconfortable sensors. Training time too long.
Chanel et al. [19]	Entertainment Computer Game	Electrodermal Activity sensors	Deep Learning Techniques	Dynamic adaptation of the game, based on two possible emotional states.	Module just uses physiological data, nothing related to player's performance.
Blom et al. [12]	Entertainment Computer Game	Computer Camera	Machine Learning Classifiers; Heuristic Algorithm	Dynamic difficulty adjustment based on players' facial expression.	Adaptation based only on players' facial expression. More features could be added.
Hocine et al. [44]	Motor Rehabilitation	Computer Mouse; Graphics tablet	Monte Carlo Tree Search; Reward Based System; Computer Vision Algorithms	The game adapts to each profile in real time.	The number of different subjects and profiles is limited.
Sekhavat [108]	Motor Rehabilitation	Microsoft Kinect	RL; Multiple-Periodic RL	Automatic Adaptation based on real time patient's skills.	Low accuracy. It only considers players' skill level.
Ávila et al. [145]	Motor Rehabilitation - Virtual Reality Based	Abstract definition of the method; it was unclear how data was received	Reinforcement Learning; Markov Decision Process	Online Adaptation during game.	The therapist needs to be available during game.
Andrade et al. [4]	Motor Rehabilitation	Robotic Device	Reinforcement Learning	Motor Rehabilitation using robotics.	Small sample size. Just four players were included.
Pezzera et al. [87]	Motor Rehabilitation	Microsoft Kinect; Nintendo Wii Balance Board	Fuzzy Logic	Adaptation based not only on players' performance but also on theirs' physiological state.	Physiological state extracted by Virtual Theraphist during game, could be improved with other mechanisms, such as cameras, sensors, etc.
Balan et al. [14]	Mental Rehabilitation - Virtual Reality Based	Acticap Xpress Bundle (EEG data); Shimmers Multi-Sensory (Heart Rate and Galvanic Skin Response data)	Machine Learning classifiers; Deep Neural Networks	Virtual reality game that dinamically adapts its' difficulty based on patient current level of fear.	Small sample size, possibly leads to overfitting.
Balan et al. [15]	Mental Rehabilitation - Virtual Reality Based	Two HTC Vive Trackers; HMDS; EEG Device	Machine Learning classifiers; Deep Neural Networks	Adaptation based on physiological signals, facial expression and respiration rate.	Small sample size. Discomfort in the head by using sensors.

2.1 Human Action Recognition

HAR has received a lot of attention by researchers during recent years [22, 59, 90, 138]. There are numerous approaches to solve this classification problem. Our application seeks an efficient and fast classification of which action a human is performing, since this is needed in real time. This section aims to provide a set of relevant techniques that may be applicable to our specific problem. In addition, the current advances in HAR in a VR environment were also studied, to analyze what progresses have been made in this field.

Ullah et al. [126] have proposed an efficient action recognition system for processing data streams obtained from real-world dynamic scenarios. Their system's input can come from a variety of sources, including internet surveillance video data streams, websites, social media feeds, and other visual content resources. The authors used a pre-trained VGG-16 for frame level representation of an action in video streams, but in order to reduce computational time, they modified previous state-of-the-art Convolutional Neural Network (CNN) models by reducing the convolutional kernel size and also the stride. Then, from raw video frame data, an optimized Deep Autoencoder (DAE) was trained to accurately represent actions (sequence of motion patterns in consecutive frames). The DAE's goal was to compress those features so it could associate hidden changes in the low-dimensional feature plane from frame to frame. The authors emphasized the effectiveness of this method in comparison to more complex learning approaches such as Long Short-Term Memory (LSTM). Then, using a non-linear learning approach, they trained a quadratic Suport Vector Machine (SVM) to recognize actions from a low-dimensional features plane (DAE output), at a rate of 25 frames per second. Finally, in the testing phase, an iterative fine-tuning method was included to update the parameters of the trained model using freshly gathered data from the non-stationary environment. The authors have tested their method in several datasets such as: UCF50 [97], UCF101 [114], HMDB51 [61], and YouTube Action dataset [68]. They demonstrated system efficiency by achieving state-of-the-art accuracy results of 96.4%, 94.33%, 70.3%, and 96.21% respectively, while also reducing running time. The authors concluded that their system can be extended for video classification, human activity recognition, violent event recognition, and crowd analysis in dense environments.

Further more, Ullah et al. [125] have proposed an activity recognition framework for industrial surveillance systems. To analyze activity analysis, the authors have only selected the parts of a video where a human appears. To accomplish this, significant shots were chosen from a video stream using CNN-based human saliency features. The authors used a pre-trained MobileNet [105] and trained the feature maps on the INRIA person dataset [28], which learns to select only the salient regions that are activated for people in a video frame. These regions were used to extract salient features and segment shots, yielding representative shots suitable for industrial video stream analysis and activity recognition. Then, the authors used FlowNet2 [50], a CNN-based optical flow model, to extract temporal features. They chose optical flow because it is a popular source of motion estimation in video sequences. Due to the greater impact of LSTM in learning time series data, the authors used a multilayer LSTM to learn the long-term sequences in the temporal optical flow features for activity recognition. The authors tested their method using different benchmark action and activity recognition datasets, achieving an accuracy of 94.45%, 72.21%, 69.5%, 94.9% and 95.8% in the UCF101 [114], HMDB51 [61], HOLLYWOOD2 [77], UCF50 [97] and YouTube Action dataset [68] respectively. The results in all datasets are very similar to the ones presented by the same authors in [126].

Rao [96] was able to identify various kinds of human actions (wave, stand, punch, kick, squat, sit, walk, run and jump). The author started by estimating the human pose with OpenPose [17], preprocessing the data and scaling the images to deal with images with different sizes. The joints of the head and the frames where the neck does not appear were discarded. After this, the author proceeded to the feature extraction step. Using 5 frames, the most salient features were extracted, calculating the average height of the skeleton to normalize the features, the velocity of the joints and the length of each limb. Finally, Rao proceeded to classify the actions with the following classifiers: K-Nearest Neighbour (KNN), SVM, Deep Neural Network (DNN) and Random Forest. He used a division of 70% for training and 30% for testing. DNN achieved the highest accuracy (99.4%), due to its ability to deal with large amounts of data.

Zhang et al. [140] proposed a self-regulated view adaptive (VA) scheme who re-posit the observation in order to facilitate the action recognition. Then, based on a Recurrent Neural Network (RNN) and a CNN, they created two VA neural networks called VA-RNN and VA-CNN. The first was based on a subnetwork RNN, which consists of a network capable of transforming the skeleton into new representations for various observable viewpoints, and an LSTM to recognize actions from the skeletons. The second was based on a subnetwork CNN to learn and determine the sequence-level observation viewpoint and a main CNN capable of extracting features from the skeleton (both networks were trained end-to-end to optimize the classification performance). After, they fused the scores of the two networks (VA-fusion) to achieve better results and better performance to provide the final prediction. They used five different datasets: NTU [109], SYSU [46], UWA [93], N-UCLA [128], SBU [137] obtaining 95% , 86.7%, 81.4%, 88.1% and 98.3% of classification accuracy respectively. It should be noted that these values were obtained with the VA-fusion, concluding that it provides better results.

By combining multiple vision cues from an RGB-D sensor, Khaire et al. [56] presented a novel method for generating pose images from joint sequences that represent motion. They have created a data processing method, trained on a CNN, which acted as individual classifier to recognize activity. The authors combined 5 CNN streams (classifiers) of RGB, Depth, and Skeletal data at the decision level, the last one revealed a critical factor in their approach. For action recognition they used both pre-trained VGG-16 (large architecture) and VGG-F models (small architecture), which proved to be competitive models comparing to state-of-the-art. To evaluate the relevance of their approach, the authors have tested their method in three well known datasets, the CAD-60 [119], SBU Kinect interaction [137] and UTD-MHAD [20], achieving an accuracy of 95.11%, 96.67% and 94.60% respectively. The authors also highlighted that their method is better suited for indoor scenes where the background is almost static or constant.

Kamel et al. [54], presented a new HAR method that used three channels (RGB - Red, Green and Blue) of deep CNNs from posture data and depth maps. To represent body posture sequences, the authors have used a Moving Joints Descriptor (MJD). This descriptor provides crucial information about the joints' movement directions, as well as changes in joint poses, based on the size of angles. To represent depth maps sequences the authors have used a Depth Motion Image (DMI) descriptor. This descriptor was used to represent changes in action depth from the front view only, rather than two views as in [57]. This was possible with the help of the MJD descriptor, which also reduced computation complexity. Three CNN channels were used in the action recognition process, which were trained with DMI and MJD descriptors for feature extraction and classification. DMI was used to train the first channel, and the second channel is a link between two sub-channels. DMI was used to train one sub-channel, while MJD was used to train the other. Only MJD was used to train the third channel. To maximize the score value of the right action, the authors employed score fusion operations. In a nutshell, each CNN channel assigns a score to each action, with the highest score indicating the correct action. To evaluate their approach, they used three public datasets: MSRAction3D [66], UTD-MHAD [20], and MAD dataset [48], achieving a competitive accuracy comparing to state-of-the-art, of 94.50% (a little lower than state-of-the-art), 88.14% and 91.86% respectively.

Jaouedi et al. [51], proposed a fully automated scheme for HAR using a fusion of DNN and multiview features. The majority of cutting-edge techniques use deep learning to focus on a single view of human representation. The recognition of multi-view actions is a difficult task due to a variety of factors such as illumination, human styles (walking, jogging, listing phones, punching, etc.), and the quality of selected videos. The authors extracted multiview features by using gradient information from both the x- and y-axes and then combining this information. They used the entropy max activation function to perform transfer learning on original pre-trained models (VGG19) in order to improve classification accuracy. Three metrics, relative entropy, mutual information, and high correlation coefficient, were used to choose the best features (SCC). Furthermore, using a higher probability based threshold function, these parameters were used to choose the best subset of characteristics. For final recognition, the final selected characteristics were provided to the Naive Bayes classifier. To compare their work with state-of-the-art methods, the authors have used five datasets: HMDB51 [61], UCF Sports [101], YouTube [68], IXMAS [131], and KTH [68]. They have achieved better results than state-of-the-art, with a classification accuracy of 93.7%, 98%, 99.4%, 95.2%, and 97%, respectively.

Recently in 2021 Mroz et al. [82], performed a comparing study between OpenPose [17] and BlazePose [9] to determine if these models can produce clinically valid body keypoints for virtual motion evaluation. The authors of this study assessed the efficacy of detecting keypoints using Pearson correlation and root mean square error metrics. When compared to OpenPose, BlazePose exhibited more instances where keypoints strayed from anatomical joint centers, indicating that the BlazePose was not yet the ideal solution for clinically meaningful evaluations. The BlazePose runtime, on the other hand, was significantly quicker than OpenPose (approximately 6 times faster) and generated metrics that could be used in a smartphone application. The authors concluded that for their application the OpenPose was the best model for pose estimation. However, BlazePose has also the z coordinate which can represent significant information about human pose.

In order to provide a brief overview of the efficacy of the detailed studies, the Table 2.2 shows all state-of-the-art methods described and also their respective accuracy.

Table 2.2: Accuracy State-of-the-Art HAR methods on different datasets

Author	UCF50 [97]	UCF101 [114]	HMDB [61]	Youtube [68]	HOLLY [77]	NTU [110]	SYSU [47]	UWA [94]	N-UCLA [129]	CAD-60 [119]	SBU [137]	UTD-MHAD [20]	MSR3D [66]	MAD [48]	UCF Sports [101]	IXMAS [131]	KTH [68]
Ullah et al. [126]	96.4%	94.33%	70.3%	96.21%	-	-	-	-	-	-	-	-	-	-	-	-	-
Ullah et al. [125]	94.9 %	94.45%	72.21%	95.8%	69.5%	-	-	-	-	-	-	-	-	-	-	-	-
Zhang et al. [140]	-	-	-	-	-	95%	86.7%	81.4%	88.1%	-	98.3%	-	-	-	-	-	-
Khaire et al. [56]	-	-	-	-	-	-	-	-	-	95.11%	95.11%	94.60%	-	-	-	-	-
Kamel et al. [54]	-	-	-	-	-	-	-	-	-	-	-	88.14%	94.50%	91.86%	-	-	-
Jaouedi et al. [51]	-	-	93.7%	99.4%	-	-	-	-	-	-	-	-	-	-	98%	95.2%	97%

2.2 HAR in a VR environment

Earlier in 2008, Choi et al. [24], proposed a real-time system that robustly tracks multiple people and recognizes their actions using image sequences acquired from a single fixed camera, allowing multiple people in a virtual environment to interact with virtual agents simultaneously and conveniently. They did this by processing each frame individually, extracting blobs using the Mixture of Gaussian technique [13], and removing shadows and highlights to get a more accurate object silhouette. Finally, they model an action as a Motion History Image (MHI) based on specified object tracks, normalize the MHI, reduce the MHI using Principal Component Analysis (PCA), and categorize an action using a multi-layer perceptron. To demonstrate their approach, they used it in an Augmented Reality (AR) application where numerous people could interact with a virtual pet, demonstrating that the system works. However, they did not test their system with any available dataset.

Kwon et al. [62], presented a virtual training simulator that used multiple Kinect sensors to provide a trainee with an interactive training environment (military training). To this goal, a 360-degree multiview human action identification system including coordinate system transformation, front-view Kinect sensor tracking, multi-skeleton fusion, skeleton normalization, orientation adjustment, feature extraction, and an SVM to perform classification was used. This enabled trainees to enjoy a realistic and immersive virtual training by recognizing their actions and synchronizing them with VR information. They tested their system with their own database, demonstrating its utility in military training, achieving an average accuracy of 96.5%.

Fangbemi et al. [33], presented a novel AR and VR interaction interface based on HAR with a new binary motion descriptor that can describe and recognize different actions in videos quickly and accurately. To accomplish this, they claim to have developed a new compact patch pattern (PP pattern) that includes all pixel information in the closest vicinity of a keypoint to describe its motion between three consecutive frames. With this, they created the Binary Proximity Patches Ensemble Motion (BPPEM) descriptor, which computes the change in texture at the same point from three consecutive frames using the PP pattern, as opposed to previous works that computed the descriptors using different positions. They used the Weizmann and KTH [68] datasets to compare their method to state-of-the-art spatio-temporal binary descriptors in order to evaluate their approach (through the use of an SVM, to perform the classification). They achieved lower classification accuracy than most state-of-the-art methods, despite being the faster method and having a good classification accuracy of 92.22 % for the Weizmann dataset [11] and 91.14 % for the KTH dataset [68].

Since the reality of VR is based primarily on human-computer interaction, which is closely related to human action recognition technology, Ma in [76] has explored HAR algorithms to improve smart cultural tourism in a VR environment. The author proposed an action recognition algorithm based on a Self-Organizing Mapping network (SOM). A SOM is a low-dimensional discrete mapping produced by learning the data in the input space and gradually optimizing the network using a competitive learning strategy, which has the self-organizing properties of the human brain and is capable of identifying the intrinsically related characteristics in a problem [58]. Due to this useful properties, the authors used a SOM neural network to obtain the key frame of tourists' actions, shortens recognition time, and combined it with multi-feature recognition method to improve action recognition accuracy. They have evaluated their method with the UT-Kinect action dataset [133], achieving an accuracy of 93.68%, the highest among the methods they tested (Histograms of 3D joints, Skeleton joint features, Random forest fusion STIP feature method).

When dealing with HAR classification problems, it is possible to conclude that skeleton-based, image-based, shallow-based, and deep-based methodologies can all be successful. Some of these methods combine skeleton information with deep learning methods that analyze the entire image rather than just the detected skeleton. This can be very effective because the two pieces of information can complement each other.

2.3 Facial Expression Recognition

Over the centuries there have been several attempts to organize emotions into several categories. Cicero [91], held that they should be divided into four categories: fear (*metus*), pain (*aegritudo*), lust (*libido*) and pleasure (*laetiti*). Darwin and Prodger [92], for example, have used twenty-two categories, including anxiety, joy, love, devotion, anger, helplessness, surprise, fear, among others.

More recently, Plutchik has introduced the wheel of emotions (Figure 2.1), with 8 primary dimensions of emotion, namely joy, trust, fear, surprise, sadness, disgust, anger, anticipation [31]. The distance to the center dimension represents the intensity, that is, the emotions are intensifying as they move from the outside to the center. For example, boredom can intensify to disgust, and from this to loathe, if not controlled. Each sector has an opposite emotion (the opposite of joy is sadness, the opposite of anger is fear, and so on). Emotions that have no corresponding color represent an emotion that is the mixture of two primary emotions, for example, the combination of joy and trust results in love.

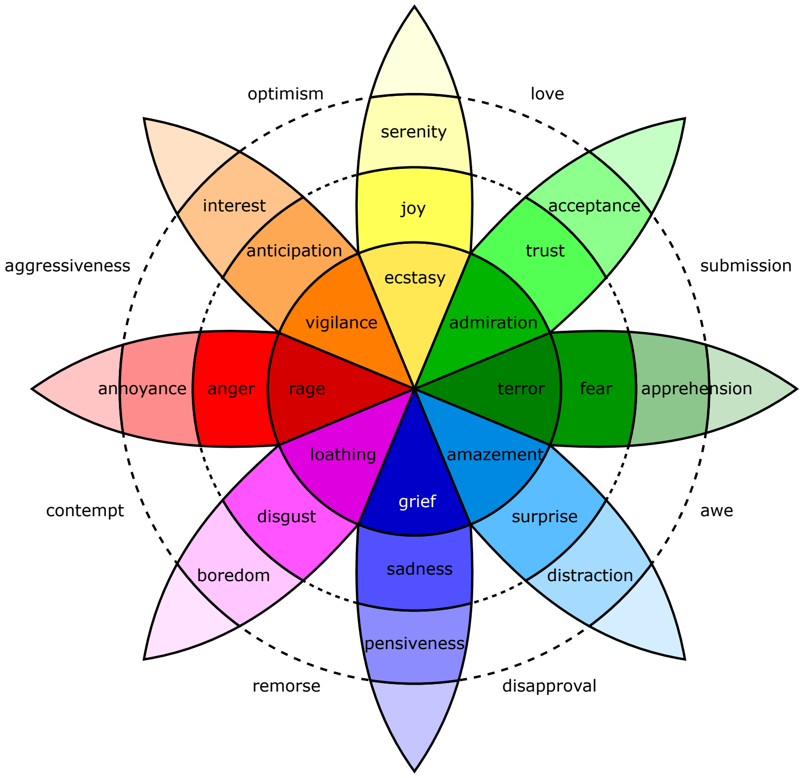


Figure 2.1: Wheel of emotions [31].

The sole analysis of facial expressions is not enough to assess a person's sentiment. However, 55% of information is communicated by facial expressions, 38% by other gestures and signals (such as voice and sound) and 7% by spoken language [78]. Based on this, FER account as an important factor for human sentiment recognition. With constantly improving of the computing power, jointly with the development of big data processing technology and algorithms, the automatic classification of facial expressions have been growing in accuracy and interest.

Devries et al. [30], demonstrated that a system trained to reason about facial geometry while recognizing expressions outperforms one trained solely to recognize expressions (Figure 2.2). Since facial landmark prediction has many publicly implementations available, the authors decided to use Zhu and Ramanan's facial landmark detector, which uses coordinates for 68 reference points per face. All of these points are used to outline facial features like the mouth, nose, eyes, and eyebrows. The authors simplified the problem by focusing on the most expressive features of humans facial expression: the eyebrows and mouth (rough reference points). Each of the positions of the left brow, right brow, and mouth were represented by a binary mask image.

The authors have used a CNN that was based on the winning architecture of the 2013 ICML Facial Expression Recognition Competition [124]. They opted for a CNN with 3 convolutional layers fully connected, each with a ReLU activation function and max pooling. The network's

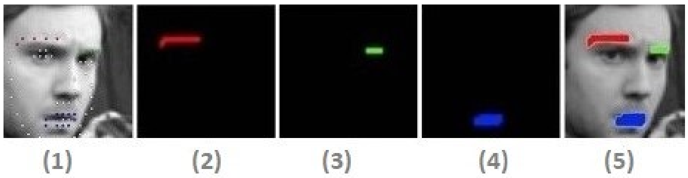


Figure 2.2: Facial landmarks for expression recognition: 1) image with 68 facial landmarks; 2) left eyebrow created by aggregating the red landmarks; 3) right eyebrow; 4) mouth; 5) overlay aggregated landmarks on the original image [30].

output consisted of three binary output maps (one for each of the reference locations), and it was thus in charge of modeling the location and shape of each of its features. Two datasets were used in their work: ICML Dataset [40] and TFD database [120]. The ICML dataset consists of 28709 48x48 training images, each with 7 labels and 7177 test images. Since the images were taken from a wild environment, the faces have different orientations and are not always facing forward. In this dataset, the authors evaluated 2 different techniques: CNN with 3 convolutional layers fully connected and a Multi-task CNN. TFD database is also made up of 48x48 images and 7 labels, however, all faces are looking directly at the camera. It contains 4178 images, 70% of the images were used for training, 10% for validation, and 20% for testing. The authors used the same evaluation techniques applied in the ICML dataset. The authors achieved the best results by employing the multi task network, which resulted in a classification accuracy of 67.21% in the IMCL dataset and 85.13% in the TFD database.

Baltrusaitis et al. [5] use the OpenFace facial behavior analysis pipeline (Figure 2.3), which includes the following algorithms: facial landmark detection, head pose tracking, eye gaze and facial Action Units. For the facial landmark detection and tracking, the authors have used the Conditional Local Neural Fields (CLNF), with 2 main components: Point Distribution Model (PDM) and patch experts. The PDM was been trained in 2 datasets, the LFPW and Helen. The CLNF patch experts has been trained in 3 datasets, including the ones used in the PDM and also the Multi-Pie dataset [42].

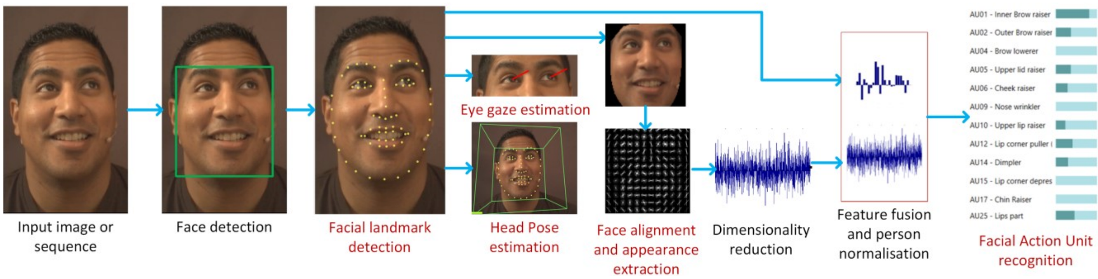


Figure 2.3: OpenFace behavior analysis pipeline [5].

In order to estimate the head pose, information about the position of the head (translation and orientation) was extracted, as well as the detection of facial landmarks. This information was obtained thanks to the CLNF, which internally uses a 3D facial landmark representation and projects it to the image using orthographic camera projection. To train their model the authors have used the Mpiigaze [141] dataset.

Regarding the eye gaze estimation, CLNF and PDM were also used, since they allow detecting reference points of the eye region, such as the eyelids, iris and pupil. The PDM was trained with the Syntheseyes [132] dataset. The information obtained by the CLNF was used to compute the eye gaze vector individually for each eye (lightning is fired from the camera's source through the center of the pupil in the image plan, and its intersection is computed to determine the pupil's location in 3D camera coordinates). Finally, Openface predicts AU presence using a linear (SVM) kernel and AU intensity using a linear Support Vector Regression (SVR) kernel.

Loizou [71] proposed and evaluated a system for analyzing automated speech signals and images for seven different human emotions: normal, happy, sad, dislike, fear, anger, and surprise. Voice and image recordings of more than 70000 people aged twenty to seventy-four years old were organized. The authors have used multi-classification models to select the features that identify the seven emotions through an SVM with a 10-fold cross validation, using a Gaussian Radial Basis Function with c=1 and γ=0.01. Statistically, a correct classification score of 93% was obtained.

Mindlink-Eumpy [65], an open-source toolbox, was designed to detect emotions by integrating information from EEG and facial expressions. First, a set of tools was used to automatically collect physiological data, which was then used to analyze user facial expressions and EEG data. Regarding the analysis of user's facial expression, they used a multitask CNN pre-trained with the FER2013 dataset [40].

On the surface, the idea of using a pre-trained CNN, is to transfer the labeled data or knowledge from some domains (previously performed tasks - Source Tasks), to help the Machine Learning (ML) algorithm to perform better in the domain of interest. Regarding the EEG analysis, MindLink-Eumpy [65] uses two different algorithms: SVM and LSTM. In the decision-level fusion, Weight enumerator and Adaboost techniques were applied to combine the predictions of the CNN and the SVM. The authors have achieved an accuracy of 71% for SVM and 78.56% for the LSTM.

Almeida and Rodrigues [3], developed a system capable of capturing real-time images and alerting the user if there are any signs of stress. The system was divided into several modules, such as (Figure 2.4):

1. Real-time image capture, via computer camera, sending the images to the next module.

2. Determining the user's face position by the Haar-like feature. The image is further readjusted and normalize.

3. After properly training the classification model, the model is able to classify each face and return a list of seven classification probabilities, one for each facial expression.

4. The facial expression most likely to be embedded in this module determines whether or not the person is under stress.

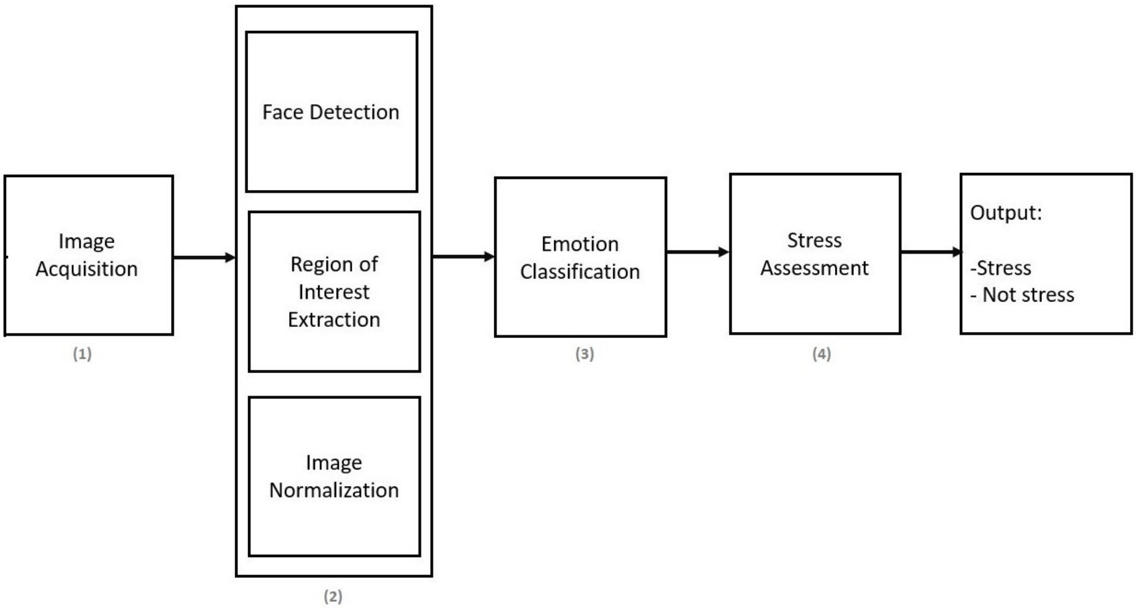


Figure 2.4: Real face detection scheme [3]

The authors have used a CNN previously trained (transfer learning), that was applied using two different techniques: Global Average Pooling (GAP) and Convolution Layer. The classification took into account seven emotions (as in [71]). The authors used two multi-classification models, using them to predict facial expressions and binary classification to classify images with stress/non-stress. They achieved an accuracy of 92% in the best model (VGG16 [122]).

2.4 Occlusion

In [8], Bartlett et al. developed a real-time face recognition system that can recognize faces in a video sequence and encode each frame with one of the seven corresponding emotions. All the recognized faces were converted into images of the same size, using Gabor energy filters and were later analyzed by Adaboost, which encodes facial expressions in 7 dimensions (corresponding to 7 emotions). The system was trained and tested with an SVM in the DFAT-504 dataset [55], which is constituted by 100 university students between the ages of 18 and 30: 65% were female, 15% African American, and 3% Asian or Latino. To validate their approach the authors combined an SVM with Adaboost and named it as Adasvm, which produced an accuracy of 93.3%. By themselves, Adaboost achieved an accuracy of 90.1% and SVM 89%, so both were used simultaneously.

Considering situations where the face is partially hidden, mainly the eyes, Cheng et al. [23] used the following approach:

1. Images containing faces are segmented from human images of the same size;
2. The result obtained in the first step is then used to normalize and transform the images into figures of Gabor magnitude through multi-scale and multi-orientation of the Gabor filters.

Through these filters the low-level image characteristics of facial figures are reinforced, such as the edges, peaks, contours of crests, eyes, nose and mouth, which are considered to be the main components of the face;

3. The Gabor characteristics are extracted to form a 2D matrix, and the sampling completed downwards to slightly reduce the dimension;

4. The samples are divided into mini-batches and the weights are updated to speed up the pre-training of each Restricted Boltzmann Machine (RBM), which is a bipartite structure with a visible layer and a hidden layer;

5. According to the dimension of the features, set the size of each layer from three layers network (in general);

6. Generate weights and adjust them by fine-tuning;

7. The deep structure training process is divided into two stages: pre-tuning, which treats the data labelled as unmarked for unsupervised training to provide each weight from the lower layer to the top, and fine-tuning which is a simple process of gradient descent under supervision;

8. The test is carried out in number 6 until convergence.

In addition to the Gabor filter, the authors also used other methods that use this filter to compare with the proposed method: Local Gabor Binary Pattern Histogram Sequence (LGBHPS), modified LGBHPS, with the KNN Classification method respectively. The authors used the JAFFE dataset [34], which consists of 213 images of 10 different individuals with seven different facial expressions: happiness, anger, sadness, fear, surprise, disgust and neutral. Considering that this dataset is not available with natural partial occlusion facial images, the authors simulated the occlusion by overlay graphic masks in the images of this dataset (Figure 2.5).

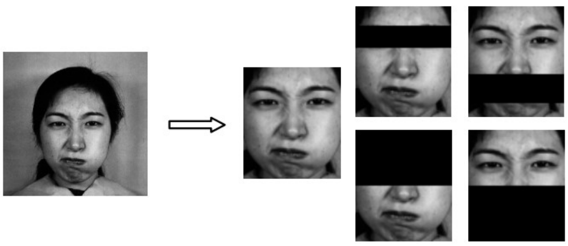


Figure 2.5: Partial occlusion process made in the JAFFE dataset [23]

The resulting images are divided into 2 parts: 143 images for training and 70 images for testing (images containing occlusion of the eyes, mouth, upper and lower parts of the face). They obtained an accuracy of 85.71% with no occlusion, 82.86% with occlusion of the eyes, 77.14% with occlusion of the upper part of the face and 82.86% with occlusion of the lower part of the face. These results are superior to all other methods used for comparison by the authors, although the reduced number of individuals in the dataset could be increased, which will be an impacting factor in the classification process.

Houshmand and Mefraz [45] focused essentially on facial expression identification in the presence of a severe occlusion while the subject is utilizing a head-mounted display in a Virtual Reality (VR) environment. Since display measurements are known, the authors were able to replicate occlusion caused by these headsets using face detection applied to grayscale images using a modification to the conventional Histogram of Oriented Gradients (HOG) and Linear SVM-based approach for object detection. The authors estimated 68 reference coordinates that map the face anatomy on the iBUG 300-W dataset [104]. Because the dataset contains images with varying sizes, the distance between the two temporal bones of the temporal landmarks was used as the reference length, and the polygonal occlusion patch was generated using the midpoint of the line that passes through the center points of the eye as the VR headset's central coordinate (Figure 2.6).

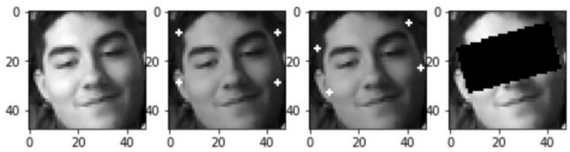


Figure 2.6: VR occlusion patch [45].

The authors have evaluated the effectiveness of two different architectures: VGG and ResNet. They chose three datasets that cover different scales of face images and contain images with momentary occlusions to test these two architectures, namely FER+ [7], RAF-DB and Affectnet. The authors have defined a rescale factor of 224*224 at the input of the CNNs, which meant that all images used for training, testing, and validation were rescaled for the same measurements and normalized using min-max normalization. The training procedure was carried out by optimizing the multinomial logistic regression objective, which employs mini-batch gradient descent based on momentum back propagation. To regularize the train phase in the ResNet, a max-norm kernel restriction was added. The best result obtained by the authors was 79.98% of accuracy in the ResNet model with transfer learning in the FER+ dataset.

Cornejo et al. [95] have structured their approach in 5-steps. First, pre-processing on all images. This includes the automatic detection of the facial fiducial point, and then the coordinates of the eyes are extracted, the image is rotated, and the image is aligned. Still at this stage, facial expression regions are cut through an appropriate bounding rectangle, RGB images are converted to grayscale, and then randomly generated rectangles are applied over various regions of the face, such as the left lower eye, right eye, both eyes, right lower side, or lower side. Next, occluded facial expression is reconstructed with the Dual Algorithm based on the principles of the Robust Principal Component Analysis (RPCA) where the Contrast-Limit Adaptive Histogram Equalization (CLAHE) is subsequently applied to reconstructed facial regions, to increase image contrast levels (Figure 2.7). Third, a set of facial expression characteristics were extracted through 3 strategies: Weber Local Descriptor (WLD) - applied over the entire facial expression image for extraction of textural features, Local Binary Patterns (LBP) - applied over the entire image to extract histograms of LBP and HOG - applied to the entire image, to extract the HOG features. Fourth, reduction of dimensionality of the characteristics extracted in the previous step, and the resulting descriptor is transferred in a lower dimensional space through PCA and LDA, applied sequentially. Finally, occluded facial expressions are recognized through KNN and SVM classifiers.

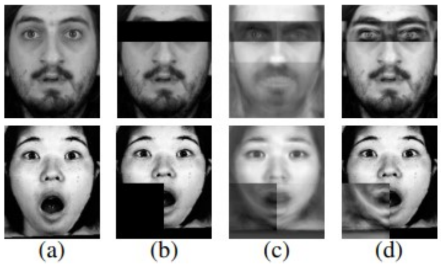


Figure 2.7: (a) Cropped images without occlusions from the MUG dataset [2] and Jaffe dataset [34]; (b) faces with occlusive areas; (c) reconstructed faces via RPCA; (d) filling the occluded facial regions from (c) [95].

The authors have tested the proposed method on three datasets: CK+ [75], JAFFE [34] and Facial MUG Expression [2] and obtained results of 91.01% accuracy in CK+ dataset with KNN, 92.86% in the JAFFE dataset with SVM and 90.1% in MUG with KNN, in occluded images.

The work described in this dissertation builds on these, to study the impact of occlusion caused by VR goggles in the identification of facial expressions.

Chapter 3

Human Action Recognition

A proper and structured methodology is one of the most important components of a successful application. This chapter will go over each of the methods used to perform HAR, such as feature extraction, image normalization (based on angles, MJD, keypoint normalization and graph embedding) and the deep neural network used for classification. Figure 3.1 shows an overview of the methodology that will be detailed in this chapter.

3.1 Feature Extraction

To process video sequences, AI algorithms typically treat them frame by frame, applying the algorithms throughout the sequence. One of the key elements for an algorithm to succeed in HAR, is the correct detection of a human as well as the ability to identify the motion patterns. Many algorithms have been proposed over the years to accomplish this [84, 106], some of which are open source and easy to be adapted. Although skeleton-based approaches are very popular in HAR, there are also other options for dealing with this classification problem [25,43,52,118,130]; however, this dissertation relies solely on the use of these skeleton algorithms.

3.1.1 OpenPose

Due to its popularity, OpenPose [17] is currently one of the most adopted algorithms to perform this detection. OpenPose sequences (video), are processed frame by frame to obtain the entire skeleton, which is then represented by 18 points, using the COCO and MPII models [16]. OpenPose (Figure 3.2) employs a multistage feedforward CNN to identify the locations of anatomical bodypoints for each person in the image. Part Affinity Fields were used by the authors to represent the limb probabilities between each pair of connected part-types. The original algorithm was implemented in Caffe [53], however to synchronize it with the rest of the project, it was adapted to Pytorch [116].

[1]Source: https://www.istockphoto.com/pt/foto/a-young-bearded-man-in-casual-wear-stands-with-hands-on-his-sides-on-a-white-background-gm965662288-263521418?phrase=standing%20straight

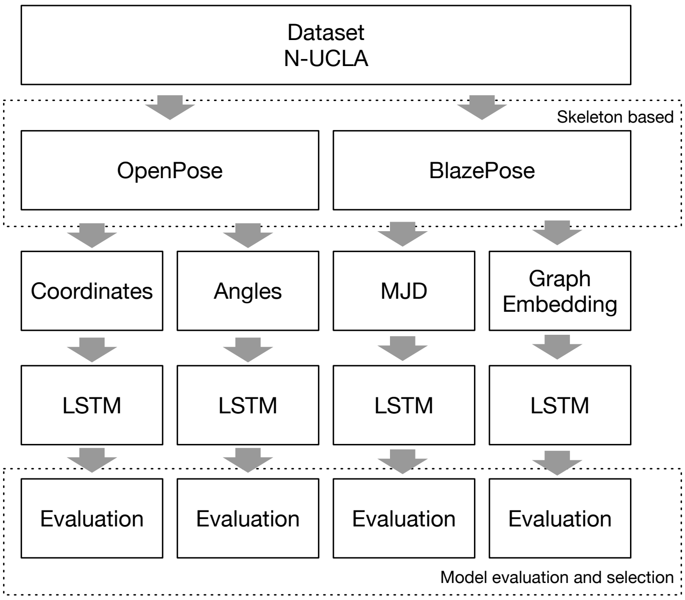


Figure 3.1: HAR Methodology Scheme.

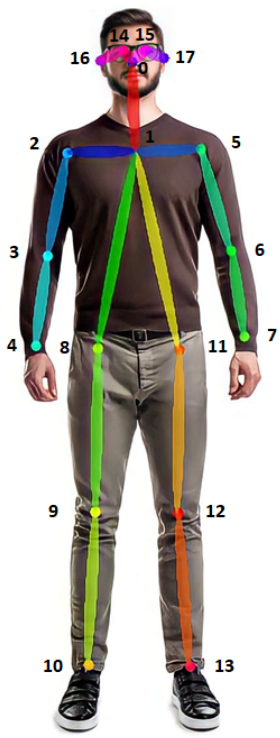


Figure 3.2: OpenPose Skeleton Detection[1].

Figure 3.2 shows OpenPose's perfect detection on a front-facing person. This algorithm recognizes the human skeleton using 18 keypoints ranging from the head to the legs. The OpenPose algorithm is limited, since it can only detect x and y coordinates and cannot extract a 3D representation of the skeleton (no depth information).

3.1.2 BlazePose

BlazePose [9], introduced by Google in 2020, is a 3D skeleton detector algorithm designed for real-time inference on mobile devices. This model is a ML method for high-fidelity body posture tracking, that uses RGB video frames to infer 33 3D landmarks and a background segmentation mask for the entire body. For inference, current state-of-the-art algorithms rely on powerful desktop environments, but their solution delivers real-time performance on most recent mobile phones, desktops/laptops, using Python, and even on the web. To forecast heatmaps for all joints, the authors employed an encoder-decoder network architecture, followed by another encoder that regresses straight to the coordinates of all joints. Figure 3.3, represents the topology of the keypoints detector.

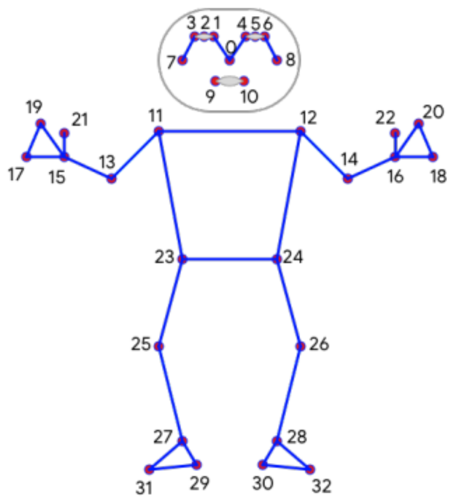


Figure 3.3: BlazePose Keypoint Topology [9].

3.2 Image Normalization

Four different approaches were investigated for describing the detected skeleton, in order to feed the DL models with representative and useful information. The first technique is based on the normalization of the keypoints, the second on the calculation of angles between joints, the third, the MJD descriptor (based on the work of Kamel et al. [54]) and finally the fourth is a graph representation approach. The coordinates of each skeleton joint were used in all described algorithms. It should be noted that not all methods described in this section, were applied to both algorithms' extracted skeletons. The keypoints were only normalized in the OpenPose algorithm because the coordinates are already normalized in BlazePose. The computation of the angles between joints was limited to the OpenPose skeleton (2D), since the 3D formulation would be much more complex. The MJD was only applied to BlazePose normalized coordinates because this descriptor relies on 3D information for classification. The Graph Embedding approach was applied to both skeletons, extracting the connection of the keypoints, forming edges.

3.2.1 Keypoints normalization

The OpenPose algorithm outputs its 18 joints, represented in image coordinates. One way to normalize them, is to find lowest x and y in the image, the maximum x and y. Having these four points, it is possible to normalize each real coordinate. Using the same image used before, Figure 3.2 illustrates this referential on a standing still person.

Figure 3.4: OpenPose Rectangle Referential.

The new coordinates will be then computed as follows (Equation 3.1 and Equation 3.2). For each joint:

$$new_x = \frac{x - min_x}{max_x - min_x} \tag{3.1}$$

$$new_y = \frac{y - min_y}{max_y - min_y} \tag{3.2}$$

where:

min x - Minimum x of all joints.

min y - Minimum y of all joints.

max x - Maximum x of all joints.

max y - Maximum y of all joints.

3.2.2 Angles between joints

As shown in Figure 3.2, the detected OpenPose skeleton contains 18 keypoints (vertexes). The angles formed by the skeleton are one way to potentially extract important information from a human's action. To this end, 12 angles have been defined, being them the angle formed by a set of 3 vertices:

1. 0, 1, 2;
2. 1, 2, 3;
3. 2, 3, 4;
4. 0, 1, 5;
5. 1, 5, 6;
6. 5, 6, 7;
7. 0, 1, 8;
8. 1, 8, 9;
9. 8, 9, 10;
10. 0, 1, 11;
11. 1, 11, 12;
12. 11, 12, 13;

It is then necessary to develop a method for computing the angles formed by the three points using these vertexes sequences. To accomplish this, two new vertexes were designed, which are defined as:

New Vertex1:

$$x = vertex2[x] - vertex1[x] \tag{3.3}$$

$$y = vertex2[y] - vertex1[y] \tag{3.4}$$

New Vertex2:

$$x = vertex3[x] - vertex2[x] \tag{3.5}$$

$$y = vertex3[y] - vertex2[y] \tag{3.6}$$

For example, in the first set of three vertexes, vertex1 would be keypoint 0, vertex2 would represent keypoint 1 and vertex3, the keypoint 2. Having these new vertexes, a normalization is performed in order to translate the real coordinates to a unit vector (0 or 1). The angles are then calculated with numpy's *arccos* function, which computes the inverse of a cosine. Figure 3.5, illustrates how the angle was computed.

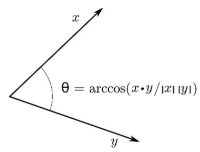


Figure 3.5: Angle calculation.

3.2.3 Moving Joint Descriptor

The MJD in our work is inspired by the work of Kamel et al. [54], although with some adaptations. They have only used 13 joints (out of 20) and in this work, the original 33 joints that the BlazePose algorithm outputs were used. Furthermore, they used the hip joint as a central keypoint, which BlazePose does not recognize. To overcome this, our hip estimation, can be calculated by the mid-point between the closest joints (Figure 3.6). The hip center, since it is usually the most stable in the body, it is chosen as the spherical coordinates origin, which means that all the other joints will be described by two angles and a distance to this reference.

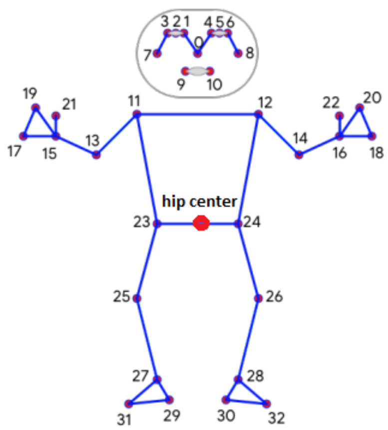


Figure 3.6: Hip Center - BlazePose.

Because the z coordinate has a significant impact on the entire process, this algorithm was only used to normalize BlazePose's coordinates since it is ineffective without the three coordinates (x, y, and z). Since the BlazePose algorithm normalizes the real coordinates of each joint, it was necessary to convert them back to real coordinates. After this process it was then possible to convert the coordinates, to spherical ones. Equations 3.7, 3.8 and 3.9, represent how it was processed the conversion of real coordinates to spherical.

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3.7}$$

$$\theta = \arccos\left(z/r\right) \tag{3.8}$$

$$\phi = \arctan 2(y, x) \tag{3.9}$$

Following this process it was then necessary to compute the MJD. For each joint, the hip location was estimated and subtracted from the coordinates of each joint.

3.2.4 Graph Embedding

Graph embedding techniques attempt to reproduce an entire graph as a single vector. Since neural networks expect vectors rather than graph data, this transformation can be very useful as an input to DL techniques. The conversion to vector format is extremely useful when comparing two graphs and computing the differences between them.

Graph2Vec [83] is a graph embedding approach inspired by word2vec [80] that treats a whole graph as a document and the rooted subgraphs surrounding each node as words that make up the document, and extends document embedding neural networks to learn representations of entire graphs. Graph2Vec has also the ability to learn the embedding of arbitrary graph sizes, which means that there is no need to respect a fixed or limited size. It is worth noting that since graph2vec's embeddings are learned unsupervised, there is no need to specify the graph's label. Graph2Vec is an open source Python algorithm provided by the KarateClub library [102], which provides the necessary tools to convert a list of edges to a graph format. It is expect that these graphs will then be used as input for the Graph2Vec model, that will embed the input graphs.

FeatherGraph [103] is a graph embedding descriptor that uses a characteristic function to generate node-level feature vectors and combines them to form graph embeddings. The authors proved that FEATHER is resilient to data corruption and defines isomorphic graphs with the same representation. Using node feature characteristic functions, they defined parametric models where the function evaluation points are learned parameters of supervised classifiers. This algorithm can be considered as a node embedding technique since it generates a mapping of nodes to Euclidean space by simply evaluating the characteristic function for metadata-based generic, neighborhood, and structural node attributes.

With these algorithms in mind for embedding graph information, the OpenPose algorithm and BlazePose were used to generate the list of graphs responsible for converting each frame of each video to a list of edges. By taking into account the coordinates of each respective vertex of the detected skeleton, all vertices were looped and then the euclidean distance between the actual and all the vertices was computed; Then, the edges are formed by connecting the two closest vertices to the actual one. Also note, that the graphs are weighted based on the normalization of all euclidean distances (having higher weights when this distance is closer to the actual point). Following this algorithm, and since the OpenPose skeleton has 18 vertices, each graph will have 36 edges. On

(a) Person standing still. (b) Person picking a bottle.

Figure 3.7: Graph Detection Examples [128].

the other hand, since BlazePose has 33 vertexes, each graph will have 66 edges. Figure 3.7, shows the detected graph of a random frame.

For example, it is possible to see in Figure 3.7a, that when a person is standing still, the vertices representing the legs and ankles are only connected with each other and separated from the rest of the body. The graph representation can be very useful in determining whether a person is crouching, as it is visible in Figure 3.7b. The graph connections in this figure are very different from a person standing still, representing the possible benefits of using a graph representation.

3.3 Deep Neural Network Classification

The DNN classification method is also an important step in the HAR process, as it requires a quick and accurate response after training. Because of its ability to deal with sequences, LSTM can be viewed as a critical component when dealing with videos.

LSTM is a type of RNN used in Deep Learning which has memory to keep information it needs and forget the things who are no longer applicable. RNN uses an input gate that is processed by the hidden state, sigmoid and tanh functions and subsequently reaches an output. That output is calculated recursively, as a loop, added into a new input until the classifier reaches the final output. The LSTM classifier uses the logic of RNN, but it is also composed by a cell that consists of 3 gates that are forget gate, input gate and output gate. As the name says, the forget gate defines what information can be forgotten because is not longer relevant, the input gate says what information could be added into the cell and the output gate is what information should be output in an instance of the loop. Each gate is capable of sending part or the complete information. It means that the forget gate can forget all the information or part of that, for example [39].

Chapter 4

Facial Expression Recognition

Another goal of this dissertation is to study the impact of occlusion in the classification of facial expressions for the dynamic adaptation of the difficulty level in cognitive rehabiliation serious games [72]. To achieve this, this chapter describes the methods used to perform the occlusion as well as the algorithms used to perform FER with and without occlusion.

4.1 Without occlusion

The architecture of the classifier is defined starting with the dataset without occlusion.

According to the related work Sections (Section 2.3 and Section 2.4), the followed approach in this work is based on CNN ensemble, composed of a ResNet-18 and a VGG19 (Figure 4.1) [136]. The weights were initialized with the Imagenet [29] weights for both ResNet-18 and VGG19, with Xavier initialization for the final fully connected layer.

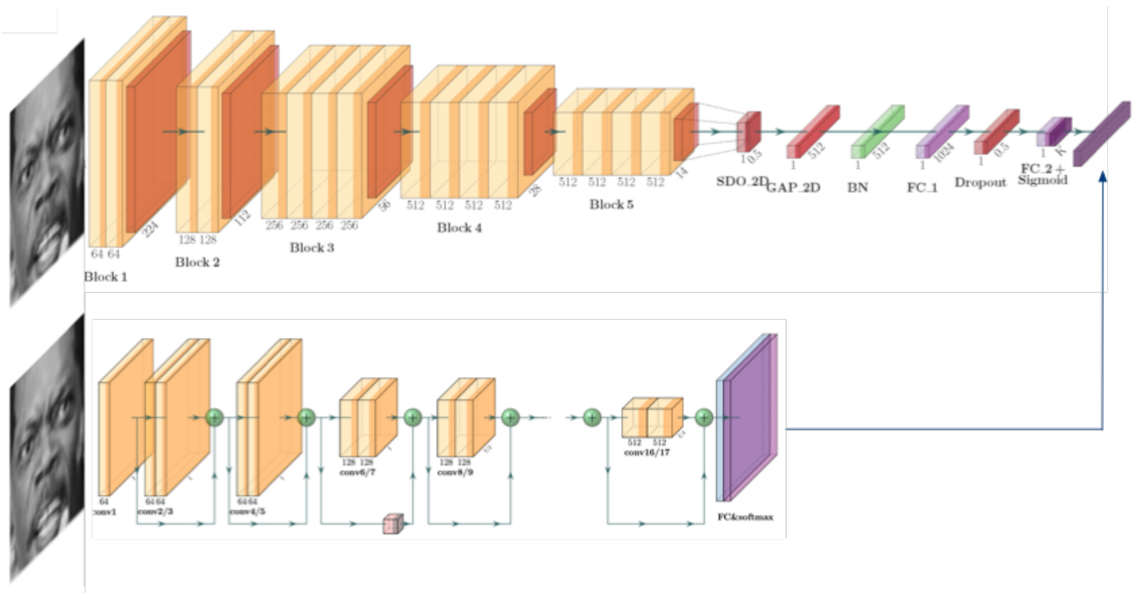


Figure 4.1: Ensemble between ResNet-18 and VGG19, with a Fully Connected Layer as output.

The same architecture was used in both scenarios: without occlusion and with occlusion. The chosen dataset was FER2013, composed of 28709 48x48 grayscale images for training and 3589 for testing, with seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). Since the ResNet-18 and VGG19 expect 224x224 RGB images in the input, it was necessary to scale the images up and replicate a single channel image to the remaining two (Figure 4.2).

Figure 4.2: Sample of the FER2013 dataset.

4.2 Introducing occlusion

To simulate the presence of VR goggles, it was necessary to hide the upper part of the face, namelly, the eyes. Considering that the face can assume different tilt and yaw positions, the relative position of the eyes change, so the algorithm to calculate the goggles position should contemplate this. To achieve this, it was necessary to obtain the location of the face as well as facial landmarks, composed of the position of the eyes, nose and mouth. For that, Multi-task Cascade Convolutional Neural Networks (MTCNN) were used [134, 139]. It consists essentially of 3 parts:

1. A network of proposals (P-NET - Figure 4.3) that foresees potential face positions and their bounding boxes. This process results in a large number of facial detections, many of which are false;

2. A refined network (R-Net - Figure 4.4) which makes use of the result from step i), thereby refining the result to eliminate most false detection and limiting aggregates;

3. A network similar to the one used in step ii), called O-Net - Figure 4.5, further refines the forecasts and adds facials forecasts to the implementation of MTCNN.

With the position of the eyes, the algorithm starts by calculating the middle point between the eyes, and the distance between them (Algorithm 1). It continues by estimating the width and height of the goggles as 20% larger than the distance between the eyes and 150% the distance between the eye line and the nose. The tilt angle is also calculated and, with these, the rectangle is

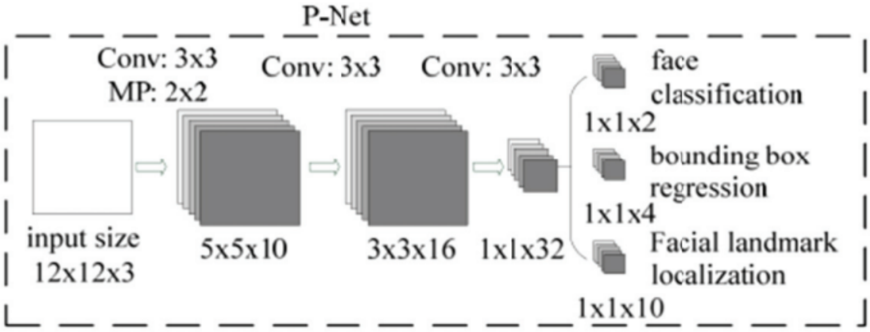


Figure 4.3: P-Net Structure [60].

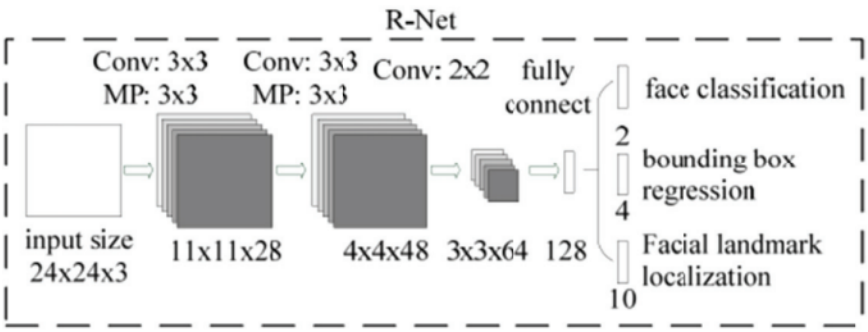


Figure 4.4: R-Net Structure [60].

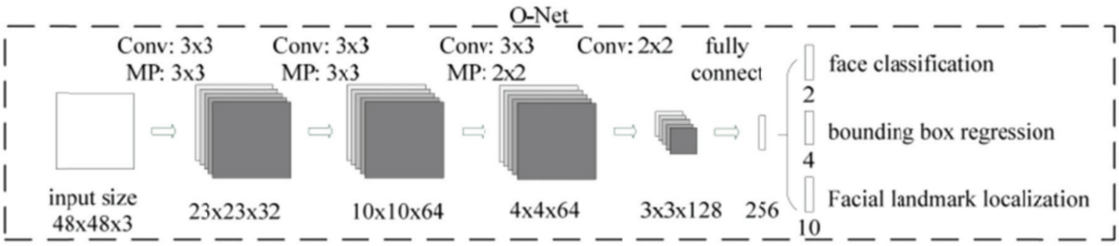


Figure 4.5: O-Net Structure [60].

drawn in gray on top of the *sample* image. When the landmarks is empty, meaning that the facial features could not be found, the goggles are not drawn (Some examples are shown in Figure 4.6).

Algorithm 1 Occlusion algorithm.

procedure MAKEGOGGLES(sample, landmarks)

$left\_eye\_x, left\_eye\_y \leftarrow landmarks[0][0], landmarks[0][5]$

$right\_eye\_x, right\_eye\_y \leftarrow landmarks[0][1], landmarks[0][6]$

$nose\_x, nose\_y \leftarrow landmarks[0][2], landmarks[0][7]$

$middle\_x, middle\_y \leftarrow \frac{right\_eye\_x + left\_eye\_x}{2}, \frac{right\_eye\_y + left\_eye\_y}{2}$

$googles\_width = 2.2 * \sqrt{(right_eye_y - left_eye_y)^2 + (right_eye_x - left_eye_x)^2}$

$googles\_height = 1.5 * \sqrt{(middle_eyes_y - nose_y)^2 + (middle_eyes_x - nose_x)^2}$

$rectangle = (0, 0, googles\_width, googles\_height)$

$middle\_rectangle\_x, middle\_rectangle\_y = \frac{googles_width}{2}, \frac{googles_height}{2}$

$angle = \frac{right_eye_y - left_eye_y}{right_eye_x - left_eye_x} * \frac{180}{\pi}$

$rectangle = rectangle.rotate(-angle, (middle_rectangle_x, middle_rectangle_y))$

$final_size = rectangle.size$

$sample.paste(rectangle, \frac{middle_eyes_x - final_size[0]}{2}, \frac{middle_eyes_y - final_size[1]}{2})$

Figure 4.6: Sample of the FER2013 dataset with the occlusion algorithm.

The accuracy of the classification was measured through the confusion matrix and accuracy of each class.

Chapter 5

Experimental Setup

To be able to validate HAR and FER methods, it is crucial to test the methods in a dataset. Public datasets are the ideal solution to compare the efficacy of the actual method with state-of-the-art. Nevertheless, for specific applications the use of a personal dataset is enough, when a reasonable accuracy is reached. This chapter goes over the experimental setup used in this work, detailing each dataset used, the development of our own and also the configuration of the methods described in Chapters 3 and 4.

5.1 HAR Dataset

By using known datasets, it is possible to compare the accuracy of our method with the state-of-the-art. The main goal of this project is not to achieve higher performance than state-of-the-art, instead, this project focuses on the development of prompt and efficient methods, that will be further applied in our specific application. Since our own dataset is not finished, the N-UCLA Dataset [129], was used to test and validate the proposed HAR methodology.

5.1.1 VR-ACT Dataset

The VR-ACT is currently under development, being recorded at the Polytechnic Institute of Bragança (IPB). For the development of this dataset a set of several students of the Bachelor in Computer Science of IPB, is being used. The dataset will consist on a set of several actions, being performed while the student is using VR glasses and playing a very simple game. An example of the actions that will be recorded, is shown on Figure 5.1.

5.1.2 N-UCLA Dataset

Most HAR datasets contain videos recorded outdoors with a variety of different actions. The N-UCLA dataset [129] contains ten action categories: picking up with one hand, picking up with two hands, dropping trash, walking around, sitting down, standing up, donning, doffing, throwing, and

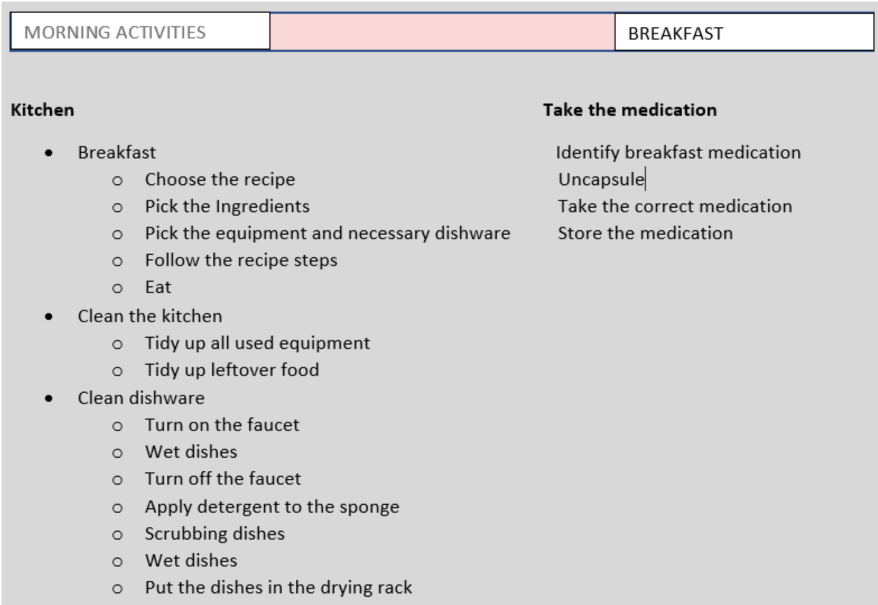


Figure 5.1: Samples of VR-ACT Dataset activities.

carrying (Figure 5.2). Each action was performed by ten actors. This dataset has the particularity that most videos are straight to the performing action, being very short.

5.2 Facial Expression Dataset

For the recognition of humans facial expression, the FER-2013 dataset was used, as it is a common choice to evaluate these approaches [26, 36, 89]. It is composed by 32298 images, belonging to a set of 7 different classes (Angry - 4953, Disgust - 547, Fear - 5121, Happy - 8989, Sad - 6077, Surprise - 4002, Neutral - 6198). Figure 5.3 shows an example of each of the classes.

5.3 HAR Setup

Following the methodology described in Chapter 3, this work proposes several different techniques to classify a action a human is performing:

1. OpenPose skeleton extraction, followed by the normalization of the real coordinates. The frame sequence extracted from each video was used as the input of the LSTM;

2. OpenPose skeleton extraction, then the selected sets of joints were used to compute the angles that they form. The sequence of frames extracted from each video was used as the input of the LSTM;

3. BlazePose skeleton extraction, followed by the normalization of the real coordinates. The frame sequence extracted from each video was used as the input of the LSTM;


(a) One hand pick.

(b) Two hand pick.

(c) Put in trash.

(d) Walking.

(e) Sitting Down.

(f) Standing up.

(g) Donning.

(h) Doffing.

(i) Throw.

(j) Carry.

Figure 5.2: N-UCLA Dataset - Class samples.


(a) Angry.

(b) Disgust.

(c) Fear.

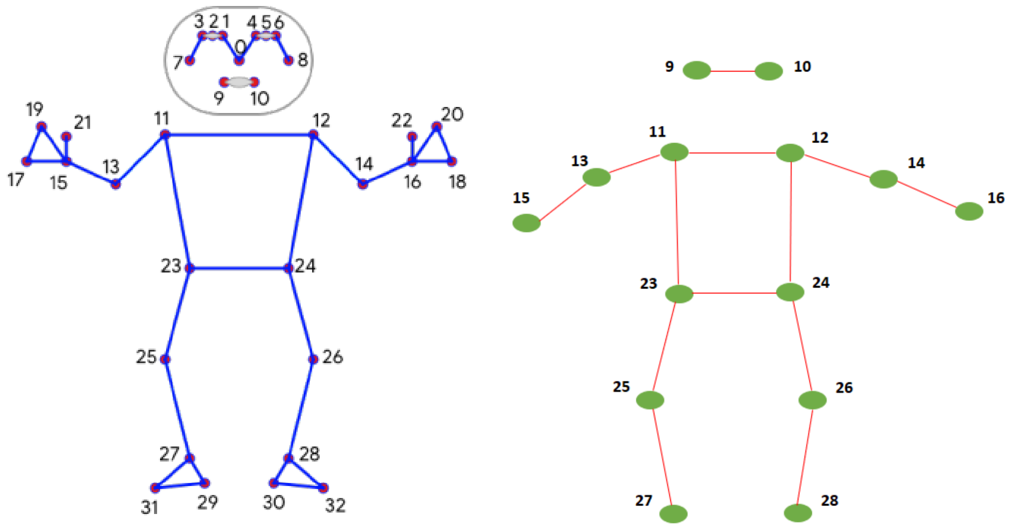
(d) Happy.

(e) Sad.

(f) Surprise.

(g) Neutral.

Figure 5.3: FER-2013 Dataset - Class samples.



(a) BlazePose full topology [9]. (b) BlazePose simplified topology.

Figure 5.4: Full and Partial topologies.

4. BlazePose skeleton extraction, followed by the computation of the MJD descriptor. The frame sequence extracted from each video was used as the input of the LSTM;

Since BlazePose's original skeleton has many keypoints, in order to try to keep the most relevant information about human motion, a simplification of this model (14 keypoints), was also done (Figure 5.4).

In this work, a very simple LSTM capable of learning and classifying data sequences, was used. To classify a given sequence as one of the ten possible actions of the N-UCLA dataset, it was only used an LSTM followed by a linear layer to extract the classification head. Some tests were made, with different hidden sizes for this RNN, including 48, 100, and 300 (different hidden sizes were tested although these sizes proved to be the best solution). To evaluate the performance of HAR methods, the F1 metric was used. The F1 metric is the harmonic mean of Precision and Recall, and it measures erroneously categorized instances accurately. When the class distribution is similar, accuracy can be useful, but when there are imbalanced classes, F1-score is a preferable metric. Also, regarding training purposes, all HAR methods used a learning rate of 0.001, a batch size of 16, the Adam optimizer, and a total of 300 epochs for training. For all HAR methods, a split of 80% for training and 20% for testing was used. Several different configurations were also tested, although this setup leaded to an optimal solution. Regarding the software, for this task, a AMD Ryzen Threadripper 3970X 32-Core Processor with an NVIDIA GeForce RTX 3090 with 64GB RAM, was used.

It should be noted that the graph embedding approach was not tested. With the extraction of the graphs, the next step was the embedding of the graphs. This is indeed possible, however Karateclub library [102], does not offer the require methods to do so, yet. The authors have been contacted, and the process of inferring a new, previously unseen graph embedding from a previous set of graphs is currently impossible. Although the authors invited us to participate in the development of a method capable of doing so. Doc2Vec [63] offers this possibility, although neither Graph2Vec or FeatherGraph, have this ability. This promising approach will be tested when it is implemented at Karateclub.

5.4 FER Setup

For the FER, two identical versions of the classification network were trained for 50 epochs with the mini batch of 64 on a AMD Ryzen Threadripper 3970X 32-Core Processor with an NVIDIA GeForce RTX 3090 with 64GB RAM.

The same training and validation datasets were used in both situations, although on the second, all the examples were changed to introduce an occlusion over the eyes. The training process happened in three steps: i) training of the ResNet18; ii) training of the VGG19; iii) training of the full assembly. Both CNNs were initialized with the pretrained ImageNet-1K weights, although all the parameters were allowed to change.

Chapter 6

Results and Discussion

This chapter shows the classification capacity of the proposed methodologies, to solve the proposed tasks: Action and Facial Recognition. This chapter also includes a comparison of our results with state-of-the-art as well as a discussion of the developed work.

6.1 Human Action Recognition

Table 6.1, shows all the considered methods and variations tested using the OpenPose algorithm.

Table 6.1: OpenPose - F1 score on N-UCLA dataset

LSTM	Keypoints Normalization	Angles
48	0.733	0.745
100	0.722	0.711
300	0.711	0.703

As Table 6.1 presents, the best result was obtained with a hidden size of 48, achieving an F1 score of 0.745.

Table 6.2, presents the tests performed using the BlazePose algorithm.

Table 6.2: BlazePose - F1 score on N-UCLA dataset

LSTM	Coordinates Normalization	MJD
48	0.584	0.476
100	0.536	0.511
300	0.557	0.444

Regarding MediaPipe's classification capability it can be seen by the analysis of Table 6.2, that this skeleton extraction method was not enough to surpass the methods using OpenPose.

Table 6.3, shows the F1 Score achieved using the simplified version of BlazePose skeleton (shown in Section 5.3).

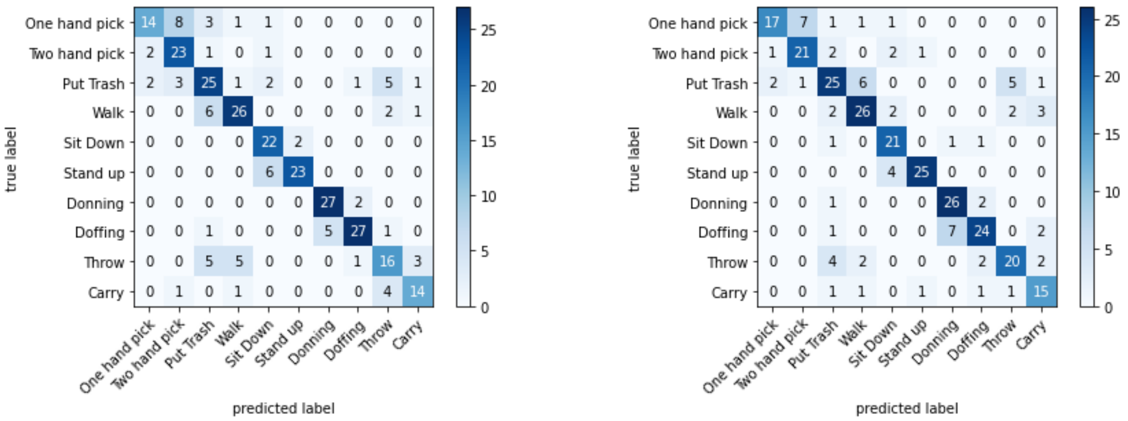
The F1-Score did not improve by using only 14 parts of BlazePose's detected skeleton, with 0.574 being the best F1 score.

Table 6.3: BlazePose with 14 keypoints - F1 score on N-UCLA dataset

LSTM	Coordinates Normalization	MJD
48	0.521	0.456
100	0.562	0.464
300	0.501	0.574

How it is possible to analyze from Tables 6.1, 6.2 and 6.3, OpenPose outperformed BlazePose by a wide margin. There is not a single approach using BlazePose that exceeds an F1 of 0.60, which is quite low for our application. Although more information about the extracted skeleton was expected to improve accuracy, OpenPose proved to be much more robust and consistent, achieving F1 scores higher than 0.70.

To better understand the best model's recognition capability, Figure 6.1 shows the prediction and true labels of the best methods using the OpenPose skeleton.



(a) Keypoints Normalization. (b) Angles.

Figure 6.1: Confusion matrix of the best methods - OpenPose.

By comparing the confusion matrix of the Keypoints Normalization method (Figure 6.1a) with the Angles method (Figure 6.1b), it is possible to conclude that distinct and different classification abilities were achieved by using the same method for extracting the skeleton, but using a different normalization method. Although it should be noted that due to the similar characteristics of these classes, both configurations struggle to predict 'donning' and 'doffing,' and the same can be verified for 'sit down' and 'stand up', 'one hand pick' and 'two hand pick', and also 'walk' and 'put trash', since in both activities the walking action is performed.

In order to compare our approach with state-of-the-art, Table 6.4 illustrates the accuracy achieved by several methods using N-UCLA dataset. As most state-of-the-art approaches use accuracy instead of F1-score, this table also presents the accuracy achieved by our best solution.

By the analysis of Table 6.4, we can conclude that our method is a way far of achieving state-of-the-art results. Most of these studies rely on the use of Graph Convolutional Networks (GCNs), which work in a similar way to our graph embedding approach, but this approach could not be

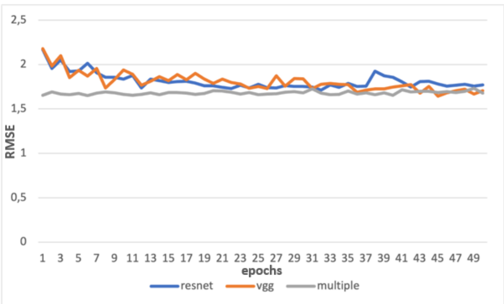
Table 6.4: Comparison of proposed approach with state-of-the-art methods on N-UCLA dataset

Methods	Accuracy	F1-Score
HBRNN-L [32]	0.805	
Glimpse Clouds [6]	0.876	
GCN–HCRF [70]	0.915	
VE-GCN [69]	0.918	
AGC-LSTM [111]	0.933	
CTR-GCN [21]	0.965	
LST [135]	0.972	
Our method	0.746	0.745

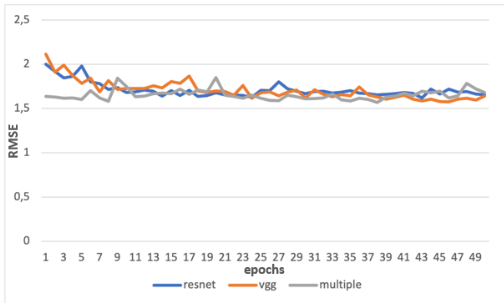
validated due to a lack of implemented libraries. It should be also noted that GCNs has also some limitations that leaded to the use of the LSTM network, for example, robustness to coordinates, interoperability with other inputs and scalability to multi-person.

6.2 Facial Expression Recognition

A division of 80% for training and 20% for testing, was used. In the training process, there is a slightly better result with the no occlusion dataset, as expected, since we loose part of the information (Figure 6.2).



(a) Occlusion.

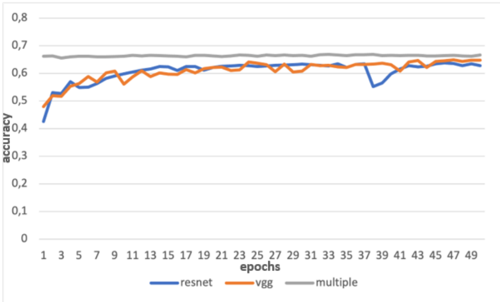


(b) No occlusion.

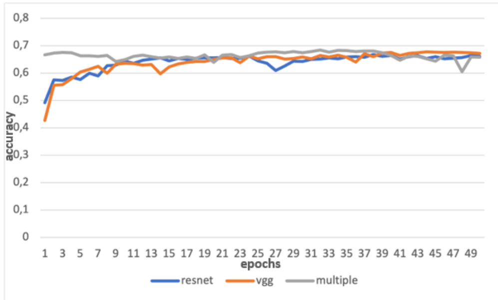
Figure 6.2: Evolution of RMSE during training.

The accuracy also improves with the epochs (Figure 6.3). In the occlusion dataset (Figure 6.3a) the highest accuracy is obtained with the multiple model. However, in the no occlusion situation (Figure 6.3b) it is the lowest. Looking at the progress during the epochs, it seems that the increase in the number of epochs would achieve better accuracy.

To illustrate the classification capacity of the three models in predicting the seven facial expressions in the FER-2013 dataset, confusion matrices were built comparing the situation between occlusion and no occlusion datasets (Figures 6.4, 6.5 and 6.6). The true labels correspond to the labels displayed on the left side of the figures and the predicted ones, to the labels displayed on the lower side.

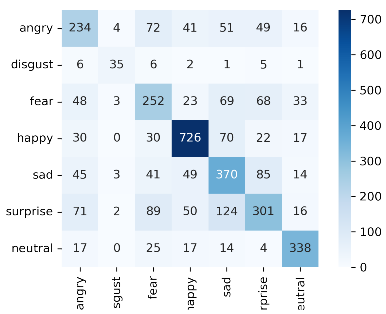


(a) Occlusion.

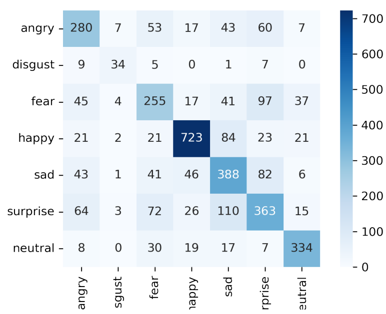


(b) No occlusion.

Figure 6.3: Evolution of the accuracy during training.



(a) Occlusion.

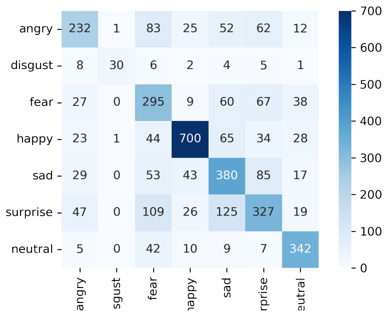


(b) No occlusion.

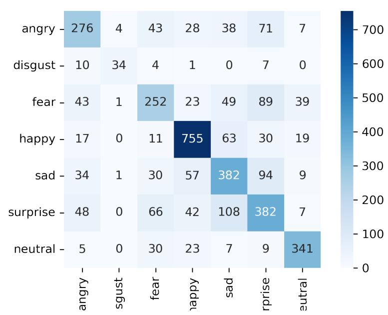
Figure 6.4: Confusion matrix for the ResNet18.

As Figure 6.4a displays, the angry class presents 72 wrong classifications that belong to fear class, 51 as sad and 49 as surprise. The fear class has 69 wrong classifications of sad class and 68 of surprise. The class sad also has 85 wrong classifications of examples of surprise class and finally the class surprise presents 124 wrong classifications in sad class.

These values can be also verified in Figure 6.4b, highlighting in general, values a little bit superior of correct classifications. For example, while the angry class, in the confusion matrix without occlusion, presents 234 classifications performed correctly, in the confusion matrix with no occlusion, it presents 280 of correct classifications.



(a) Occlusion.



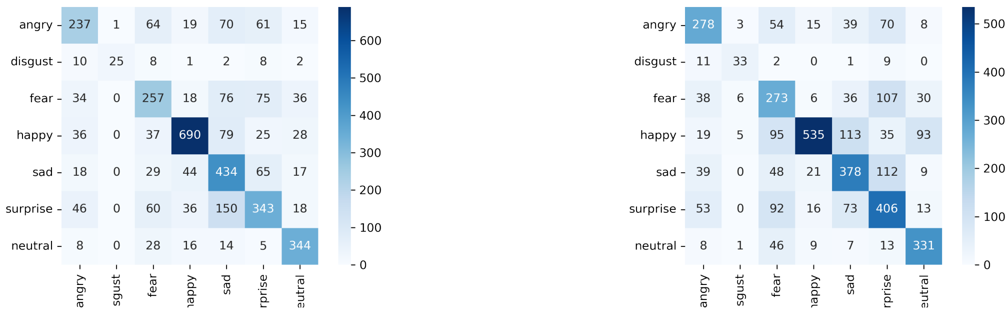
(b) No occlusion.

Figure 6.5: Confusion matrix for the VGG19.

In the confusion matrix with occlusion, of the VGG-19 network (Figure 6.5a), there is the same tendency of classifications performed incorrectly and in the same classes of the matrix in Figure 6.4a. It is also noteworthy that, in the aforementioned classes, compared to the confusion matrix with occlusion from Resnet18, this matrix presents higher values of higher accurate classifications in angry (232), sad (380) and surprise (327) and lower in the fear class (295).

The matrix represented in Figure 6.5b also follows the correct classifications of the matrix in Figure 6.4b. It can be observed that, in relation to the confusion matrix without occlusion, the classifications performed correctly, in general, is superior. For example, the angry class presents 276 correct classifications in the matrix with occlusion, while in the matrix without occlusion it shows 232 of correct classified samples. It contains superior values of true classifications, within the same 4 classes already mentioned and in relation to the same matrix of the Resnet18 network, in classes fear (276) and surprise (382) and lower values in the remaining two classes.

In the confusion matrix represented in Figure 6.6a, it is possible to verify the same situation mentioned in the confusion matrices of Figures 6.4a and 6.5a. In the confusion matrix represented in Figure 6.6b, there is a general increase in correctly performed classifications, in relation to the confusion matrix of the model combined with no occlusion (for example, in the angry class, there is an increase from 237 ratings to 278 ratings). When compared with the matrices of the networks without occlusion, mentioned above, there is a decrease in classifications performed correctly in the sad class.



(a) Occlusion. (b) No occlusion.

Figure 6.6: Confusion matrix for the combined model.

The classes are not balanced so, intra-class normalization was also performed (Table 6.5). The impact of occlusion of the eyes is, apparently, marginal. The highest F1-Score in both situations (occlusions and no occlusion) is in the class ‘neutral’, immediately followed by ‘happy’ in the occlusion and ‘sad’/‘surprise’ for the no occlusion. The lowest F1-Score was in the classes ‘disgust’ and ‘fear’, respectively. According to the previous confusion matrices, the ‘disgust’ class is often misclassified as ‘angry’ or ‘surprise’.

The overall F1-Score of the combined model was 0.649 for the occlusion and 0.628 for the no occlusion, which is far from the state of the art result. Nevertheless, the purpose of this work was to assess the impact of occlusion and the results confirm that most of the classification is performed with the mouth and chin. Note also that our method presents very low accuracy in some classes comparing to others. For example, in the fear class the accuracy is very low, although in the neutral class, the results are a way better. This is a consequence of the imbalanced classes of this dataset. This issue can be overcomed by balancing this classes, based on data augmentation. Although, the primary goal of this study, was to evaluate the impact of the occlusion of VR glasses in FER.

The chosen architecture combines two different models with different internal organization and dimension. We hope that each performs better in different aspects, each can lead to an overall better result. The overall result is obtained in a fully connected layer that has input the results of both.

To compare our best method with state-of-the-art, it is only possible to compare the F1-Score of the best method without occlusion (Table 6.6), since there are no studies that performed a similar occlusion and used the FER-2013 dataset. [37] is the most similar study to our FER module, though they used the FER+ dataset [7], achieving a result of 0.828 accuracy, with occlusion of part of the face. The FER+ dataset is a new version of the FER-2013 dataset that was relabeled and added a new class of emotion, "contempt," while maintaining the other 7 classes of FER-2013. It is worth noting that, according to recent studies [38, 79, 142], it is possible to extract an average of 0.153 of improved classification in the FER+ dataset.

As shown in Table 6.6, our method was not able to surpass most of the methods. This was not our primary goal, which was to study and interpret the impact of the occlusion; in future work,

Table 6.5: F1-Score per class

(a) Occlusion

Class	ResNet18	VGG19	Combined
angry	0,501	0,497	**0,508**
disgust	**0,625**	0,536	0,446
fear	0,508	**0,595**	0,518
happy	**0,811**	0,782	0,771
sad	0,61	0,626	**0,715**
surprise	0,461	0,501	**0,525**
neutral	0,814	0,824	**0,829**
F1-Score	0,627	0,645	**0,649**

(b) No occlusion

Class	ResNet18	VGG19	Combined
angry	**0,6**	0,591	0,595
disgust	**0,607**	**0,607**	0,589
fear	0,514	0,508	**0,55**
happy	0,808	**0,844**	0,598
sad	**0,639**	0,629	0,623
surprise	0,556	0,585	**0,623**
neutral	0,805	**0,822**	0,798
F1-Score	0,663	**0,673**	0,628

Table 6.6: Comparison of proposed approach with state-of-the-art methods on FER 2013 dataset

Methods	Accuracy	F1-Score
VCNN [1]	0,657	
DeepEmotion [81]	0,700	
CNN-MNF [64]	0,703	
EXNET [99]	0,735	
LHC-Net [86]	0,744	
CNNs and BOVW + local SVM [38]	0.754	
GLFCNN+SVM [117]	0,844	
Our method	0.675	0,673

the improvement of this accuracy will be the focus, with the goal of achieving competitive results with the state-of-the-art. Note that by improving our method without any occlusion, will certainly lead to a better capacity of classification under occlusion.

Chapter 7

Conclusions and Future Work

The present dissertation aimed at the development of two distinct computer visions tasks, that will be important components of a DDA system. To achieve this, several methods to perform HAR and the recognition of humans facial expression, were developed.

For the FER task, this work presents the development of 3 CNNs, in order to be able to make comparisons between them, in order to select the network that obtained the best values. Regarding the networks that were trained with the original dataset (FER-2013), F1-score of 0.663 was obtained for ResNet 18, 0.673 for VGG19 and the combination of these networks has resulted in a F1-score of 0.628. When trained with the modified dataset (dataset FER-2013 to which the occlusion algorithm was applied), ResNet18 obtained a F1-Score of 0.627, VGG19 of 0.645 and 0.649 for the combined model. Thus, the model selected for this work was the combined model (with occlusion). The results obtained are below expectations (since, when compared to the state of the art, a lower classification accuracy was achieved). Regarding the occlusion algorithm, it does not correctly perform the occlusion in some occasions. Sometimes the MTCNN does not find the facial features, which means that the landmarks are empty and therefore occlusion is not performed. It should be noted that the development of FER methods described in this dissertation, has already resulted in an accepted publication [100].

Regarding HAR, this dissertation proposes several HAR recognition methods, with the goal of incorporating the best method into a future application. A skeleton based approach was followed, using one algorithm to extract 2D information (OpenPose) and also other algorithm to extract 3D information of the detected human skeleton (BlazePose). The keypoints of the detected skeletons were extracted, using these two methods and then the extracted coordinates were processed using several techniques, including coordinate normalization, the use of the MJD descriptor, and the computation of the angles between joints.

Our results did not achieve state-of-the-art results, but that was not our primary goal, which was to develop simple and fast real-time techniques that allowed understanding the challenges and possibilities to address HAR. OpenPose (2D) proved to be much more robust than BlazePose (3D), achieving a reasonable F1-Score of 0.745 on the N-UCLA dataset.

A graph embedding approach was also developed, altough KarateClub (a Python graph library) did not support the embedding of previously unseen graphs. In the future, it is expected our collaboration with the KarateClub authors, to develop a method capable of doing it. The acquisition of our dataset was already been started, with the goal of creating a dataset that can be applied to a final application, which will consist of a VR game.

The work described in this dissertation will be improved in the context of the GreenHealth project [74], integrated in a Phd plan, by doing the follow future work:

- Finalize our dataset, with humans using VR glasses and performing daily basis actions;
- Test different methods to perform the recognition of humans facial expression;
- Achieve better results by creating new versions of the skeleton, like the 14-joint skeleton;
- Using different Deep Learning networks to classify the actions, like image-based methods and the combination of them with skeletal information.

References

[1] Abhinav Agrawal and Namita Mittal. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. *The Visual Computer*, 36(2):405–412, February 2020.

[2] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The MUG facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010.

[3] José Almeida and Fátima Rodrigues. Facial Expression Recognition System for Stress Detection with Deep Learning:. In *Proceedings of the 23rd International Conference on Enterprise Information Systems*, pages 256–263, Online Streaming, — Select a Country —, 2021. SCITEPRESS - Science and Technology Publications.

[4] Kleber de O. Andrade, Guilherme Fernandes, Glauco A.P. Caurin, Adriano A.G. Siqueira, Roseli A.F. Romero, and Rogerio de L. Pereira. Dynamic Player Modelling in Serious Games Applied to Rehabilitation Robotics. In *2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol*, pages 211–216, October 2014.

[5] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. OpenFace: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, Lake Placid, NY, USA, March 2016. IEEE.

[6] Fabien Baradel, Christian Wolf, Julien Mille, and Graham W. Taylor. Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points, August 2018. arXiv:1802.07898 [cs].

[7] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 279–283, Tokyo Japan, October 2016. ACM.

[8] M.S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 1, pages 592–597, The Hague, Netherlands, 2004. IEEE.

[9] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. BlazePose: On-device Real-time Body Pose tracking, June 2020. arXiv:2006.10204 [cs].

[10] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, November 2020.

[11] M. Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. *Action as space-time shapes*, volume 29. November 2005. Journal Abbreviation: Pattern Analysis and Machine Intelligence, IEEE Transactions on Pages: 1402 Vol. 2 Publication Title: Pattern Analysis and Machine Intelligence, IEEE Transactions on.

[12] Paris Mavromoustakos Blom, Sander Bakkes, and Pieter Spronck. Modeling and adjusting in-game difficulty based on facial expression analysis. *Entertainment Computing*, 31:100307, August 2019.

[13] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science*, 1(3):219–237, November 2008.

[14] Oana Bălan, Gabriela Moise, Alin Moldoveanu, Marius Leordeanu, and Florica Moldoveanu. An Investigation of Various Machine and Deep Learning Techniques Applied in Automatic Fear Level Detection and Acrophobia Virtual Therapy. *Sensors*, 20(2):496, January 2020. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Oana Bălan, Alin Moldoveanu, and Marius Leordeanu. A Machine Learning Approach to Automatic Phobia Therapy with Virtual Reality. In Ioan Opris, Mikhail A. Lebedev, and Manuel F. Casanova, editors, *Modern Approaches to Augmentation of Brain Function*, Contemporary Clinical Neuroscience, pages 607–636. Springer International Publishing, Cham, 2021.

[16] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, January 2021. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, Honolulu, HI, July 2017. IEEE.

[18] Júlio Castro Lopes and Rui Pedro Lopes. Dynamic difficulty adjustment in serious games for rehabilitation. In *Submited to: 2022 IEEE 10th International Conference on Serious Games and Applications for Health (SeGAH)*, 2022.

[19] Guillaume Chanel and Phil Lopes. User Evaluation of Affective Dynamic Difficulty Adjustment Based on Physiological Deep Learning. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition. Theoretical and Technological Approaches*, Lecture Notes in Computer Science, pages 3–23, Cham, 2020. Springer International Publishing.

[20] Chen Chen, Roozbeh Jafari, and Nasser 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)*, pages 168–172, September 2015.

[21] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition, August 2021. arXiv:2107.12213 [cs].

[22] Guangchun Cheng, Yiwen Wan, Abdullah N. Saudagar, Kamesh Namuduri, and Bill P. Buckles. Advances in Human Action Recognition: A Survey, January 2015. arXiv:1501.05964 [cs].

[23] Yue Cheng, Bin Jiang, and Kebin Jia. A Deep Structure for Facial Expression Recognition under Partial Occlusion. In *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 211–214, 2014.

[24] Jin Choi, Yong-il Cho, Kyusung Cho, Su-jung Bae, and Hyun S. Yang. A View-based Multiple Objects Tracking and Human Action Recognition for Interactive Virtual Environments.

[25] Kuang-Pen Chou, Mukesh Prasad, Di Wu, Nabin Sharma, Dong-Lin Li, Yu-Feng Lin, Michael Blumenstein, Wen-Chieh Lin, and Chin-Teng Lin. Robust Feature-Based Automated Multi-View Human Action Recognition System. *IEEE Access*, 6:15283–15296, 2018. Conference Name: IEEE Access.

[26] Tee Connie, Mundher Al-Shabi, Wooi Ping Cheah, and Michael Goh. Facial Expression Recognition Using a Hybrid CNN–SIFT Aggregator. In Somnuk Phon-Amnuaisuk, Swee-Peng Ang, and Soo-Young Lee, editors, *Multi-disciplinary Trends in Artificial Intelligence*, Lecture Notes in Computer Science, pages 139–149, Cham, 2017. Springer International Publishing.

[27] Mihaly Csikszentmihalyi. Flow: The Psychology of Optimal Experience. January 1990.

[28] 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)*, volume 1, pages 886–893 vol. 1, June 2005. ISSN: 1063-6919.

[29] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. ISSN: 1063-6919.

[30] Terrance Devries, Kumar Biswaranjan, and Graham W. Taylor. Multi-task Learning of Facial Landmarks and Expression. In *2014 Canadian Conference on Computer and Robot Vision*, pages 98–103, Montreal, QC, Canada, May 2014. IEEE.

[31] Melissa Donaldson. Plutchik's wheel of emotions—2017. Update, 2017.

[32] Yong Du, Yun Fu, and Liang Wang. Representation Learning of Temporal Dynamics for Skeleton-Based Action Recognition. *IEEE Transactions on Image Processing*, 25(7):3010–3022, July 2016. Conference Name: IEEE Transactions on Image Processing.

[33] Abassin Fangbemi, Bin Liu, Neng Yu, and Yanxiang Zhang. Efficient Human Action Recognition Interface for Augmented and Virtual Reality Applications Based on Binary Descriptor: 5th International Conference, AVR 2018, Otranto, Italy, June 24–27, 2018, Proceedings, Part I. pages 252–260. July 2018.

[34] Fei Cheng, Jiangsheng Yu, and Huilin Xiong. Facial Expression Recognition in JAFFE Dataset Based on Gaussian Process Classification. *IEEE Transactions on Neural Networks*, 21(10):1685–1690, October 2010.

[35] Daniel Gabana, Laurissa Tokarchuk, Emily Hannon, and Hatice Gunes. Effects of valence and arousal on working memory performance in virtual reality gaming. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 36–41, October 2017. ISSN: 2156-8111.

[36] Yanling Gan, Jingying Chen, and Luhui Xu. Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognition Letters*, 125:105–112, July 2019.

[37] Mariana-Iuliana Georgescu and Radu Tudor Ionescu. Teacher-Student Training and Triplet Loss for Facial Expression Recognition under Occlusion, February 2021. arXiv:2008.01003 [cs].

[38] Mariana-Iuliana Georgescu, Radu Tudor Ionescu, and Marius Popescu. Local Learning with Deep and Handcrafted Features for Facial Expression Recognition. *IEEE Access*, 7:64827–64836, 2019. arXiv:1804.10892 [cs].

[39] Felix A Gers, Nicol N Schraudolph, and Jurgen Schmidhuber. Learning Precise Timing with LSTM Recurrent Networks. page 29, 2002.

[40] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64:59–63, 2015.

[41] Maurits Graafland and Marlies Schijven. How Serious Games Will Improve Healthcare. pages 139–157. January 2018.

[42] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, May 2010.

[43] Ritam Guha, Ali Hussain Khan, Pawan Kumar Singh, Ram Sarkar, and Debotosh Bhattacharjee. CGA: a new feature selection model for visual human action recognition. *Neural Computing and Applications*, 33(10):5267–5286, May 2021.

[44] Nadia Hocine, Abdelkader Gouaïch, and Stefano A. Cerri. Dynamic Difficulty Adaptation in Serious Games for Motor Rehabilitation. In Stefan Göbel and Josef Wiemeyer, editors, *Games for Training, Education, Health and Sports*, Lecture Notes in Computer Science, pages 115–128, Cham, 2014. Springer International Publishing.

[45] Bita Houshmand and Naimul Mefraz Khan. Facial Expression Recognition Under Partial Occlusion from Virtual Reality Headsets based on Transfer Learning. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 70–75, New Delhi, India, September 2020. IEEE.

[46] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. page 9, 2015.

[47] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[48] Dong Huang, Shitong Yao, Yi Wang, and Fernando De La Torre. Sequential Max-Margin Event Detectors. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8691, pages 410–424. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science.

[49] Tobias Huber, Silvan Mertes, Stanislava Rangelova, Simon Flutura, and Elisabeth André. Dynamic Difficulty Adjustment in Virtual Reality Exergames through Experience-driven Procedural Content Generation. *arXiv:2108.08762 [cs]*, August 2021. arXiv: 2108.08762.

[50] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. *arXiv:1612.01925 [cs]*, December 2016. arXiv: 1612.01925.

[51] Neziha Jaouedi, Noureddine Boujnah, and Med Salim Bouhlel. A new hybrid deep learning model for human action recognition. *Journal of King Saud University - Computer and Information Sciences*, 32(4):447–453, May 2020.

[52] Xiaopeng Ji, Qingsong Zhao, Jun Cheng, and Chenfei Ma. Exploiting spatio-temporal representation for 3D human action recognition from depth map sequences. *Knowledge-Based Systems*, 227:107040, September 2021.

[53] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv:1408.5093 [cs]*, June 2014. arXiv: 1408.5093.

[54] Aouaidjia Kamel, Bin Sheng, Po Yang, Ping Li, Ruimin Shen, and David Dagan Feng. Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(9):1806–1819, September 2019. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[55] T. Kanade, J.F. Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 46–53, Grenoble, France, 2000. IEEE Comput. Soc.

[56] Pushpajit Khaire, Praveen Kumar, and Javed Imran. Combining CNN streams of RGB-D and skeletal data for human activity recognition. *Pattern Recognition Letters*, 115:107–116, November 2018.

[57] D. Kim, Woo-han Yun, Ho-Sub Yoon, and J. Kim. Action recogntion with depth maps using hog descriptors of multi-view motion appearance and history. *UBICOMM 2014 - 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 126–130, 01 2014.

[58] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990. Conference Name: Proceedings of the IEEE.

[59] Yu Kong and Yun Fu. Human Action Recognition and Prediction: A Survey. *arXiv:1806.11230 [cs]*, February 2022. arXiv: 1806.11230.

[60] Hongchang Ku and Wei Dong. Face recognition based on mtcnn and convolutional neural network. *Frontiers in Signal Processing*, 4(1):37–42, 2020.

[61] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, November 2011. ISSN: 2380-7504.

[62] Beom Kwon, Junghwan Kim, Kyoungoh Lee, Yang Koo Lee, Sangjoon Park, and Sanghoon Lee. Implementation of a Virtual Training Simulator Based on 360° Multi-View Human Action Recognition. *IEEE Access*, 5:12496–12511, 2017. Conference Name: IEEE Access.

[63] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[64] Chao Li, Ning Ma, and Yalin Deng. Multi-Network Fusion Based on CNN for Facial Expression Recognition. pages 166–169. Atlantis Press, February 2018. ISSN: 2352-538X.

[65] Ruixin Li, Yan Liang, Xiaojian Liu, Bingbing Wang, Wenxin Huang, Zhaoxin Cai, Yaoguang Ye, Lina Qiu, and Jiahui Pan. MindLink-Eumpy: An Open-Source Python Toolbox for Multimodal Emotion Recognition. *Frontiers in Human Neuroscience*, 15:621493, February 2021.

[66] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–14, June 2010. ISSN: 2160-7516.

[67] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, and Shuo Chen. Dynamic Difficulty Adjustment in Computer Games Through Real-Time Anxiety-Based Affective Feedback. *International Journal of Human–Computer Interaction*, 25(6):506–529, August 2009. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447310902963944.

[68] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, June 2009. ISSN: 1063-6919.

[69] Kai Liu, Lei Gao, Naimul Mefraz Khan, Lin Qi, and Ling Guan. Integrating vertex and edge features with Graph Convolutional Networks for skeleton-based action recognition. *Neurocomputing*, 466:190–201, November 2021.

[70] Kai Liu, Lei Gao, Naimul Mefraz Khan, Lin Qi, and Ling Guan. Graph Convolutional Networks-Hidden Conditional Random Field Model for Skeleton-Based Action Recognition. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 25–256, December 2019.

[71] Christos P. Loizou. An automated integrated speech and face imageanalysis system for the identification of human emotions. *Speech Communication*, 130:15–26, June 2021.

[72] Júlio Castro Lopes and Rui Pedro Lopes. A Review of Dynamic Difficulty Adjustment Methods for Serious Games. In Ana I. Pereira, Florbela P. Fernandes, João P. Coelho, João P. Teixeira, Maria F. Pacheco, Paulo Alves, and Rui P. Lopes, editors, *Optimization, Learning Algorithms and Applications*. Springer International Publishing, Cham, 2022.

[73] Rui Lopes. An Award System for Gamification in Higher Education. In *Conference: 7th International Conference of Education, Research and Innovation*, pages 5563–5573, January 2014.

[74] Rui Pedro Lopes, Bárbara Barroso, Leonel Deusdado, André Novo, Manuel Guimarães, João Paulo Teixeira, and Paulo Leitão. Digital Technologies for Innovative Mental Health Rehabilitation. *Electronics*, 10(18):2260, September 2021.

[75] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pages 94–101. IEEE, 2010.

[76] Zaosheng Ma. Human Action Recognition in Smart Cultural Tourism Based on Fusion Techniques of Virtual Reality and SOM Neural Network. *Computational Intelligence and Neuroscience*, 2021:e3495203, December 2021. Publisher: Hindawi.

[77] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, Miami, FL, June 2009. IEEE.

[78] Albert Mehrabian. Communication Without Words. In C. David Mortensen, editor, *Communication Theory*, pages 193–200. Routledge, 2 edition, September 2017.

[79] Si Miao, Haoyu Xu, Zhenqi Han, and Yongxin Zhu. Recognizing Facial Expressions Using a Shallow Convolutional Neural Network. *IEEE Access*, PP:1–1, June 2019.

[80] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality, October 2013. arXiv:1310.4546 [cs, stat].

[81] Shervin Minaee and Amirali Abdolrashidi. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network, February 2019. arXiv:1902.01019 [cs].

[82] Sarah Mroz, Natalie Baddour, Connor McGuirk, Pascale Juneau, Albert Tu, Kevin Cheung, and Edward Lemaire. Comparing the Quality of Human Pose Estimation with BlazePose or OpenPose. In *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*, pages 1–4, December 2021.

[83] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning Distributed Representations of Graphs, July 2017. arXiv:1707.05005 [cs].

[84] Preksha Pareek and Ankit Thakkar. A survey on video-based Human Action Recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, 54(3):2259–2322, March 2021.

[85] Sang-yong Park, Hanmoi Sim, and Wonhyung Lee. Dynamic Game Difficulty Control by Using EEG-based Emotion Recognition. *International Journal of Control and Automation*, 2014.

[86] Roberto Pecoraro, Valerio Basile, Viviana Bono, and Sara Gallo. Local Multi-Head Channel Self-Attention for Facial Expression Recognition, November 2021. arXiv:2111.07224 [cs] version: 2.

[87] Manuel Pezzera and N. Alberto Borghese. Dynamic difficulty adjustment in exer-games for rehabilitation: a mixed approach. In *2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–7, August 2020. ISSN: 2573-3060.

[88] Joana F. Pinto, Henrique R. Carvalho, Goncalo R. R. Chambel, João Ramiro, and Afonso Goncalves. Adaptive gameplay and difficulty adjustment in a gamified upper-limb rehabilitation. In *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8, May 2018. ISSN: 2573-3060.

[89] Tanusree Podder, Diptendu Bhattacharya, and Abhishek Majumdar. Time efficient real time facial expression recognition with CNN and transfer learning. *Sādhanā*, 47(3):177, August 2022.

[90] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[91] Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances. *IEEE Access*, 7:100943–100953, 2019.

[92] Phillip Prodger. *Darwin's camera: Art and photography in the theory of evolution*. Oxford University Press, 2009.

[93] Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. Histogram of Oriented Principal Components for Cross-View Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2430–2443, December 2016. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[94] Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2430–2443, 2016.

[95] Jadisha Yarif Ramirez Cornejo and Helio Pedrini. Emotion Recognition from Occluded Facial Expressions Using Weber Local Descriptor. In *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5, Maribor, Slovenia, June 2018. IEEE.

[96] Aishrith Rao. Efficient Min-Cost Real Time Action Recognition using Pose Estimates. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–6, Bangluru, India, November 2020. IEEE.

[97] Kishore K. Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, July 2013.

[98] Luz Rello, Clara Bayarri, and Azuki Gorriz. What is wrong with this word? dyseggxia: a game for children with dyslexia. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, October 2012.

[99] Muhammad Naveed Riaz, Yao Shen, Muhammad Sohail, and Minyi Guo. eXnet: An Efficient Approach for Emotion Recognition in the Wild. *Sensors*, 20(4):1087, January 2020. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[100] Ana Sofia Figueiredo Rodrigues, Júlio Castro Lopes, and Rui Pedro Lopes. Classification of facial expressions under partial occlusion for VR games. In Ana I. Pereira, Florbela P. Fernandes, João P. Coelho, João P. Teixeira, Maria F. Pacheco, Paulo Alves, and Rui P. Lopes, editors, *Optimization, Learning Algorithms and Applications*. Springer International Publishing, Cham, 2022.

[101] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[102] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs, August 2020. arXiv:2003.04819 [cs, stat].

[103] Benedek Rozemberczki and Rik Sarkar. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models, August 2020. arXiv:2005.07959 [cs, stat].

[104] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 397–403, Sydney, Australia, December 2013. IEEE.

[105] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, June 2018. ISSN: 2575-7075.

[106] Arya Sarkar, Avinandan Banerjee, Pawan Kumar Singh, and Ram Sarkar. 3D Human Action Recognition: Through the eyes of researchers. *Expert Systems with Applications*, 193:116424, May 2022.

[107] S. Saurav, A.K. Saini, R. Saini, and S. Singh. Deep learning inspired intelligent embedded system for haptic rendering of facial emotions to the blind. *Neural Computing and Applications*, 34(6):4595–4623, 2022. Publisher: Springer Science and Business Media Deutschland GmbH.

[108] Yoones A. Sekhavat. MPRL: Multiple-Periodic Reinforcement Learning for difficulty adjustment in rehabilitation games. In *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–7, April 2017.

[109] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis, April 2016. arXiv:1604.02808 [cs].

[110] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[111] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition, March 2019. arXiv:1902.09130 [cs].

[112] S. Singh, A. Gupta, and R.S. Pavithr. Automatic Classroom Monitoring System Using Facial Expression Recognition. *Lecture Notes in Electrical Engineering*, 836:151–165, 2022. ISBN: 9789811685415 Publisher: Springer Science and Business Media Deutschland GmbH.

[113] Jan Smeddinck, Sandra Siegel, and Marc Herrlich. Adaptive difficulty in exergames for Parkinson's disease patients. In *Proceedings of Graphics Interface 2013*, GI '13, pages 141–148, CAN, May 2013. Canadian Information Processing Society.

[114] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv:1212.0402 [cs]*, December 2012. arXiv: 1212.0402.

[115] Adi Stein, Yair Yotam, Rami Puzis, Guy Shani, and Meirav Taieb-Maimon. EEG-triggered dynamic difficulty adjustment for multiplayer games. *Entertainment Computing*, 25:14–25, March 2018.

[116] Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep Learning with PyTorch*. Simon and Schuster, August 2020. Google-Books-ID: fff1DwAAQBAJ.

[117] Suparshya Babu Sukhavasi, Susrutha Babu Sukhavasi, Khaled Elleithy, Ahmed El-Sayed, and Abdelrahman Elleithy. A Hybrid Model for Driver Emotion Detection Using Feature Fusion Approach. *International Journal of Environmental Research and Public Health*, 19(5):3085, January 2022. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[118] Shuyang Sun, Zhanghui Kuang, Wanli Ouyang, Lu Sheng, and Wei Zhang. Optical Flow Guided Feature: A Fast and Robust Motion Representation for Video Action Recognition, July 2018. arXiv:1711.11152 [cs].

[119] Jaeyong Sung, Colin Ponce, B. Selman, and Ashutosh Saxena. Unstructured human activity detection from RGBD images. *2012 IEEE International Conference on Robotics and Automation*, 2012.

[120] Josh M Susskind, Adam K Anderson, and Geoffrey E Hinton. The toronto face database. *Department of Computer Science, University of Toronto, Toronto, ON, Canada, Tech. Rep*, 3, 2010.

[121] Jonathan Sykes and Simon Brown. Affective gaming: measuring emotion through the gamepad. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 732–733, New York, NY, USA, April 2003. Association for Computing Machinery.

[122] Srikanth Tammina. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10):p9420, October 2019.

[123] Chin Hiong Tan, Kay Chen Tan, and Arthur Tay. Dynamic Game Difficulty Scaling Using Adaptive Behavior-Based AI. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(4):289–301, December 2011. Conference Name: IEEE Transactions on Computational Intelligence and AI in Games.

[124] Yichuan Tang. Deep Learning using Linear Support Vector Machines, February 2015. Number: arXiv:1306.0239 arXiv:1306.0239 [cs, stat].

[125] Amin Ullah, Khan Muhammad, Javier Del Ser, Sung Wook Baik, and Victor Hugo C. de Albuquerque. Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM. *IEEE Transactions on Industrial Electronics*, 66(12):9692–9702, December 2019. Conference Name: IEEE Transactions on Industrial Electronics.

[126] Amin Ullah, Khan Muhammad, Ijaz Ul Haq, and Sung Wook Baik. Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments. *Future Generation Computer Systems*, 96:386–397, July 2019.

[127] Isabel Viana. Comunicação não verbal e expressões faciais das emoções básicas. *Revista de Letras*, 13(II):165–181, January 2014.

[128] Jiang wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view Action Modeling, Learning and Recognition, May 2014. arXiv:1405.2941 [cs].

[129] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[130] Lichen Wang, Zhengming Ding, Zhiqiang Tao, Yunyu Liu, and Yun Fu. Generative Multi-View Human Action Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6211–6220, October 2019. ISSN: 2380-7504.

[131] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes, January 2006. Journal Abbreviation: Comput. Vis. Image Underst. Publication Title: Comput. Vis. Image Underst. Volume: 104.

[132] Erroll Wood, Tadas Baltruaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764, Santiago, Chile, December 2015. IEEE.

[133] Lu Xia, Chia-Chih 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*, pages 20–27, Providence, RI, USA, June 2012. IEEE.

[134] Jia Xiang and Gengming Zhu. Joint Face Detection and Facial Expression Recognition with MTCNN. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pages 424–427, 2017.

[135] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. Language Supervised Training for Skeleton-based Action Recognition, August 2022. arXiv:2208.05318 [cs].

[136] Zhiding Yu and Cha Zhang. Image based Static Facial Expression Recognition with Multiple Deep Network Learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 435–442, Seattle Washington USA, November 2015. ACM.

[137] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, June 2012. ISSN: 2160-7516.

[138] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A Comprehensive Survey of Vision-Based Human Action Recognition Methods. *Sensors*, 19(5):1005, January 2019. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[139] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, October 2016.

[140] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1963–1978, August 2019. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[141] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, Boston, MA, USA, June 2015. IEEE.

[142] He Zhao, Qiming Wang, Zhaozhu Jia, Yiming Chen, and Jianxin Zhang. Bayesian based Facial Expression Recognition Transformer Model in Uncertainty. In *2021 International Conference on Digital Society and Intelligent Systems (DSInS)*, pages 157–161, December 2021.

[143] Xiaoming Zhao and Shiqing Zhang. A Review on Facial Expression Recognition: Feature Extraction and Classification. *IETE Technical Review*, 33(5):505–517, September 2016.

[144] Mohammad Zohaib. Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. *Advances in Human-Computer Interaction*, 2018:e5681652, November 2018. Publisher: Hindawi.

[145] Shender Ávila Sansores, Felipe Orihuela-Espina, and Luis Enrique-Sucar. Patient Tailored Virtual Rehabilitation. In José L Pons, Diego Torricelli, and Marta Pajaro, editors, *Converging Clinical and Engineering Research on Neurorehabilitation*, Biosystems & Biorobotics, pages 879–883, Berlin, Heidelberg, 2013. Springer.