Vision, Graphics and Interactive Systems, Master of Science in Engineering

AALBORG UNIVERSITY

STUDENT REPORT

Master's Thesis

# Deep Emotion Recognition through Upper Body Movements and Facial Expression

Ana Rita Viana Nunes

This report was written using the document markup language LaTeX using the tool www.overleaf.com. For development and testing Python 3.6 and the Spyder IDE were used.

**School of Information and Communication Technology**
Aalborg University
http://www.aau.dk

**AALBORG UNIVERSITY**
STUDENT REPORT

**Title:**
Deep Emotion Recognition through Upper Body Movements and Facial Expression

**Theme:**
Computer Vision

**Project Period:**
Spring Semester 2019

**Project Group:**
1042

**Participant(s):**
Ana Rita Viana Nunes

**Supervisor(s):**
Mohammad Ahsanul Haque
Chaudhary Muhammad Aqdus Ilyas

**Page Numbers:** 63

**Date of Completion:**
June 4, 2019

**Abstract:**

The automatic recognition of human emotions has become a subject of interest in recent years. The need to improve the interaction between human and machine has led researchers to focus on the subject of human emotion recognition as a solution to the minimal level of human-machine interaction nowadays. By being able to recognize emotions, machines such as robots will be able to better interact with humans by reacting according to their emotions, thus enriching the user experience.

In this thesis, two modalities of emotion expression will be analyzed, namely, facial expression and upper body movements. Both these modalities contribute greatly to the communication of a person's emotions, much more than their words.

To recognize emotions from both modalities, Convolutional Neural Networks will be trained using benchmark datasets of subjects performing different emotions. Later, the results from each modality will be fused to formulate the final bimodal emotion recognition system.

# Preface

This report documents the thesis for the Master of Science in Engineering, in Vision, Graphics and Interactive Systems at Aalborg University. The report was written by Ana Rita Viana Nunes during the Spring semester of 2019.

The author would like to thank the supervisors Mohammad Ahsanul Haque and Chaudhary Muhammad Aqdus Ilyas for the guidance given throughout the semester.

Aalborg University, June 4, 2019

Ana Rita Viana Nunes
aviana17@student.aau.dk

# Contents

# Chapter 1

# Introduction

Emotion is a spontaneous mental state that lasts for a few seconds or minutes. It solely indicates the current state of mind of a person and not their long-term feelings. The aim of emotion recognition is, therefore, to automatically identify a person's current emotional state.

The interaction between two parties is impaired when one of the parties is not able to recognize or understand the other's emotions. This applies to human-human interaction but also to human-computer interaction. Affective computing is a field that attempts to enhance the interactions between human and machine by developing artificial systems that are able to recognize human emotions and react according to them [1].

For the interaction between humans and machines to be more natural, machines should have the capability of recognizing human emotions. This could be applied in sociable robotics where robots are able to assist people in simple tasks such as delivering meals or vacuuming the house. These human-machine interactions right now are still minimal and could improve if the robot had more knowledge about the person they need to interact with [2].

Knowing what people feel when interacting with machines would allow the said machines to adapt and improve their interaction by having a suitable reaction to people's feelings. Taking the example of humanoid robots that provide services to people, the robot-human interaction would greatly improve if these robots were able to adjust their reactions to people's current emotions [3, 4, 5].

Face and speech recognition can even be used to remotely monitor patients. Using such a system could help health care professionals monitor elderly patients' state such as pain level through their face images and speech recordings and allow for assistance when

necessary [6, 7].

Human emotions are multimodal, they can be portrayed by different modalities such as facial expression, body language and movements or tone of voice. Each modality has its constraints and thus needs to be analyzed differently. This makes it hard for machines to process all this information and be able to recognize whatever emotion is being displayed. Hence, the most optimal way of representing emotions must be researched.

So far, automatic emotion recognition research has focused mainly on the recognition of facial expressions and speech [8, 9]. However, recent research shows that body language comprises a significant amount of affective information. Body language can be expressed in different ways, from facial expressions to body posture, eye movement, gestures, touch or even personal space.

The best modalities to use and how to combine them in order to get the best recognition rate of human emotions is still an object of research. Even though current techniques can achieve good recognition results, there is still room for improvement in order to get the best results possible and also to be able to achieve them using less computational power.

A lot of research has been done regarding human emotion recognition. Most of the research has focused on facial expression recognition and fewer has been focused on body movements emotion recognition. Studies fusing both of these emotion expression modalities are not many, so this is an area that can still be studied further.

From the studies regarding these emotion recognition modalities, most have taken conventional approaches into account, like Gunes and Piccardi in [10] or Shan et al. in [11], as discussed in Section 2.2. Only in recent years, researchers have turned their focus into deep learning approaches to solve this issue, which have been able to achieve the best recognition rates.

## 1.1 Contribution of the Thesis

Barros et al. in [12] and Sun et al. in [13] used Convolutional Neural Networks (CNN) to recognize emotion in both face and body movements, additionally, both studies incorporated temporal features into their classification, which forces them to analyze an entire video before it can be classified.

In this thesis a different approach will be studied, Convolutional Neural Networks will also be used to classify emotions, however, temporal features will not be considered. This way it is possible to use this system in single images and, this way, use it in videos in real-time, by classifying each frame of the video.

Even though deep learning approaches show an improved accuracy when compared to conventional approaches, they are still more computationally demanding, which is a big downside and often makes them not the most viable option.

Hence, this work will also explore a solution to make the implementation of this method less computationally demanding, so this solution can be used in almost any system, regardless of its computational capabilities.

## 1.2 Report Organization

An overview of previous work done in the field of emotion recognition will be provided in Chapter 2, along with benchmark datasets that are used to test the researched solutions.

Chapter 3 will explore the technical aspects of the technologies that will be used to extract features and categorize the video samples from the datasets.

The implementation of the proposed solution will be described in Chapters 4, 5, and 6, where each aspect of the created system will be explained in detail.

The results of the system are revealed in Chapter 7 and, lastly, Chapter 8 presents the conclusion of the work.

# Chapter 2

# Literature Review

An initial analysis of the problem is performed about the subject of emotion recognition. Related works are researched to discover what approaches have been taken in the emotion recognition research field, and to discuss which ones have worked best and should be further explored in this work.

## 2.1 Emotion Classification

Psychologists Paul Ekman and Wallace V. Friesen proved that distinct facial behaviors are universally associated with distinct emotions. They identified six basic emotions and concluded that they are universally recognized across different cultures [14]. The emotions are anger, disgust, fear, happiness, sadness, and surprise, they are represented in Figure 2.1.

Another approach to emotion classification is to model emotions according to two main emotion dimensions: arousal and valence. The valence corresponds to the pleasantness or unpleasantness of a feeling and can be assigned a positive or negative score, respectively. Arousal measures how engaged or apathetic a person is and is assigned a high or low score.

Most of the current literature adopts Ekman's and Friesen's approach of the six basic emotions. But often, a seventh neutral emotion is added.

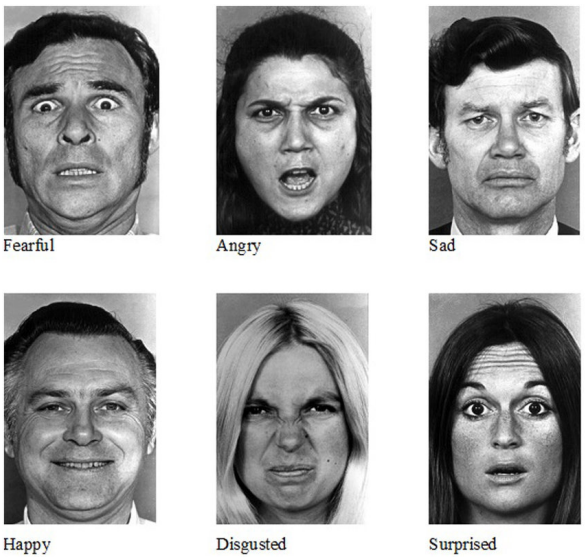


**Figure 2.1:** Examples of the Ekman and Friesen pictures of facial affect [15].

## 2.2 Emotion Recognition Approaches

Automatic emotion recognition has been a growing field of research in the last decade. According to Mehrabian [16], only 7% of human communication is conveyed through words, 38% through vocal tone and 55% through non-verbal elements such as facial expression, body language, and gestures.

Therefore, emotion recognition can be done through different modalities such as facial expressions [17, 18], speech recognition [19] or body language [20]. Different modalities can also be combined in an attempt to form a more accurate emotion recognition system [8, 9, 21].

In this work, only non-verbal emotion communication will be further explored, hence the modalities of facial expression and body language.

### 2.2.1 Facial Expression Recognition

In interpersonal communication, facial expressions are one of the main ways of providing information about emotions. Facial Emotion Recognition (FER) is, therefore, the most researched modality of emotion recognition. There are two paths researchers can follow when implementing a system for facial emotion recognition, a conventional approach with handcrafted features or a deep learning approach.

### Conventional Approaches

Conventional FER approaches are composed of three steps: face and facial component detection, feature extraction and expression classification, Figure 2.2. From an input image, the face region and facial landmarks or components (e.g. eyes and mouth) are detected. Then, a number of spatial and temporal features are extracted from the facial components or landmarks. Finally, the facial emotion is classified based on a pre-trained classifier (e.g. Support Vector Machine, AdaBoost) [22].

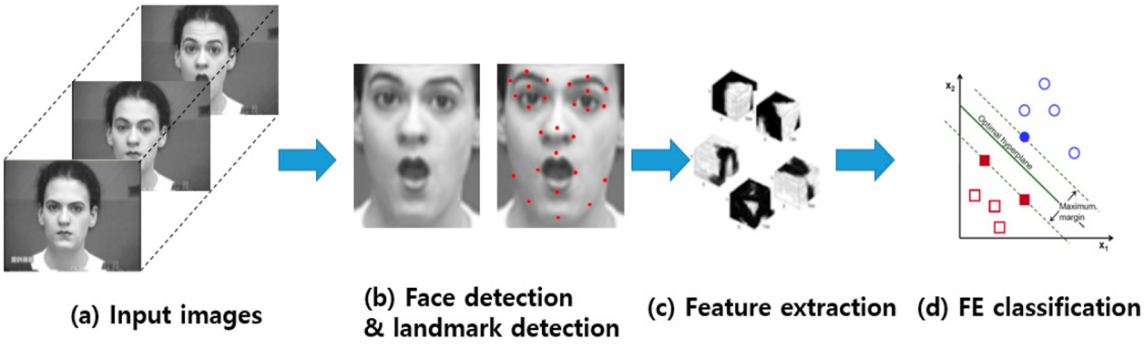


**Figure 2.2:** Procedure used in conventional FER approaches [22].

The feature extraction step can be performed by extracting geometric features, appearance features or a hybrid combination of both.

Noroozi et al. [8] proposed a method that recognizes emotions from audio-visual data. To analyze the visual data, the data is represented by key-frames which are selected automatically using a clustering-based strategy. Geometric features are used to define the key-frames, and 68 landmark points that correspond to six regions of the face are extracted. The geometric features correspond to the shape of certain parts of the face such as eyes and nose, and the location of facial points. Principal Component Analysis (PCA) is used to reduce the dimensionality of the data and, to classify it, Multiclass-SVM is used. Appearance features are based on the entire face or on regions of it. Happy et al. [23] used, as feature vectors, a Local Binary Patterns (LBP) histogram of different block sizes from a face image, and classified multiple facial expressions using PCA.

Hybrid approaches such as the one proposed by Ghimire et al. [24] combine both geometric and appearance features in an attempt to improve the recognition accuracy. In the mentioned paper the appearance features are computed by dividing the face region into domain-specific local regions.

The advantage of conventional approaches is that they require low computing power and memory, whilst deep learning approaches are more computationally demanding.

Thus, conventional approaches are still being studied for use in real-time systems.

### Deep Learning Approaches

Nowadays, with the availability of large datasets, deep learning has become a state-of-the-art solution to problems such as emotion recognition. The Convolutional Neural Network (CNN) is a type of deep learning that is especially used in the processing of images.

Deep leaning based algorithms can be used for feature extraction and classification. With the use of CNNs the work spent on the pre-processing of the images is greatly reduced since the algorithm is already capable of detecting the best features needed to classify the images. A detailed explanation about what CNNs are and how they work will be given in Section 3.3.

Because CNN based methods cannot reflect temporal variations, recently some researchers have combined CNN, for the spatial features of single frames, with Long Short-Term Memory (LSTM), for temporal features of successive frames. LSTM is a special type of a Recurrent Neural Network (RNN) that can solve long-term dependency using short-term memory [22].

Taking a deep learning approach, Khorrami et al. [25] combined CNNs with RNNs and analyzed how much each neural network component contributed to the emotion recognition system's overall performance. Two different frameworks were used for training, the first a single frame CNN, and the second, a combination of CNN and RNN. The CNN models were trained using the Anna software library [26]. Even though the single frame regression CNN learns useful features from the video data, it disregards temporal information. Through the use of RNN, this information can be incorporated. Results determined that the CNN+RNN model translates to more accurate predictions.

With a similar approach, Fan and Ke in [27] also used CNNs for emotion recognition and RNN to model temporal information.

On an approach that explores the use of real-time emotion recognition, Arriaga et al. in [28] created a real-time vision system that detects faces and classifies them based on gender and displayed emotion. Their aim was to reduce the number of CNN parameters as a way to reduce computational cost and achieve better generalization. They proposed two different models. The first was a standard fully-convolutional neural network. The second was inspired by the Xception architecture [29], which combines the use of residual modules and depth-wise separable convolutions (which reduce the computation compared to the standard convolutions). They called this last model mini-Xception. They also introduced a real-time guided backpropagation visualization technique to observe the learned features.

Tests were performed with the FER-2013 dataset [30] and an accuracy of 66% for the emotion classification task was obtained with both models.

#### Remarks

Comparing the discussed approaches, the conventional approaches have an average accuracy of 63.20% while the deep learning approaches completely surpass that value with an average accuracy of 72.65% [22].

Even though deep learning methods demand large training datasets and great computing power, they are the best approach to achieve the highest performance.

### 2.2.2 Body Gestures Emotion Recognition

Upper body movements are of special interest. People often, though sometimes unaware of that, use head and hand movements to express their state of mind. Therefore, others are capable of inferring a person's current emotion when observing their body language [31, 32].

For instance, when a person displays a neutral emotion, they generally do not move their arms, however, when they are happy or sad the body tends to be extended and the hands move upwards closer to the head [33].

#### Conventional Approaches

Piana et al. [20] suggested an approach to real-time automatic emotion recognition from body movements and gestures. The features are extracted from 3D motion clips containing full-body movements that are recorded using two different systems, a professional optical motion capture system and Microsoft Kinect. The body joints are tracked, and feature vectors of the movements are extracted and used for classification using a linear SVM classifier. The emotions tested were the six standard emotions. Human validation demonstrated that three of the emotions were easily recognized from body movements (happiness, sadness, and anger), while the others (surprise, disgust, and fear) were confused with each other. Because of that, a sub-problem with only four emotions (happiness, sadness, anger, and fear) was formulated. The approach showed better results when only classifying four out of the six emotions.

A different approach was taken by Glowinski et al. [34] which analyzed affective behavior solely based on upper-body movements. A range of twelve different emotions

was classified according to their valence and arousal. Features were extracted from two videos, one that displayed a frontal view of the subjects and another that displayed a lateral view. The trajectories of the head and hands were tracked, and low-level physical measures i.e. position, speed, acceleration were extracted. Higher-level expressive and dynamic features were then computed, e.g. smoothness and continuity of movement, spatial symmetry of the hands, gesture duration, etc., forming a 25-features vector. In Figure 2.3, it is possible to notice the hands and head detection and a bounding triangle between these that was used to extract features from the movements of the body parts. Principal Component Analysis (PCA) was later applied to reduce the dimensionality of the data. Furthermore, clustering was used to classify the data into four clusters according to the categorical variables i.e. valence (positive, negative) and arousal (high, low). The framework was tested on the GEMEP (GEneva Multimodal Emotion Portrayals) dataset [35]. The results demonstrate that gestures can be effectively used to detect human emotion expression.

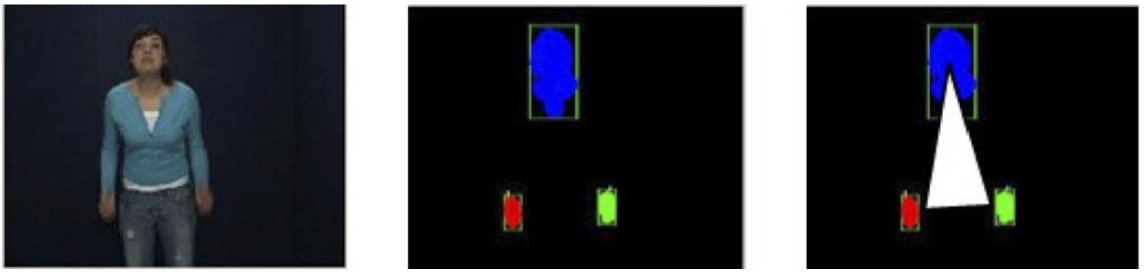

**Figure 2.3:** On the left the original image is presented, in the middle, the detection of the body parts is displayed, and on the right, the bounding triangle that connects the body parts [34].

### Deep Learning Approaches

Using a deep neural network approach, the system does not rely on different feature extraction techniques, instead, the model learns by itself, layer after layer, which are the most important features and continuously passes those features onto deeper layers.

Barros et al. [12] integrated multiple modalities of non-verbal emotion recognition. To recognize emotion through body movements they resorted to a previous work, done by some of the same authors, that focused on recognizing gestures in real-time with a deep neural architecture. In [36], Barros et al. extracted temporal and spatial features of gesture sequences using a Deep Neural Network (DNN). The first layer of the DNN receives as input a series of frames and generates a motion representation. Moreover, a Multichannel Convolutional Neural Network (MCCNN) is used to learn and extract features from the previously generated motion representation and uses such features to classify different gestures.

### 2.2.3 Bimodality

Though single modality emotion recognition (only based on facial expression or body movements) shows good results, the fusion of multiple modalities is capable of achieving better recognition performance. Nevertheless, a good fusion strategy must be applied, otherwise, the fusion of modalities can have a negative effect on the accuracy of the recognition system.

Gunes and Piccardi studied this case precisely [10, 37], they conducted experiments where only single modalities were tested (facial expression or body gestures) and where both modalities were fused to formulate a detection. The results revealed that, as expected, the bimodal approach had better performance.

Barros et al. [12] also considered a bimodal approach to emotion recognition, taking into account facial expression and body movements. They used neural networks on their solution and achieved a much higher average accuracy when fusing both modalities together, as opposed to testing them individually. Going from $57.84 \pm 7.7\%$ on body motion and $72.70 \pm 3.1\%$ on facial expression, to $91.30 \pm 2.7\%$ average accuracy on bimodal emotion recognition.

## 2.3 Benchmark Datasets

There are a number of public datasets available suited for emotion analysis. In this section, some of them will be analyzed in order to choose the correct one to be used for this work.

One of the firsts datasets made accessible was the FABO database, created by Gunes and Piccardi [38]. It is a bimodal database that combines face and body expressions recorded simultaneously. The videos were obtained in a lab setting with artificial light and the emotions were posed. Subjects were placed in front of a plain blue background to facilitate background subtraction. 23 subjects were filmed, with ages ranging from 18 to 50. The subjects were instructed to start off with a neutral expression and body position, and then perform the supposed emotion based on scenarios they were presented with. The expressions recorded were: neutral, uncertainty, anger, surprise, fear, anxiety, happiness, disgust, boredom, and sadness. Thus, containing all of the six basic emotions. Figure 2.4 represents some frame examples from the database.

The GEneva Multimodal Emotion Portrayals (GEMEP) database [35] contains audio-visual files that include 18 different emotions displays. Twelve of the emotion classes are categorized by two emotional dimensions: valence and arousal. The subjects that performed the emotions were French theater actors between 25 and 57 years old. The

**Figure 2.4:** Example sequences from FABO obtained from body (left columns) and face (right columns) cameras [38].

videos were recorded in a controlled setting in a studio. The expressions were recorded with three digital cameras, one was zoomed in on the face, another was zoomed out displaying the body and posture from a frontal view, and the last from a side view, Figure 2.5. The audio was also recorded using four microphones.

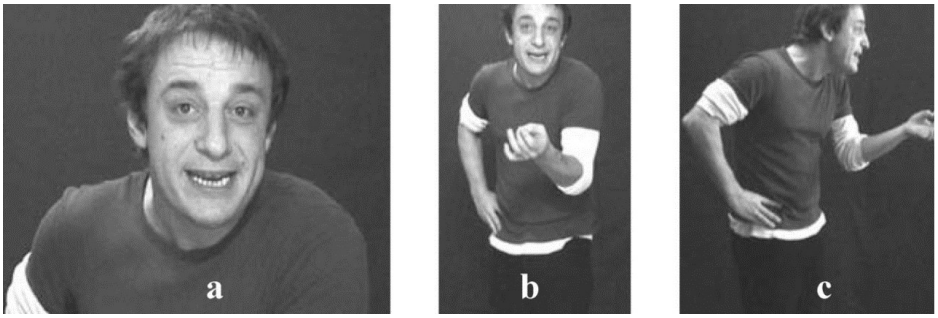


**Figure 2.5:** Frames demonstrating the three camera angles used in the video recordings [35].

Pierre-Luc Carrier and Aaron Courville created the FER-2013 dataset introduced in the Facial Expression Recognition Challenge organized by ICML 2013 Workshop in Challenges in Representation Learning. The dataset is publicly available at Kaggle [30]. The images were collected using Google Image search API and were cropped to only display faces. The faces are centered and occupy similar areas on the images. The images were resized to 48x48 pixels and converted to grayscale. The dataset contains 35887 annotated images, displaying the six basic emotions (anger, disgust, fear, happiness, sadness, surprise) plus a neutral emotion. Ian Goodfellow discovered that human accuracy on this dataset was $65 \pm 5\%$ [39].

The MMI Facial Expression Database was created by Pantic et al. [40] and includes about 1500 samples of static images and image sequences of faces both with a frontal and

**Figure 2.6:** Samples of the FER-2013 dataset [30].

a profile view. These faces display various types of emotions from subjects from both genders and from different ethnicities.

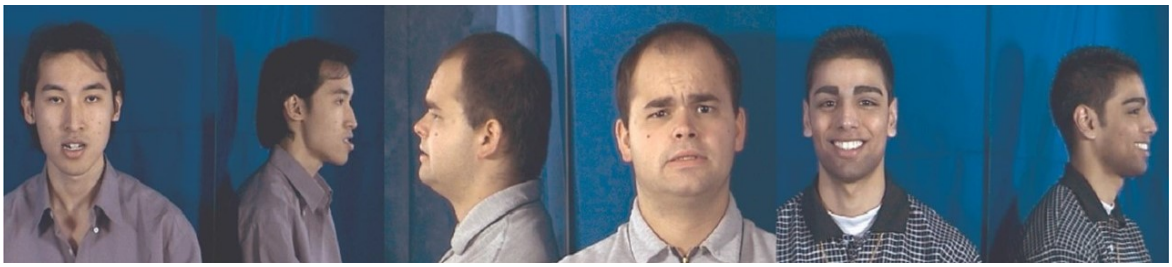

**Figure 2.7:** Samples from the MMI Facial Expression Database [40].

# Chapter 3

# Technical Background

As discussed in the previous chapter, the state-of-the-art results reside in deep learning approaches and more specifically in the use of Convolutional Neural Networks. As grounds for the work to be implemented in the next chapters, this chapter will give a brief overview of what are neural networks and what is deep learning. Furthermore, it will describe how a CNN works and what are the functions of its different layers.

## 3.1 Neural Networks

Artificial Neural Networks (ANNs) simulate the connected networks of neurons in the human brain. Different pieces of information are processed by different parts of the human brain which are organized in layers. Information enters the brain and is processed and passed through each of these layers, see Figure 3.1. As a simulation of this, ANNs can have different layers that, first receive information, process it and then pass it through to the following layer.

A simple ANN can merely have an input layer where the input data is received, a hidden layer where the data is processed, and an output layer where a decision is made about what to do given the collected data, Figure 3.2.

ANNs have multiple nodes that replicate neurons. Nodes are connected by links that allow the information to be passed through node to node, layer to layer. These links have weight values which allow the network to learn. Nodes take input data, perform simple operations on it, and then pass it to other nodes. The output of a node is called node value or activation.

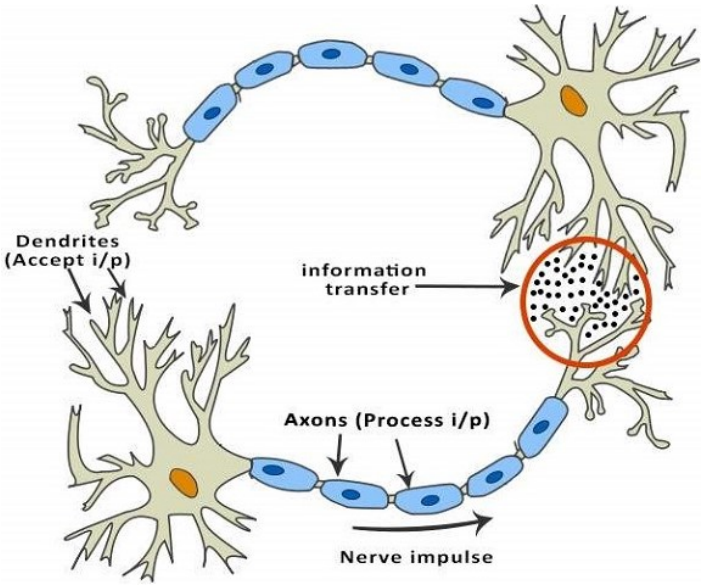


**Figure 3.1:** Inner workings of the human brain. Dendrites act as input terminals, neurons act as processing units and the axons act as output terminals [41].

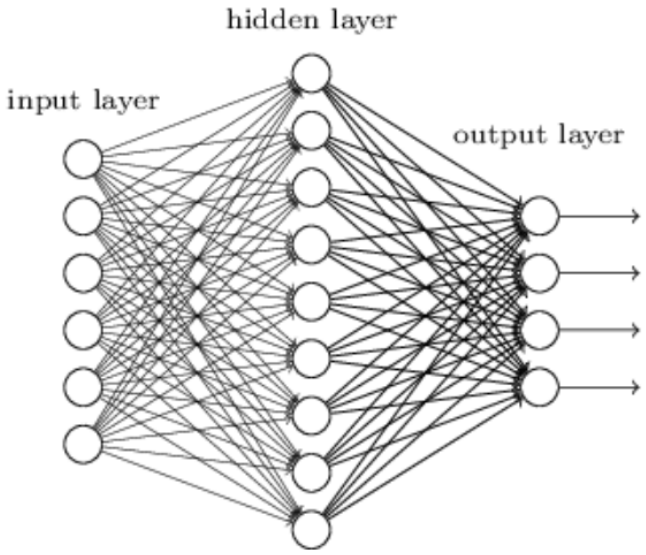


**Figure 3.2:** Artificial Neural Network [42].

## 3.2 Deep Learning

Deep learning is a machine learning method that is used in neural networks. There are different deep learning architectures, for instance, Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and so on, Figure 3.3. These methods can be used in different fields such as speech recognition, object detection, object recognition, and many others.

Deep learning uses a cascade of successive layers of nonlinear processing units that each uses the output from the previous layer as input. Each layer transforms its input into a more abstract representation of the data. A deep learning system learns on its own by filtering information through several hidden layers.

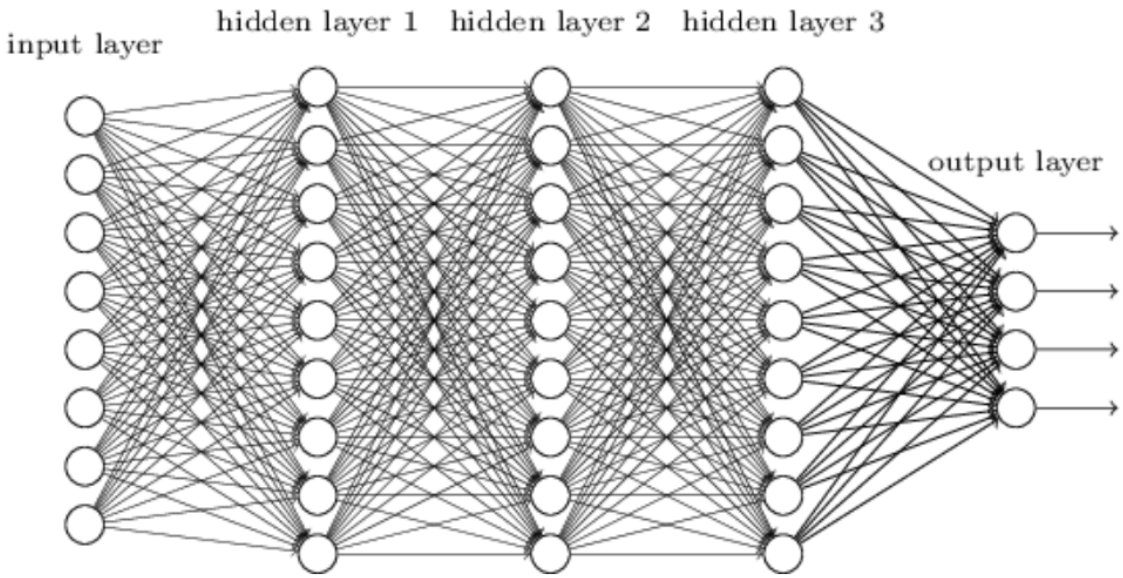


**Figure 3.3:** Deep Neural Network [42].

## 3.3 Convolutional Neural Network

In 1998, LeCun et al. [43] introduced Convolutional Neural Networks (CNN). This deep learning architecture is mainly used to classify images and perform object recognition. They can be used to identify faces, objects, handwritten characters and much more.

As neural networks, CNNs have a sequence of layers, the three main types of layers are Convolutional Layer, Pooling Layer, and Fully Connected Layer. An example of a CNN

architecture is displayed in Figure 3.4.

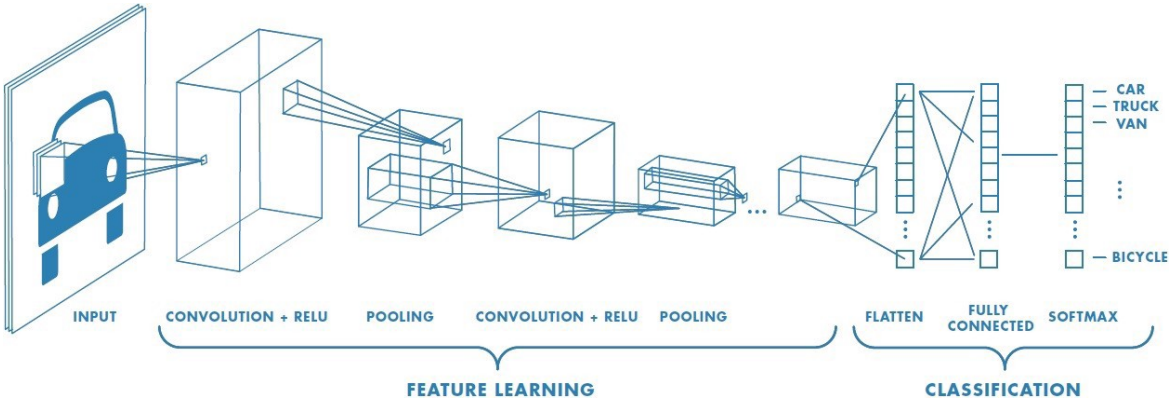


**Figure 3.4:** Architecture of a CNN [44].

### 3.3.1 Convolution Layer

RGB images are represented by 3D arrays. Two of the dimensions relate to the width and height of the image, thus portraying each pixel in the image. The third dimension conveys the three channels of color, with the intensity values of the red, green, and blue colors, for each pixel. Grayscale images only have one channel of color, representing the level of light for each pixel.

CNNs pass multiple filters over an image, each one selecting different signals. Initial layers can pass horizontal, vertical or diagonal line filters to create maps of the edges of an image.

A convolutional network selects square patches of pixels from an image and passes them through a filter. A filter or kernel is a square matrix, smaller than the dimensions of the image and equal to the dimension of the patches to be processed. The filter must have the same depth (number of channels) as the image. The matrix values will be passed through the patch and help find patterns within the image.

Starting in the upper left corner of the image, the filter is moved across the image from left to right and from top to bottom until it reaches the lower right corner. The number of pixels the filter skips before analyzing the next patch is called stride.

The filter performs a dot product with the weights in the filter and the values in the patch of the image being processed. The result of the dot product is saved in a third matrix called an activation map. Besides the weights, the result also takes into account one bias parameter. The dimensions of this matrix will depend on the stride. A larger stride leads to a smaller output matrix and therefore, less computing time.

It might be convenient to pad the input image with zeros around the borders. This is done so the pixels around the borders are not lost and the output volume can keep the same dimensions as the input image.

Multiple filters can be applied to the image, thus producing multiple activation maps which will be stacked. If the convolution of an image results on an activation map of dimension 100x100 and 20 different types of patterns were processed, the resulting volume will be 100x100x20.

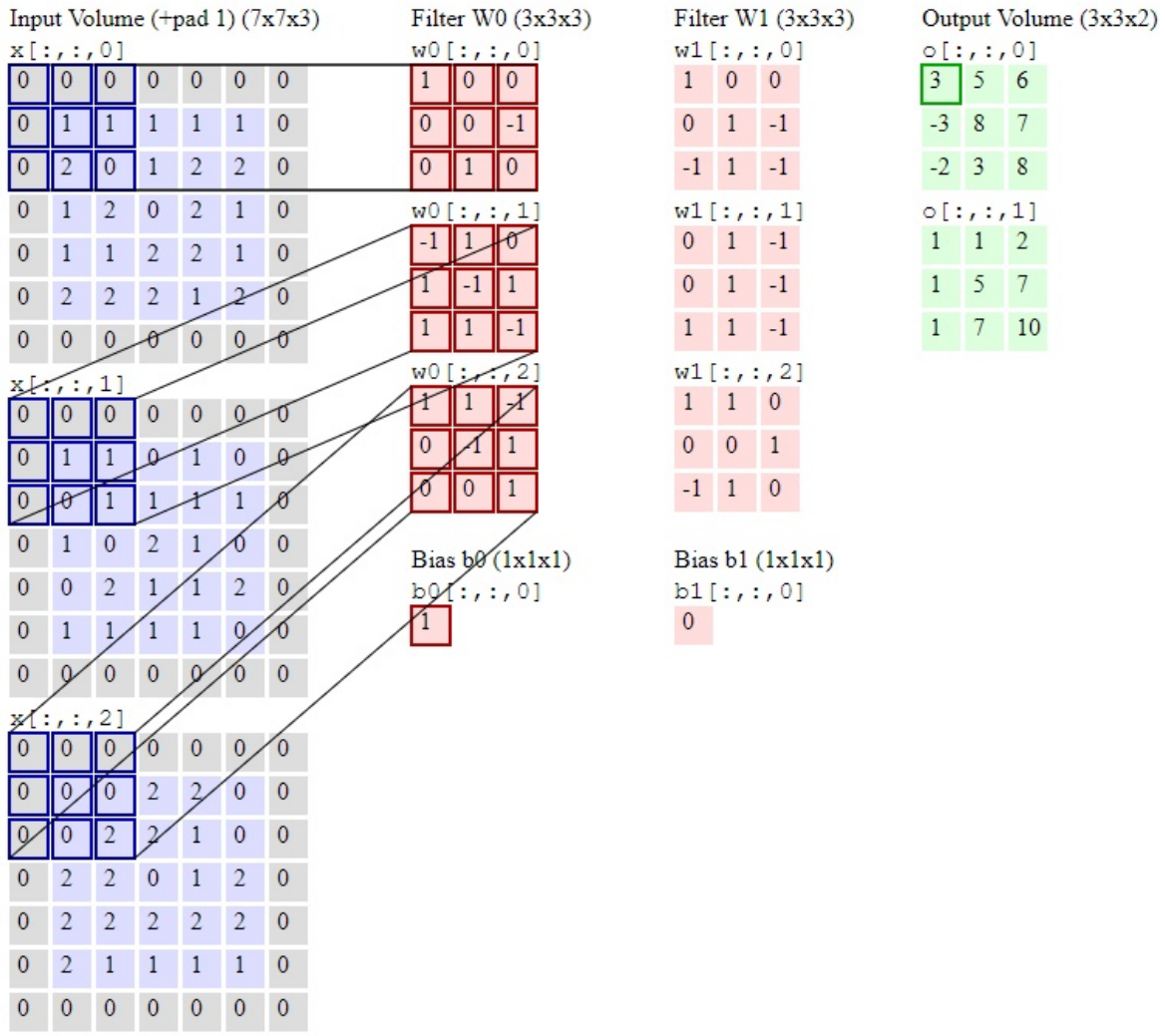


**Figure 3.5:** In blue are represented the 3 channels of an input image. The weights of two filters are shown in red. The output activation map for both filters is colored in green [45].

### 3.3.2 ReLU Activation Function

Any neural network needs to contain non-linearity, which can be achieved by passing the weighted sum of its inputs through an activation function. This activation function is usually used after each Convolution Layer.

The ReLU (rectified linear unit) removes negative values from activation maps and sets them to zero, Figure 3.6. It also allows better propagation of the error's gradient through the network because it has a constant gradient of 1 to all input values greater than zero.

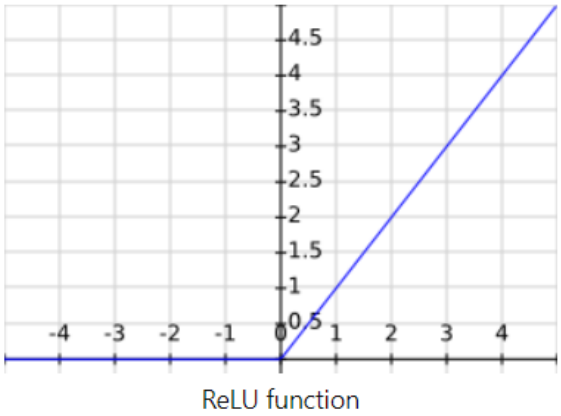


**Figure 3.6:** ReLU activation function [46].

### 3.3.3 Pooling Layer

The most common type of Pooling Layer in CNNs is Max Pooling.

The Max Pooling layers receive as input the activation maps that resulted from the Convolution Layer. Similar to convolution, this method is applied one patch at a time. Max pooling selects the greatest value from the selected patch and translates it to another matrix with the max pooling results of all patches, Figure 3.7.

Max pooling is commonly used between successive Convolution Layers. This layer reduces the spatial size of the representation and, consequently, reduces the number of parameters and computation required.

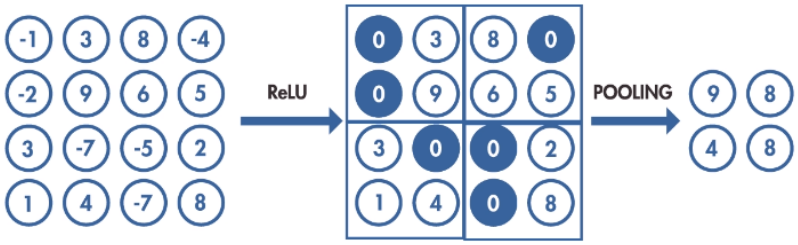


**Figure 3.7:** Application of ReLU and Max Pooling [44].

### 3.3.4 Fully Connected Layer

The Fully Connected Layer performs high-level reasoning, it connects every neuron from the layer to all activations from the previous layer.

This layer is attached to the end of the network, the input volume of whatever layer precedes it is flattened to a vector. The Fully Connected Layer will then return a 1D vector, with its dimension equal to the number of classes the network can recognize. Numbers in the vector represent the probability of an input image belonging to each of the classes.

### 3.3.5 SoftMax Activation Function

It is used in the Fully Connected Layer to provide the probabilities for each class. It delivers an output between the range of zero to one and the sum of all values amounts to one.

### 3.3.6 Training

The network adjusts the filter values or weights through backpropagation which has four sections, forward pass, loss function, backward pass and weight update.

During the forward pass a training image passes through the entire network. The weights are initialized randomly. The goal is to minimize the loss of the network by performing a backward pass, determining which weights contributed to the loss of the network, and updating the weights so the loss decreases as much as possible.

# Chapter 4

# Facial Expression Recognition

As discussed in Section 2.2, facial expression recognition will be implemented using a deep learning approach. Therefore, a CNN model for classification of the facial expressions will be trained using standard datasets.

First of all, a facial recognition algorithm needs to be applied to one of the datasets to extract solely the facial region of the videos' frames, this is explored in Section 4.1.

The datasets used to train the CNN were chosen from the benchmark datasets publicly available or made available to the research community, and they are described in Section 4.2.

More importantly, in Section 4.3, the implemented CNN architecture is explained, along with a detailed description of how the CNN model works and processes input images.

## 4.1 Face Detection

To train and test the CNN model, the FABO dataset is used. This dataset contains videos of subjects' upper body so, to uniquely test the facial emotion recognition, the face of the subjects needs to be detected in each frame.

Since 2001, Haar Cascade face detection was the state-of-the-art method for face detection, after it was introduced by Viola and Jones in [47]. This is a machine learning based method where a cascade function is trained from positive and negative images, and is then employed to detect objects in images. This method is able to work in real time and detect

faces at different scales. However, it outputs a lot of false predictions and does not work under occlusions or with non-frontal faces.

A Deep Neural Network (DNN) face detector module was included in OpenCV 3.3 [48]. The DNN model is based on the Single-Shot-Multibox detector and uses the ResNet-10 architecture. It was trained using images from the web and provides models for the Caffe and Tensorflow frameworks.

The DNN face detector method is more accurate than the Haar Cascade method. They both run in real time and detect faces with different scales but, the DNN face detector, is capable of detecting faces with different orientations and under different levels of occlusion.

For the mentioned reasons, the DNN face detector OpenCV model was used, and Figure 4.1 demonstrates the result of the face detection on a video sample from the FABO dataset.

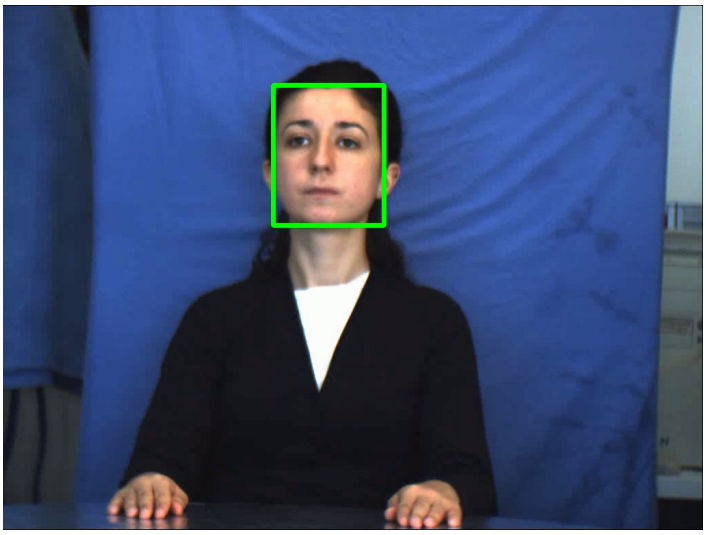

**Figure 4.1:** DNN face detection on a video frame.

## 4.2 Training Datasets

The datasets used to train the CNN will be the FER-2013 [39] and the FABO dataset [38]. The FER-2013 dataset was chosen since it has been widely used to train CNN models regarding facial emotion recognition and it is publicly available. The FABO dataset is the dataset that best meets the requirements for the upper body analysis and the only one that was made available so, since it will be used for that, it was also integrated into the facial

analysis.

### 4.2.1 FER-2013

The FER-2013 dataset was downloaded from the Kaggle challenge [30]. This data is made of 48x48 pixel grayscale images of faces that are centered on the image and occupy more or less the same image area. The dataset contains 35887 annotated images, with 4953 anger images, 547 disgust, 5121 fear, 8989 happiness, 6077 sadness, 4002 surprise, and 6198 neutral images. Some samples of the images are shown in Figure 4.2.

Each image is tagged with a numeric, ranging from 0 to 6, that corresponds to the seven emotion categories (0 - anger, 1 - disgust, 2 - fear, 3 - happiness, 4 - sadness, 5 - surprise, 6 - neutral). The image information is given by a string with the pixel values of the image in row-major order.


**Figure 4.2:** FER-2013 sample images for each emotion.

### 4.2.2 FABO

The emotion classes performed on the videos of the FABO dataset are only annotated for 16 of the 23 subjects. Each video displays the same emotions from two to four times. The videos also have annotations regarding the phases of the affective states, the demonstration of each emotion is thus divided into the neutral, onset, apex and offset phases. In the neutral phase, the subject's face is relaxed and shows no signs of emotion. In the onset phase, the face and body posture begin to change. The apex phase happens when the changes reach a stable level and the facial appearance doesn't seem to change. When there is a relaxation of the expression and movements, that phase is classified as the offset. These phase annotations are only done for twelve of the subjects.

To train the CNN, only the frames in the apex phase are considered, since they are the ones that better represent the emotions. From the annotated videos, two apex phases are considered. After splitting the dataset into train and test data the number of images for each emotion is 1410 for anger, 458 for disgust, 343 for fear, 613 for happiness, 570 for sadness and 588 for surprise. The neutral emotion is the exception, the images for this emotion were obtained from the neutral phase from each video, amounting to 786 images.

The selected images display the upper body of the subjects, therefore, a facial recognition algorithm was applied to extract only the facial region within the image. The images were also resized to match the dimensions of the images of the FER-2013 dataset, and were changed from RGB to grayscale, see Figures 4.3 and 4.4.

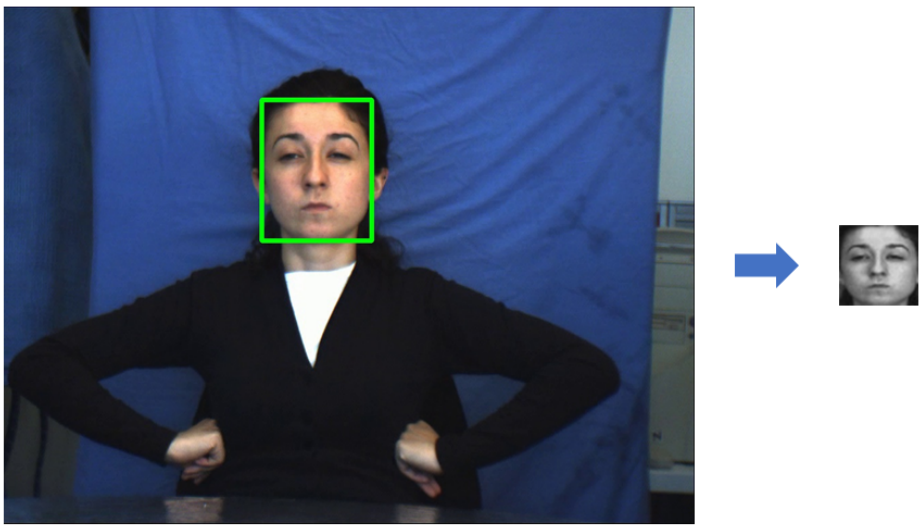

**Figure 4.3:** Selection of the facial region on a frame from the FABO dataset.

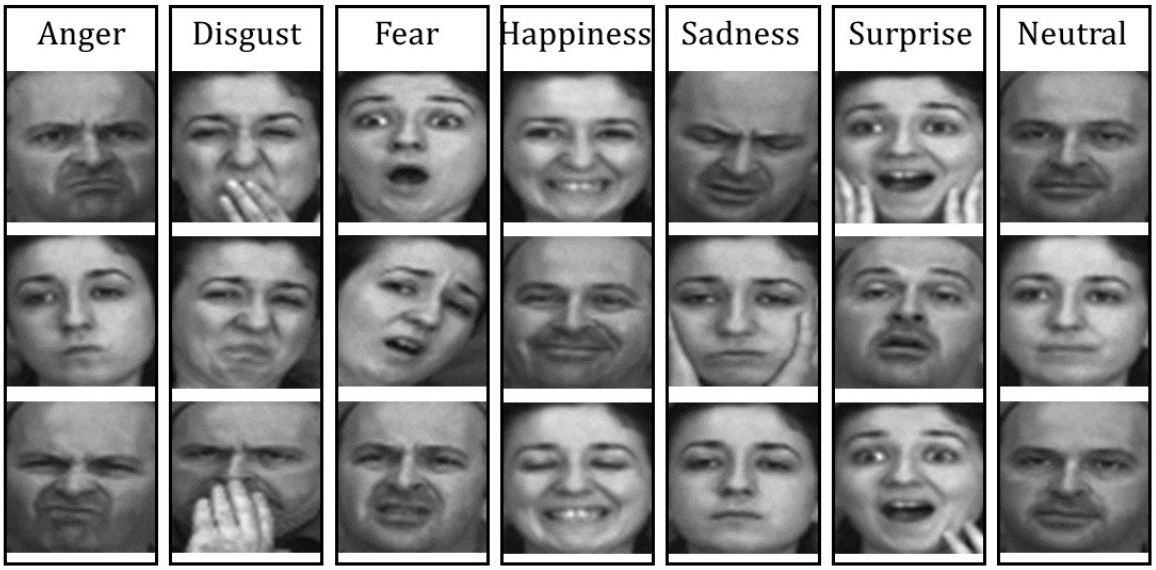


**Figure 4.4:** FABO sample images for each emotion.

### 4.2.3 Preprocessing

The data is preprocessed by scaling the pixel values to values between -1 and 1. This range of values is believed to be better for training neural networks in computer vision. Afterward, the dataset is divided into train and validation data with an 80/20 ratio.

## 4.3 Convolution Neural Network Model

A CNN model was trained in order to classify facial expressions according to which emotions they convey.

### 4.3.1 CNN Architecture

The implemented CNN architecture was proposed by Arriaga et al. in [28]. It is a simple architecture that achieves almost state-of-the-art performance classifying emotions. Arriaga et al. used this architecture to not only classify emotions based on facial expressions but also to classify faces according to gender. In their research solely the FER-2013 dataset was used for training regarding emotions classification whereas in this work also the FABO dataset will be used.

Standard CNN architectures require millions of parameters, which makes their use in real-time systems nonviable. Arriaga et al. wanted to create a CNN architecture to be used in real-time, therefore, an architecture that was simple and with low computation requirements.

Because most of the parameters are often located in the fully connected layers, Arriaga et al. decided to eliminate completely these layers. Furthermore, they also included in their architecture, depth-wise separable convolutions and residual models, both of which are used by state-of-the-art CNN architectures such as Xception [29].

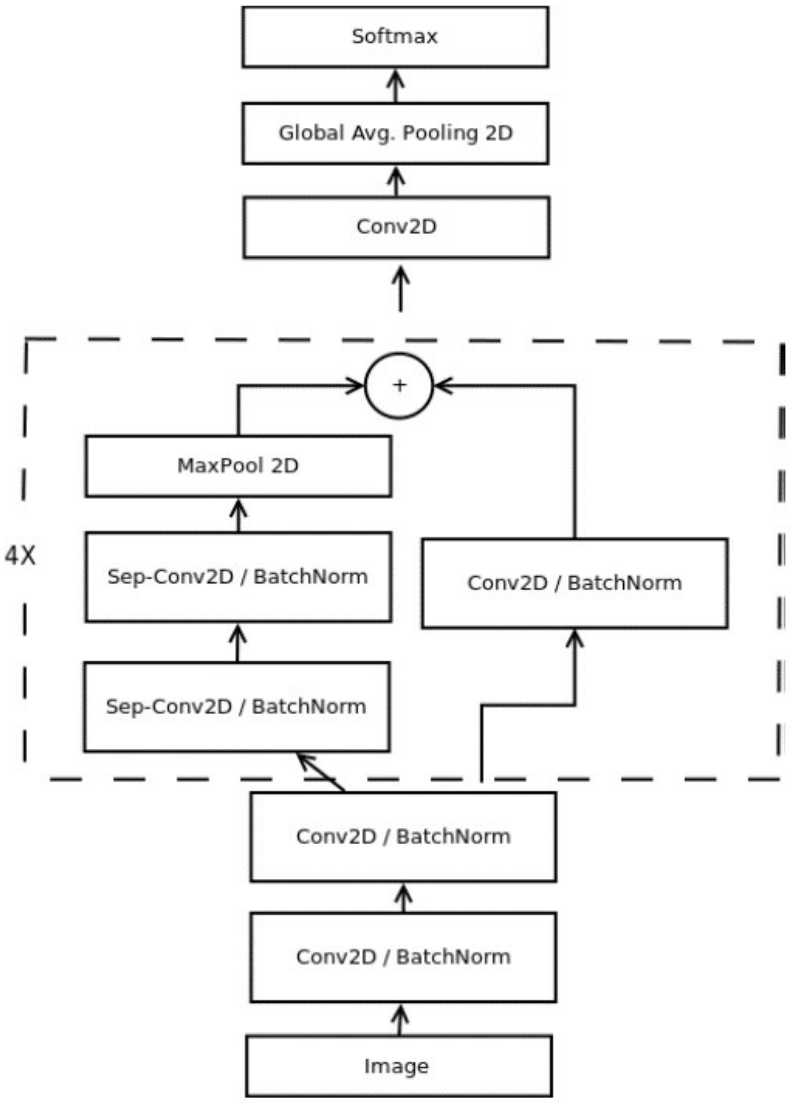


**Figure 4.5:** CNN architecture [28].

The implemented architecture is displayed in Figure 4.5. It starts off with standard 2D convolutions followed by batch normalization and the activation function ReLU. A kernel regularizer is used to apply penalties on layer parameters during optimization. Batch normalization normalizes the activation of the previous layer at each batch.

Residual modules are used, which modify the desired mapping between two subsequent layers. They connect the output of previous layers to the output of new layers.

Depth-wise separable convolutions reduce further the number of needed parameters. They are composed of depth-wise convolutions and point-wise convolutions.

Instead of the fully connected layers, this architecture uses Global Average Pooling. This reduces each feature map to a scalar by calculating the average of all the elements in the feature map. The last convolution layer has the same number of feature maps as the number of classes. In the end, a softmax activation function is applied to produce a prediction.

Listing 4.1 reveals a more detailed view of each of the layers of the CNN.

**Listing 4.1:** Layers of the CNN.

| Layer (**type**) | Output Shape | Param# | *Connected to* |
|---|---|---|---|
| input_1 (InputLayer) | (None,48,48,1) | 0 | |
| conv2d_2 (Conv2D) | (None,46,46,8) | 72 | input_1[0][0] |
| batch_normalization_2 (BatchNor | (None,46,46,8) | 32 | conv2d_2[0][0] |
| activation_2 (Activation) | (None,46,46,8) | 0 | batch_normalization_2[0][0] |
| separable_conv2d_1 (SeparableCo | (None,46,46,16) | 200 | activation_2[0][0] |
| batch_normalization_4 (BatchNor | (None,46,46,16) | 64 | separable_conv2d_1[0][0] |
| activation_3 (Activation) | (None,46,46,16) | 0 | batch_normalization_4[0][0] |
| separable_conv2d_2 (SeparableCo | (None,46,46,16) | 400 | activation_3[0][0] |
| batch_normalization_5 (BatchNor | (None,46,46,16) | 64 | separable_conv2d_2[0][0] |
| conv2d_3 (Conv2D) | (None,23,23,16) | 128 | activation_2[0][0] |
| max_pooling2d_1 (MaxPooling2D) | (None,23,23,16) | 0 | batch_normalization_5[0][0] |
| batch_normalization_3 (BatchNor | (None,23,23,16) | 64 | conv2d_3[0][0] |
| add_1 (Add) | (None,23,23,16) | 0 | max_pooling2d_1[0][0] |

```
                                                                    batch_normalization_3[0][0]
__________________________________________________________________________________________________
separable_conv2d_3 (SeparableCo (None,23,23,32)  656     add_1[0][0]
__________________________________________________________________________________________________
batch_normalization_7 (BatchNor (None,23,23,32)  128     separable_conv2d_3[0][0]
__________________________________________________________________________________________________
activation_4 (Activation)       (None,23,23,32)  0       batch_normalization_7[0][0]
__________________________________________________________________________________________________
separable_conv2d_4 (SeparableCo (None,23,23,32)  1312    activation_4[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None,23,23,32)  128     separable_conv2d_4[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None,12,12,32)  512     add_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None,12,12,32)  0       batch_normalization_8[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None,12,12,32)  128     conv2d_4[0][0]
__________________________________________________________________________________________________
add_2 (Add)                     (None,12,12,32)  0       max_pooling2d_2[0][0]
                                                         batch_normalization_6[0][0]
__________________________________________________________________________________________________
separable_conv2d_5 (SeparableCo (None,12,12,64)  2336    add_2[0][0]
__________________________________________________________________________________________________
batch_normalization_10 (BatchNo (None,12,12,64)  256     separable_conv2d_5[0][0]
__________________________________________________________________________________________________
activation_5 (Activation)       (None,12,12,64)  0       batch_normalization_10[0][0]
__________________________________________________________________________________________________
separable_conv2d_6 (SeparableCo (None,12,12,64)  4672    activation_5[0][0]
__________________________________________________________________________________________________
batch_normalization_11 (BatchNo (None,12,12,64)  256     separable_conv2d_6[0][0]
__________________________________________________________________________________________________
conv2d_5 (Conv2D)               (None,6,6,64)    2048    add_2[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None,6,6,64)    0       batch_normalization_11[0][0]
__________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None,6,6,64)    256     conv2d_5[0][0]
__________________________________________________________________________________________________
add_3 (Add)                     (None,6,6,64)    0       max_pooling2d_3[0][0]
                                                         batch_normalization_9[0][0]
__________________________________________________________________________________________________
separable_conv2d_7 (SeparableCo (None,6,6,128)   8768    add_3[0][0]
__________________________________________________________________________________________________
batch_normalization_13 (BatchNo (None,6,6,128)   512     separable_conv2d_7[0][0]
__________________________________________________________________________________________________
activation_6 (Activation)       (None,6,6,128)   0       batch_normalization_13[0][0]
__________________________________________________________________________________________________
separable_conv2d_8 (SeparableCo (None,6,6,128)   17536   activation_6[0][0]
__________________________________________________________________________________________________
batch_normalization_14 (BatchNo (None,6,6,128)   512     separable_conv2d_8[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None,3,3,128)   8192    add_3[0][0]
```

```
_________________________________________________________________________________________
max_pooling2d_4 (MaxPooling2D)  (None,3,3,128)   0       batch_normalization_14[0][0]
_________________________________________________________________________________________
batch_normalization_12 (BatchNo (None,3,3,128)   512     conv2d_6[0][0]
_________________________________________________________________________________________
add_4 (Add)                     (None,3,3,128)   0       max_pooling2d_4[0][0]
                                                         batch_normalization_12[0][0]
_________________________________________________________________________________________
conv2d_7 (Conv2D)               (None,3,3,7)     8071    add_4[0][0]
_________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None,7)         0       conv2d_7[0][0]
_________________________________________________________________________________________
predictions (Activation)        (None,7)         0       global_average_pooling2d_1[0][0]
=========================================================================================
Total params: 57,815
Trainable params: 56,359
Non trainable params: 1,456
_________________________________________________________________________________________
```

### 4.3.2 CNN Model Training

The training dataset was extended by using data augmentation. This increases the data by applying random transformations to the existing images, such as rotation, crop, zoom, flip, and so on. Besides allowing the model to train with more data, it also helps prevent overfitting and generalizes better the model.

Early stopping is used to avoid overfitting. It stops the training of the model when the error on the validation set gets higher than before.

The learning rate is reduced when validation loss has stopped improving.

The CNN was trained using the Adam optimizer. This optimization algorithm is an extension of the stochastic gradient descent. It has some benefits when compared to other algorithms, to mention a few, it requires little memory, it is computationally efficient, and it is well suited for problems with large data and parameters [49].

#### Model loss and accuracy

A loss function needs to be used to calculate the error of the model. The one used in this model is the categorical cross-entropy. It indicates the difference between the distribution of the predictions with the actual distribution. The loss value of a model indicates how well the model performs after each iteration of optimization.

Figure 4.6 displays a graph where it is possible to analyze the training and validation loss in relation to the number of epochs. These values gradually decrease over time, which means that the model progressively improves in performance. However, most of the improvement is achieved until around epoch 30 and the loss stabilizes after that.

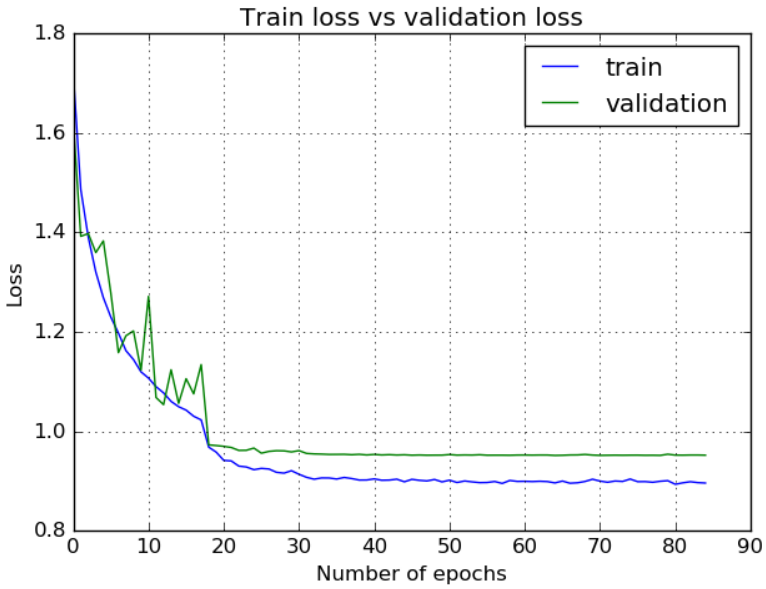


**Figure 4.6:** Train and validation loss during model training.

To determine the accuracy of the model, the test samples are used to determine the number of mistakes made by the model considering the true categories. The training and validation accuracy in relation to the number of epochs can be visualized in Figure 4.7. It is possible to notice that the accuracy continuously increases in the first epochs and stabilizes around epoch 30. The model achieves 65% accuracy in the validation set.

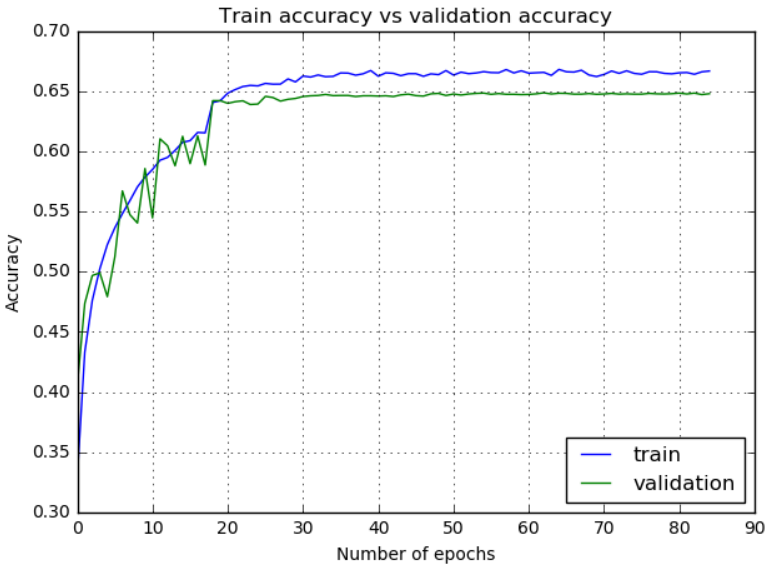


**Figure 4.7:** Train and validation accuracy during model training.

### 4.3.3 Visualization of Filters and Layers

The filters learned by the CNN model can be visualized to understand the types of features the model detects. The feature maps that result from the convolutional layers can also be analyzed to comprehend what specific features were detected for a given input image.

The filters of a CNN are a two-dimensional representation of the learned weights. The images used for training are grayscale images, so the filters have a depth of only one.

In Figure 4.8 are represented the eight 3x3 filters used on the first convolution layer of the model. The dark squares in the filters represent small weights and, contrariwise, lighter colored squares represent larger weights.

Feature maps are the result of applying filters to the input image or to another feature map. Visualizing feature maps for a specific input could help understand which features on the input are detected.

When the filters from the first convolution layer in Figure 4.8 are applied to the input image 4.9, the result is eight feature maps, each corresponding to the convolution of each filter applied to the input, see Figure 4.10.

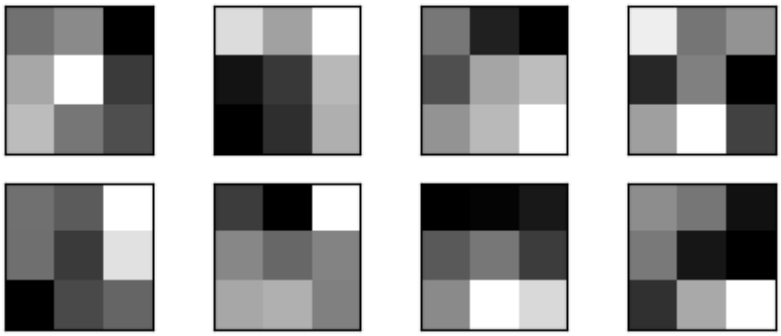

**Figure 4.8:** CNN filters learned for the first convolution layer.

**Figure 4.9:** Test input image after preprocessing of face detection, conversion to grayscale and rescaling.

**Figure 4.10:** Feature maps that result from the first convolution layer.

While feature maps closer to the input image detect small details, feature maps closer to the model's output detected more general features. Figure 4.11 displays the feature maps that result from one of the convolution layers in the middle of the model. These feature maps show less detail as a result of the model's continuous abstraction of the features into more general concepts.

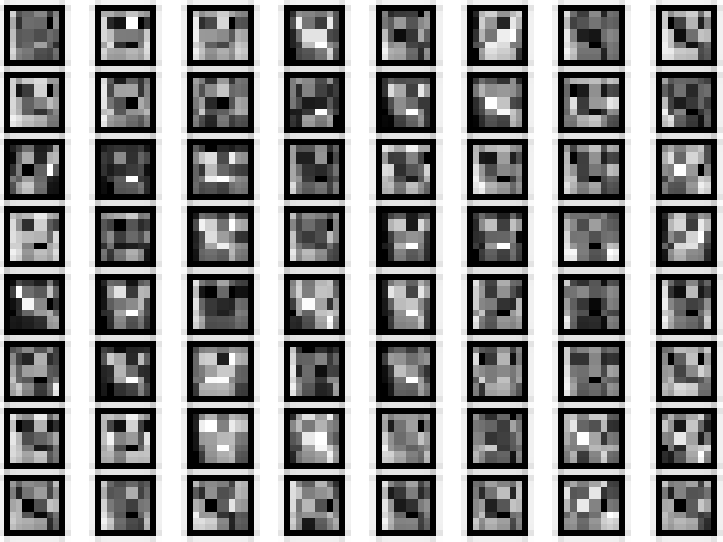

**Figure 4.11:** Feature maps that result from a convolution layer in the middle of the CNN model.

# Chapter 5

# Upper Body Movements Emotion Recognition

Emotion recognition through upper body movements will also be implemented taking a deep learning approach and thus, training a CNN model to be able to classify emotions based on body gestures.

## 5.1 Training Dataset

To train this CNN model only the FABO dataset is used. This dataset is already described in Section 4.2. However, to train this model the entirety of the images is used, not just the face of the subjects. Figure 5.1 shows some frames displaying each of the seven emotions being studied.

The images are rescaled from their original 1024 x 768 dimensions to 128 x 96 in order to ease and fasten the training process. They were also transformed from RGB to grayscale. The pixel values were normalized to a range between -1 and 1. And the data was split into train and validation data with an 80/20 ratio.

## 5.2 Convolution Neural Network Model

Once again, a CNN model is trained, but, this time, the goal is to generate a model that will accurately detect human emotions based on a person's body gestures.

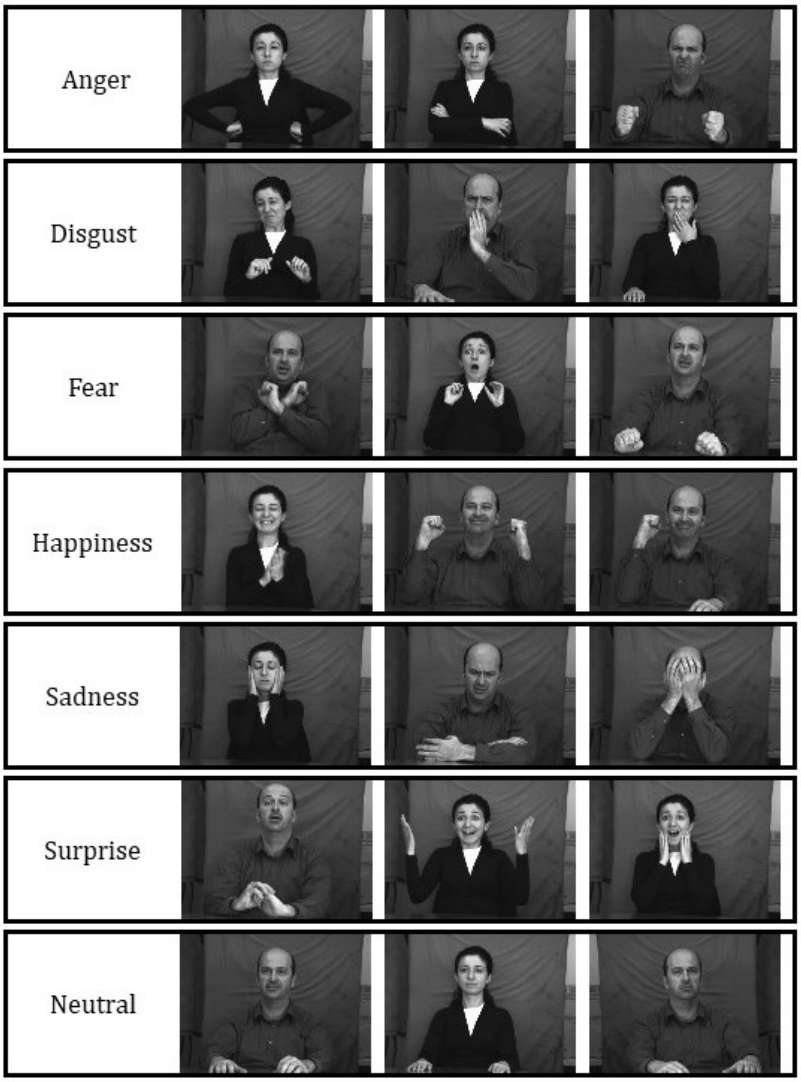


**Figure 5.1:** FABO sample images for each emotion.

### 5.2.1 CNN Architecture

The architecture implemented for this model is the same already described in Section 4.3. A simple architecture that uses depth-wise separable convolutions and residual modules. And instead of the standard fully connected layers, uses global averaging pooling as the final layer.

When implemented by Arriaga et al. [28], this architecture was used to train both emotion and gender classification based on face images. Here, the same architecture will

be used to train emotion classification based on body movements.

### 5.2.2 CNN Model Training

The same strategies as before are used on this model. Data augmentation is used to increase the number of training samples. As an attempt to avoid overfitting of the data, early stopping is used. And the Adam optimizer is applied.

#### Model loss and accuracy

Figure 5.2 shows the error of both the train and validation sets throughout each epoch. The loss value for the training data gradually decreases over time in a stable manner and reaches the lowest value at epoch 35. On the other hand, the loss value of the validation data reaches very uncertain values in the first epochs, going very rapidly from low to high loss values. However, it does begin to stabilize at epoch 18 and, after that, decreases slowly and steadily.

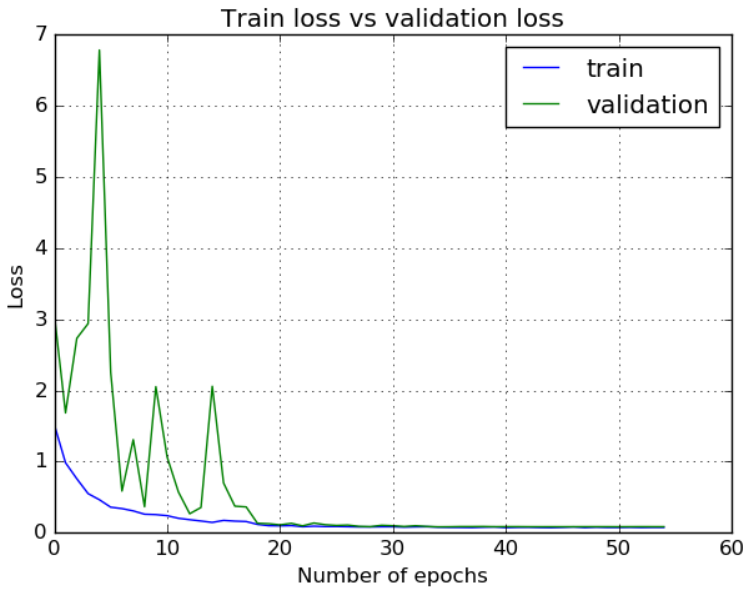


**Figure 5.2:** Train and validation loss during model training.

The training and validation accuracy are displayed on the graph of Figure 5.3. The same phenomenon as before also happens regarding the accuracy of the model. The training data accuracy increases gradually while the validation accuracy suffers some unex-

pected highs and lows, nevertheless, it also stabilizes after epoch 18 and attains a remarkable 96% accuracy value.

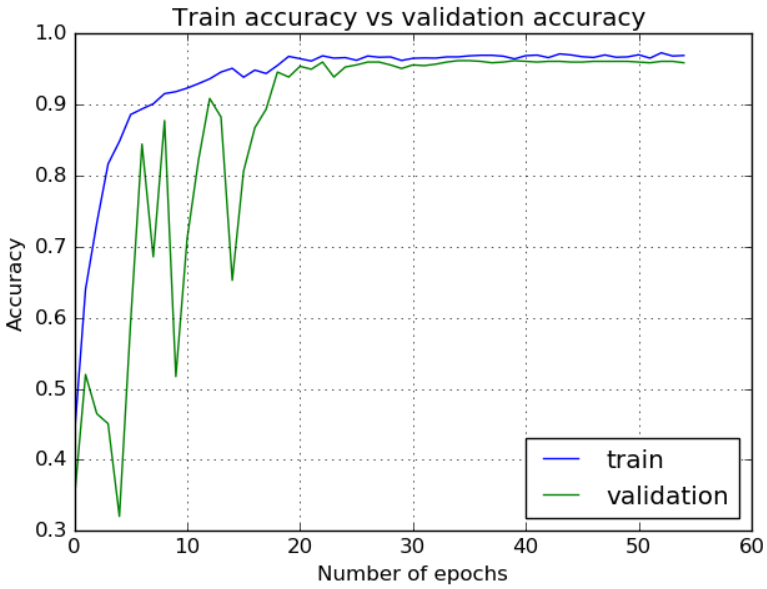


**Figure 5.3:** Train and validation accuracy during model training.

### 5.2.3 Visualization of Filters and Layers

Filters in CNNs look for specific patterns on the image, regardless of their placement on the image. The filters are shifted various times through the image until they have been applied to every pixel. Figure 5.4 shows the filters learned for the first convolution layer of the network. The colors represent the weights that are applied to pixels as the filter is passed through the image.

When these filters are applied to the input image in Figure 5.5 the result is the feature maps in Figure 5.6.

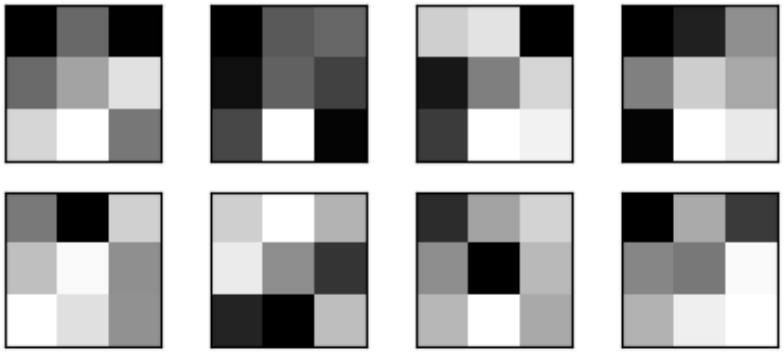

**Figure 5.4:** CNN filters learned for the first convolution layer.

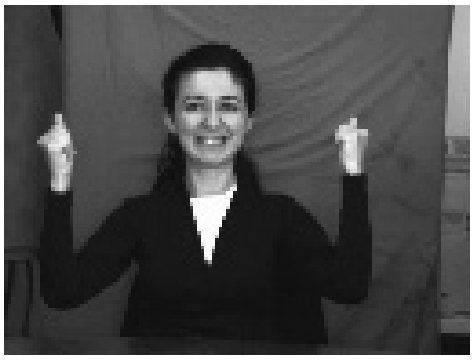

**Figure 5.5:** Test input image after conversion to grayscale and rescaling.

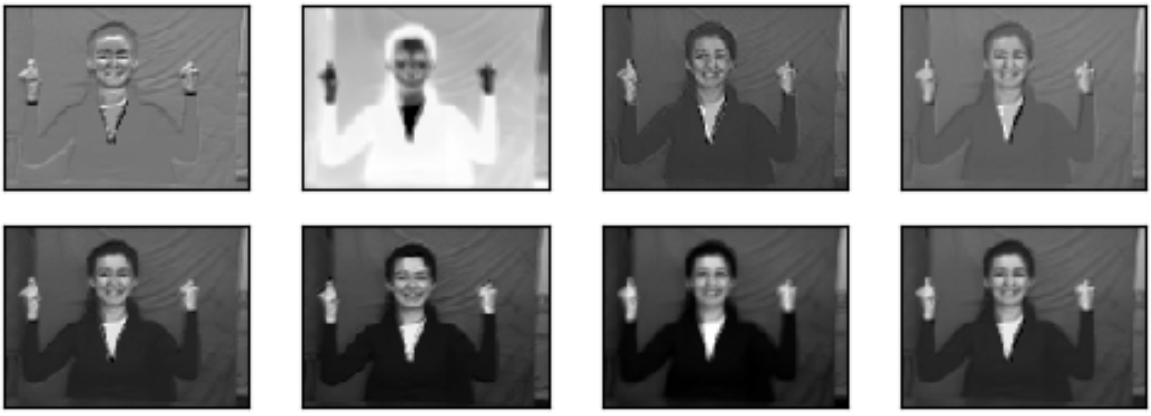

**Figure 5.6:** Feature maps that result from the first convolution layer.

# Chapter 6

# Bimodal Emotion Recognition

As stated before, the fusion of different modalities of emotion recognition is capable of achieving greater results than single modalities. Because of that, the studied modalities will be fused together to discover if, with this implementation, that is also true.

For this, a fusion method needs to be applied, to integrate single modalities into a combined representation.

## 6.1 Fusion methods

One of the issues in multimodal emotion recognition is deciding when to combine the information. There are a few different techniques to fuse the emotion recognition results of different modalities. Two different approaches are fusion at feature-level and fusion at decision-level [37].

Feature-level fusion combines the data from both modalities before classification. A single classifier is used containing features from both modalities. In contrast, in decision-level fusion, a separate classifier is used for each of the modalities, and the output results of each modality are combined following a defined norm.

Fusion at feature-level is usually not the preferred method since the concatenation of features from multiple modalities might result in high dimensional data. Besides, decision-level fusion allows greater flexibility regarding the modeling of each of the modalities. This permits that each modality is trained in different datasets and using different models for feature extraction and classification, so each of them can be better suited to the modality at hand.

## 6.2 Implementation

In this work, a CNN model was trained for each of the studied modalities. So, taking a decision-level fusion approach, each of these models will be used to classify each of the modalities and, in the end, the result of both classifications will be merged.

Two different decision-level fusion methods will be tested, an average method and a product method. In the average method, the average is calculated between both modalities and for each of the emotions. In the product method, the product of the probabilities of each modality is calculated for each of the emotions.

Let us assess an example, Figure 6.1 is a frame from a video displaying the emotion of sadness.

**Figure 6.1:** Frame showing the sadness emotion.

In Table 6.1 are indicated the results of each classification. The values show the probability of the image belonging to each of the emotion classes. The selected classification will correspond to whichever class has the highest probability value.

The facial expression recognition model classifies the image as representing the sadness emotion. Additionally, the upper body emotion recognition model classifies it as displaying the emotion of sadness as well.

Using the average method, the system outputs the values on the fourth column of the table, taking into account these values it is possible to conclude that the subject is experiencing the emotion of sadness.

On the other hand, the product method returns the values in the last column. Without a doubt, sadness is the emotion with the greatest value and, therefore, the one attributed to the image.

| Modality / Emotion | Facial Expression | Upper Body Movements | Bimodal Average | Bimodal Product |
|---|---|---|---|---|
| Anger | 0.134 | 0.000 | 0.067 | 0.000 |
| Disgust | 0.008 | 0.000 | 0.004 | 0.000 |
| Fear | 0.076 | 0.000 | 0.038 | 0.000 |
| Happiness | 0.006 | 0.010 | 0.008 | 0.000 |
| Sadness | **0.770** | **0.887** | **0.829** | **0.683** |
| Surprise | 0.001 | 0.102 | 0.051 | 0.000 |
| Neutral | 0.006 | 0.000 | 0.003 | 0.000 |

**Table 6.1:** Recognition results for each modality for the example input image.

# Chapter 7

# Results

The different classification models will be evaluated with different evaluation metrics, thus making it possible to conclude which modalities work best on emotion recognition and how the bimodal fusion of both modalities performs.

## 7.1 Evaluation Metrics

Different attributes can be used to analyze the performance of this system, such as precision, recall, accuracy, and F1-score.

To calculate those metrics, one must first determine the number of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN) for each of the emotion categories. A True Positive means an emotion that was correctly identified, on the other hand, a False Positive represents an incorrect classification of a specific emotion. In contrast, a False Negative signifies an emotion that was not correctly classified, and a True Negative symbolizes an emotion that was correctly not classified as the emotion being analyzed. These metrics have to be calculated in relation to a single emotion class.

Precision expresses the proportion of video samples that the system identified as a specific emotion that are actually related to that specific emotion, that is, the fraction of video samples that are correctly classified.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7.1}$$

Recall or True Positive Rate demonstrates the ability of the system to classify correctly

all of the emotions within the dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.2}$$

Accuracy demonstrates how close the number of detections, of a specific emotion, is to the actual true number.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{7.3}$$

F1-score relates precision and recall, it is used to get a balance between those measures. It is calculated as the weighted average of precision and recall, hence, it considers both false positives and false negatives. F1-score can become more useful than accuracy in cases where the distribution between classes in uneven. The best value to achieve is 1 and the worst is 0.

$$\text{F1-score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{7.4}$$

Another important way of measuring the performance of a classification system is the confusion matrix. It is a table layout that allows the visualization of how the samples from each category are classified and also, using the confusion matrix, it is possible to calculate the evaluation metrics described above.

## 7.2 Facial Expression Recognition

The normalized confusion matrix in Figure 7.1 shows the percentage of image samples that are of a specific emotion (true label) and that are classified as corresponding to a certain emotion (predicted label). The confusion matrix shows the best classification results corresponding to the anger and neutral emotions with 94% and 90%, respectively. The worst classification result corresponds to the surprise emotion that is often mistaken with fear, however, fear almost never is misclassified as surprise.

Figure 7.2 shows the same confusion matrix but with raw values. With these values, it is possible to calculate the various evaluation metrics that follow.

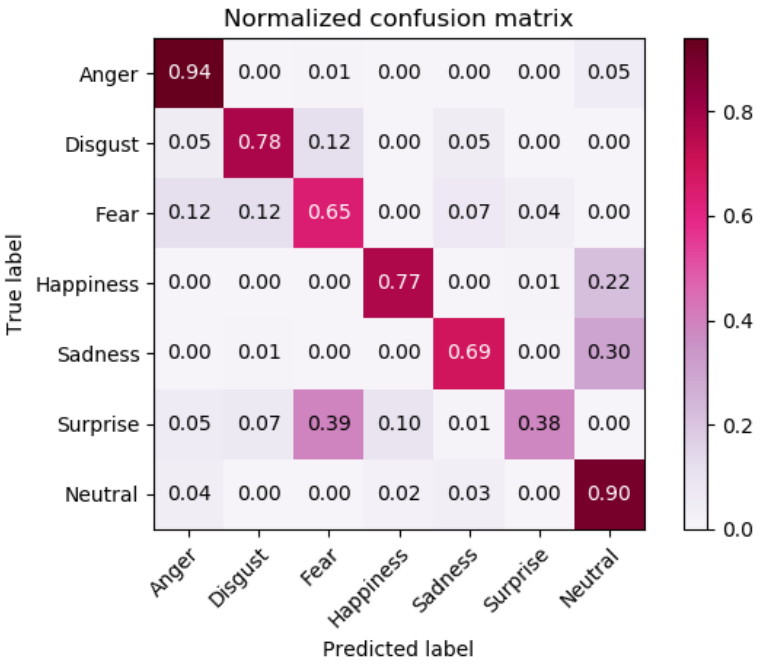


**Figure 7.1:** Normalized confusion matrix for facial expression recognition.

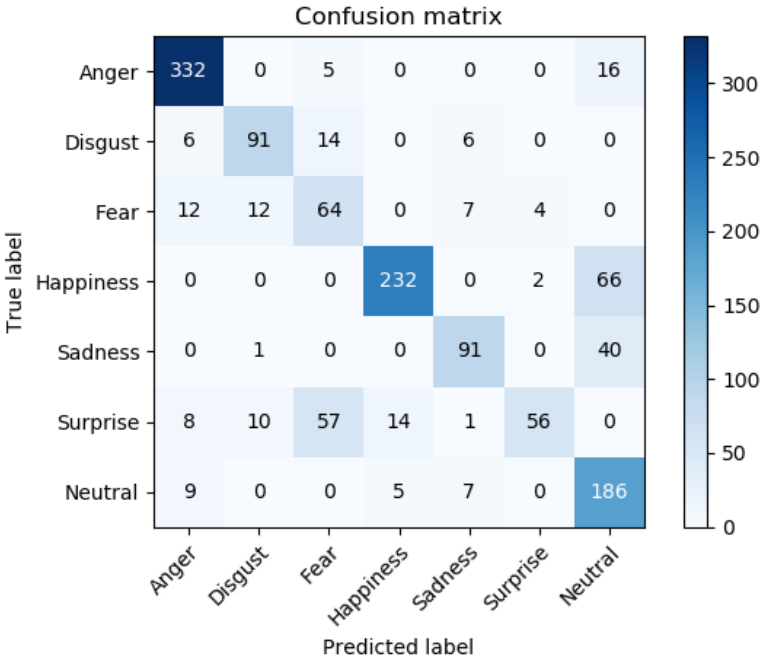


**Figure 7.2:** Confusion matrix for facial expression recognition.

- Precision = 77,2%
- Recall = 73%
- F1-score = 72,8%
- Accuracy = 77,7%

These values represent the average of the values for each emotion.

## 7.3 Upper Body Movements Emotion Recognition

Figure 7.3 displays the normalized confusion matrix for the upper body movements emotion recognition. This confusion matrix shows the values of the true positive rate, hence, the percentage of samples from each dataset that are classified correctly. Alternatively, Figure 7.4 illustrates a confusion matrix with the actual number of samples that were classified.

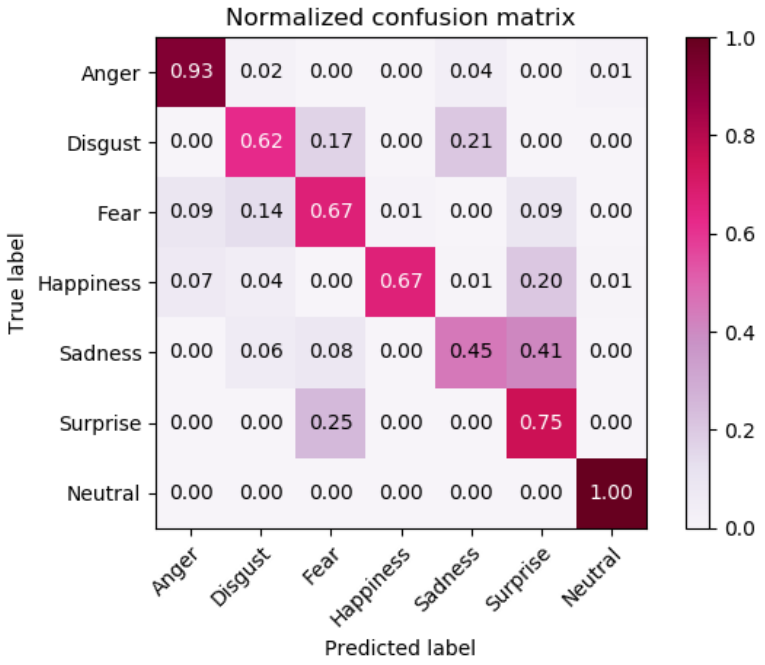


**Figure 7.3:** Normalized confusion matrix for upper body movements emotion recognition.

From Figure 7.3 it is possible to observe that, once again, the best recognition results are attributed to the anger and neutral emotions. Comparatively to the facial expression recognition, in this recognition modality, the surprise emotion has a much better recognition rate and is less often mistaken by fear. Also, the sadness emotion is quite often misclassified as surprise.

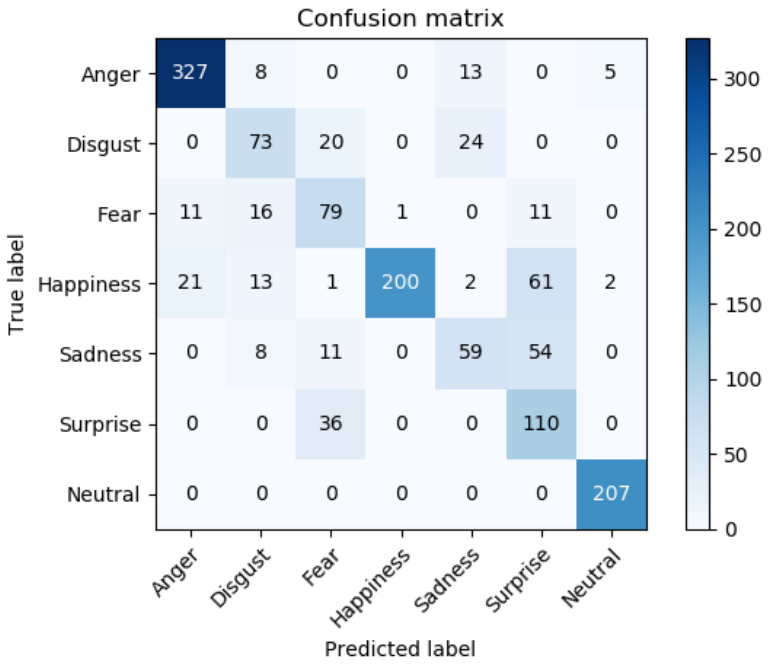


**Figure 7.4:** Confusion matrix for upper body movements emotion recognition.

The values from each of the evaluation metrics considered are listed below and correspond to the average of the values for each class of emotion. The achieved accuracy has only a 0.9% decrease compared to the facial expression recognition, which means that these modalities actually contribute a similar amount to the recognition of emotions and neither overpowers the other.

- Precision = 72, 8%
- Recall = 72.7%
- F1-score = 71, 5%
- Accuracy = 76, 8%

## 7.4 Bimodal Emotion Recognition

Finally, the combination of both modalities produces the results in Figures 7.5 and 7.6 using the average fusion method, and Figures 7.7 and 7.8 using the product fusion method.

### Average Fusion Method

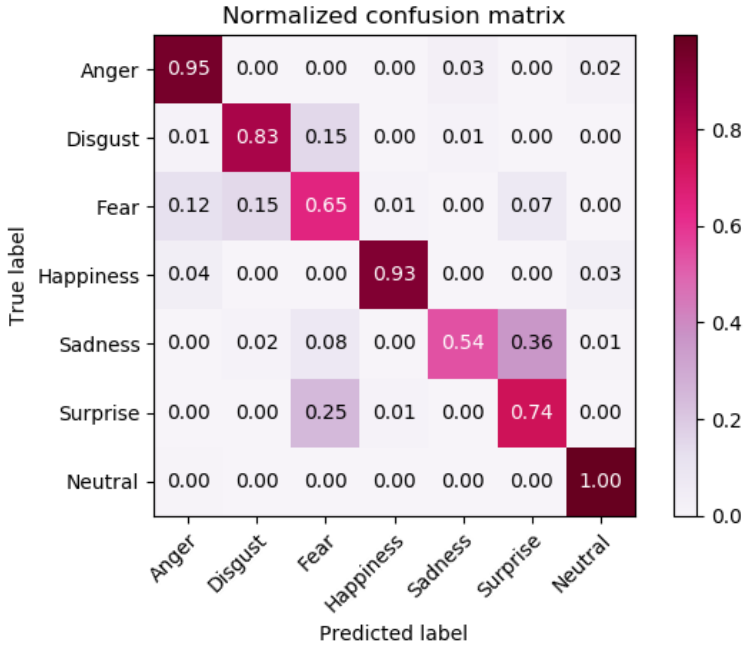


**Figure 7.5:** Normalized confusion matrix for bimodal emotion recognition using the average fusion method.

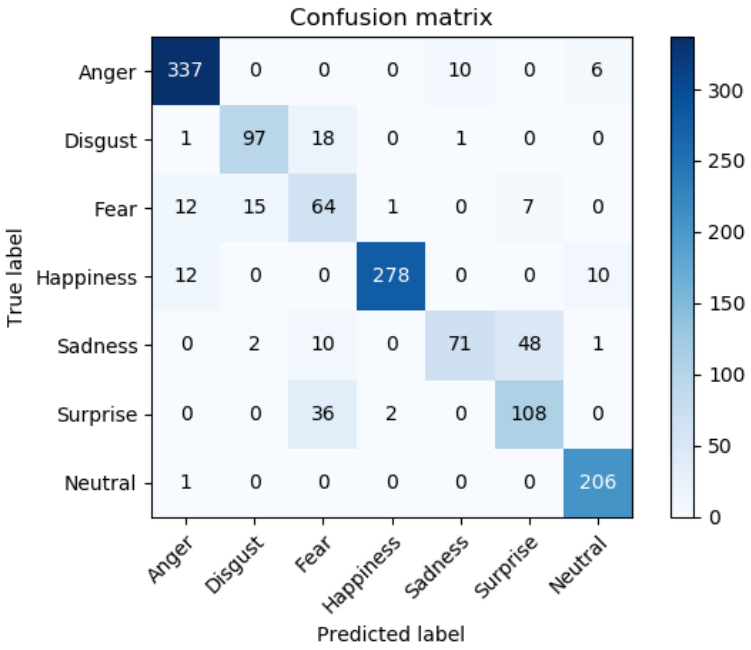


**Figure 7.6:** Confusion matrix for bimodal emotion recognition using the average fusion method.

The results of the different metrics correspondent to the average fusion method are the following.

- Precision = 81,7%
- Recall = 80,4%
- F1-score = 80,3%
- Accuracy = 85,7%

### Product Fusion Method

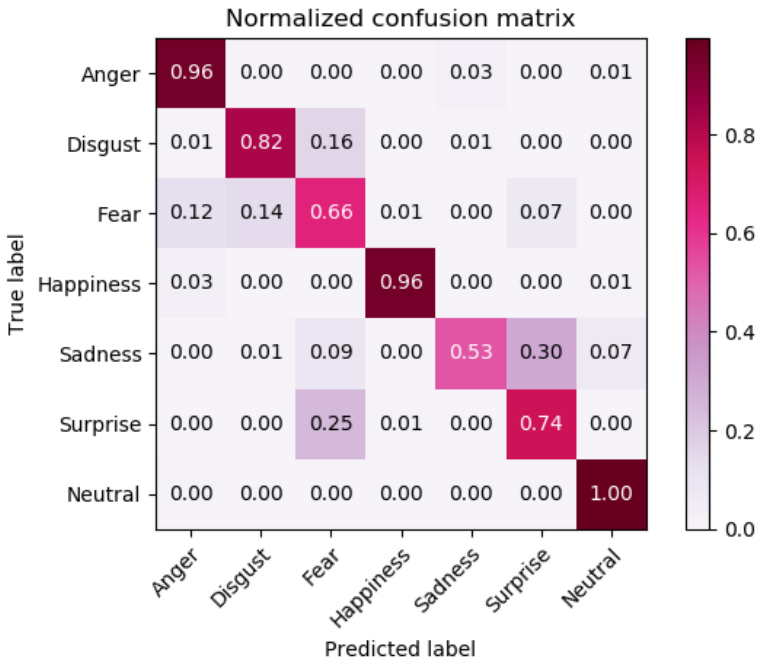


**Figure 7.7:** Normalized confusion matrix for bimodal emotion recognition using the product fusion method.

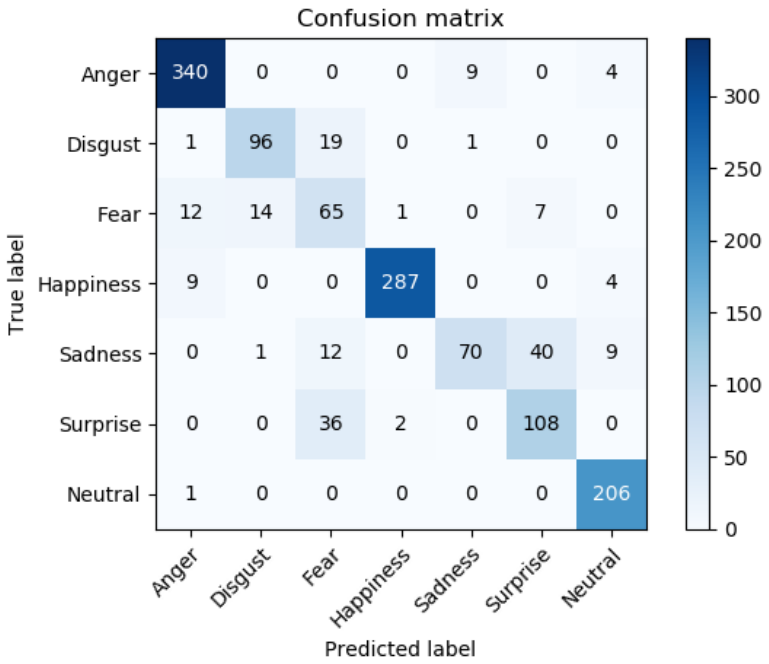


**Figure 7.8:** Confusion matrix for bimodal emotion recognition using the product fusion method.

Below are the evaluation metrics that result from the product fusion method.

- Precision = 82, 6%
- Recall = 80, 9%
- F1-score = 80, 9%
- Accuracy = 86, 6%

**Remarks**

It is possible to observe, from the confusion matrices, that the bimodal approach shows better results than the monomodal approaches, regardless of the fusion method. In this case, the best recognition rates correspond to the anger, happiness, and neutral emotions on both fusion methods. The worst recognition rate is attributed to the sadness emotion that is often misclassified as surprise.

All the evaluation metrics being considered have greater values with this approach. Accuracy shows a big improvement from 77, 7% and 76, 8% on the facial and upper body movements emotion recognition, respectively, to 85, 7% and 86, 6% on the fusion of both modalities.

Furthermore, the product fusion method shows slightly better results on all the evaluation metrics when compared to the average fusion method.

## 7.5 Comparison with Similar Systems

Table 7.1 displays the accuracy results for some similar systems developed also exploring facial expression and body movements to recognize emotions.

Gunes and Piccardi [50] used conventional methods to develop their solution, consequently, their results are inferior to the ones achieved in this work, for all modalities.

Chen et al. [51] also took a conventional approach and used appearance features combined with local motion on their solution. Their accuracy values are poorer, on all modalities, to the ones achieved on this thesis.

On the other hand, Barros et al. [12] also used CNNs on their solution, however, they also incorporated temporal features on their implementation. Their work shows inferior accuracy relative to the monomodal approaches, however, it shows a better recognition accuracy on the fusion of both modalities. This might have to do with the fact that temporal features are included.

| Modality / Approach | Facial Expression | Upper Body Movements | Bimodal |
|---|---|---|---|
| This Solution | 77,7% | 76,8% | 86,6% |
| Gunes and Piccardi [50] (AdaBoost) | 35,2% | 73,1% | 82,7% |
| Chen et al. [51] | 66,5% | 66,7% | 75,0% |
| Barros et al. [12] | 72,7% | 57,8% | 91,3% |

**Table 7.1:** Comparison between the accuracy results from each of the modalities for different systems.

# Chapter 8

# Conclusion

In this thesis, the focus was to implement a system that was able to recognize human emotions. To accomplish that, two different modalities of displaying emotions were researched, facial expression and upper body movements. Together, these two non-verbal modalities are able to transmit most of the information needed to discern between the seven considered standard emotions.

Convolutional Neural Networks were used to train recognition models for each of the studied modalities. The results returned from both modalities were later fused together to output a final recognition result. Two different fusion methods were tested, an average and a product fusion methods. The one that achieved greater accuracy was the product fusion method.

Overall, the developed system achieves a good recognition rate, achieving 86,6% accuracy on the bimodal approach. It surpasses the recognition rate of conventional approaches, as expected from the literature. It does achieve better results in single modalities recognition than the state-of-the-art research done by Barros et al., however, in the fusion of both modalities, this solution falls short.

To try to improve the results achieved, some more investigation could be done towards finding the best CNN architecture possible to achieve even greater results.

Furthermore, different datasets could be included in the training of the CNN models, which would allow for a better generalization of the model.

Different fusion methods could also be researched, the ones used were quite simplistic, perhaps a more elaborate fusion method could be able to attain better results.

For future work, it would be interesting to apply this implementation in a real-world application to verify how well it performs with real-world data.

# Bibliography


[1] Anderson, K., & McOwan, P. W. (2006). A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36*(1), 96-105.

[2] Breazeal, C. (2003). Emotion and sociable humanoid robots. *International journal of human-computer studies, 59*(1-2), 119-155.

[3] Augello, A., Dignum, F., Gentile, M., Infantino, I., Maniscalco, U., Pilato, G., & Vella, F. (2018). A social practice oriented signs detection for human-humanoid interaction. *Biologically inspired cognitive architectures, 25, 8-16.*

[4] Kim, K., Cha, Y. S., Park, J. M., Lee, J. Y., & You, B. J. (2011). Providing services using network-based humanoids in a home environment. *IEEE Transactions on Consumer Electronics, 57(4), 1628-1636.*

[5] Sorbello, R., Chella, A., Calí, C., Giardina, M., Nishio, S., & Ishiguro, H. (2014). Telenoid android robot as an embodied perceptual social regulation medium engaging natural human–humanoid interaction. *Robotics and Autonomous Systems, 62(9), 1329-1341.*

[6] Hossain, M. S., & Muhammad, G. (2015). Cloud-assisted speech and face recognition framework for health monitoring. *Mobile Networks and Applications, 20(3), 391-399.*

[7] Alhussein, M. (2016). Automatic facial emotion recognition using weber local descriptor for e-Healthcare system. *Cluster Computing, 19(1), 99-108.*

[8] Noroozi, F., Marjanovic, M., Njegus, A., Escalera, S., & Anbarjafari, G. (2017). Audio-visual emotion recognition in video clips. *IEEE Transactions on Affective Computing.*

[9] Zhang, Y., Wang, Z. R., & Du, J. (2019). Deep Fusion: An Attention Guided Factorized Bilinear Pooling for Audio-video Emotion Recognition. *arXiv preprint arXiv:1901.04889.*

[10] Gunes, H., & Piccardi, M. (2007). Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications, 30*(4), 1334-1345.

[11] Shan, C., Gong, S., & McOwan, P. W. (2007, September). Beyond Facial Expressions: Learning Human Emotion from Body Gestures. In *BMVC* (pp. 1-10).

[12] Barros, P., Jirak, D., Weber, C., & Wermter, S. (2015). Multimodal emotional state recognition using sequence-dependent deep hierarchical features. *Neural Networks, 72*, 140-151.

[13] Sun, B., Cao, S., He, J., & Yu, L. (2018). Affect recognition from facial movements and body gestures by hierarchical deep spatio-temporal features and fusion strategy. *Neural Networks, 105*, 36-51.

[14] Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology, 17*(2), 124.

[15] Friesen, W. V., & Ekman, P. (1976). *Pictures of facial affect.* Consulting psychologists press.

[16] Mehrabian, A. (1971). *Silent messages* (Vol. 8). Belmont, CA: Wadsworth.

[17] Hemalatha, G., & Sumathi, C. P. (2014). A study of techniques for facial detection and expression classification. *International Journal of Computer Science and Engineering Survey, 5*(2), 27.

[18] Kliemann, D., Rosenblau, G., Bölte, S., Heekeren, H. R., & Dziobek, I. (2013). Face puzzle—two new video-based tasks for measuring explicit and implicit aspects of facial emotion recognition. *Frontiers in psychology, 4*, 376.

[19] Ooi, C. S., Seng, K. P., Ang, L. M., & Chew, L. W. (2014). A new approach of audio emotion recognition. *Expert systems with applications, 41*, 5858-5869.

[20] Piana, S., Stagliano, A., Odone, F., Verri, A., & Camurri, A. (2014). Real-time automatic emotion recognition from body gestures. *arXiv preprint arXiv:1402.5047.*

[21] Yan, J., Zheng, W., Xin, M., & Yan, J. (2014). Integrating facial expression and body gesture in videos for emotion recognition. *IEICE TRANSACTIONS on Information and Systems, 97*(3), 610-613.

[22] Ko, B. (2018). A brief review of facial emotion recognition based on visual information. *sensors, 18*(2), 401.

[23] Happy, S. L., George, A., & Routray, A. (2012, December). A real time facial expression classification system using local binary patterns. In *4th International conference on intelligent human computer interaction (IHCI)* (pp. 1-5). IEEE.

[24] Ghimire, D., Jeong, S., Lee, J., & Park, S. H. (2017). Facial expression recognition based on local region specific features and support vector machines. *Multimedia Tools and Applications, 76*(6), 7803-7821.

[25] Khorrami, P., Le Paine, T., Brady, K., Dagli, C., & Huang, T. S. (2016, September). How deep neural networks can improve emotion recognition on video data. In *2016 IEEE international conference on image processing (ICIP)* (pp. 619-623). IEEE.

[26] Github (2019, March 20). Artificial neural networks, anytime. https://github.com/ifp-uiuc/anna

[27] Fan, L., & Ke, Y. (2017). Spatiotemporal Networks for Video Emotion Recognition. *arXiv preprint arXiv:1704.00570.*

[28] Arriaga, O., Valdenegro-Toro, M., & Plöger, P. (2017). Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*.

[29] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

[30] Kaggle (2019, March 29). Challenges in Representation Learning: Facial Expression Recognition Challenge. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

[31] Atkinson, A. P., Dittrich, W. H., Gemmell, A. J., & Young, A. W. (2004). Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception, 33*(6), 717-746.

[32] Wallbott, H. G. (1998). Bodily expression of emotion. *European journal of social psychology, 28*(6), 879-896.

[33] Noroozi, F., Kaminska, D., Corneanu, C., Sapinski, T., Escalera, S., & Anbarjafari, G. (2018). Survey on emotional body gesture recognition. *IEEE Transactions on Affective Computing.*

[34] Glowinski, D., Dael, N., Camurri, A., Volpe, G., Mortillaro, M., & Scherer, K. (2011). Toward a minimal representation of affective gestures. *IEEE Transactions on Affective Computing, 2*(2), 106-118.

[35] Bänziger, T., Mortillaro, M., & Scherer, K. R. (2012). Introducing the Geneva Multimodal expression corpus for experimental research on emotion perception. *Emotion, 12*(5), 1161.

[36] Barros, P., Parisi, G. I., Jirak, D., & Wermter, S. (2014, November). Real-time gesture recognition using a humanoid robot with a deep neural architecture. In *2014 IEEE-RAS International Conference on Humanoid Robots* (pp. 646-651). IEEE.

[37] Gunes, H., & Piccardi, M. (2005, October). Affect recognition from face and body: early fusion vs. late fusion. In *2005 IEEE international conference on systems, man and cybernetics* (Vol. 4, pp. 3437-3443). IEEE.

[38] Gunes, H., & Piccardi, M. (2006, August). A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior. In *18th International Conference on Pattern Recognition (ICPR'06)* (Vol. 1, pp. 1148-1153). IEEE.

[39] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., ... & Zhou, Y. (2013, November). Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing* (pp. 117-124). Springer, Berlin, Heidelberg.

[40] Pantic, M., Valstar, M., Rademaker, R., & Maat, L. (2005, July). Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo* (pp. 5-pp). IEEE.

[41] Tutorials Point (2019, May 14). Artificial Intelligence - Neural Networks. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm

[42] Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). San Francisco, CA, USA:: Determination press.

[43] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

[44] MathWorks (2019, May 16). Introduction to Deep Learning: What Are Convolutional Neural Networks? https://se.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

[45] CS231n Convolutional Neural Networks for Visual Recognition (2019, May 15). Convolutional Neural Networks (CNNs / ConvNets). https://cs231n.github.io/convolutional-networks/

[46] Code Project (2019, May 15). ANNT : Convolutional neural networks. https://www.codeproject.com/Articles/1264962/ANNT-Convolutional-neural-netwo

[47] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR (1), 1*, 511-518.

[48] Github (2019, May 06). OpenCV deep learning module samples. https://github.com/opencv/opencv/tree/master/samples/dnn#opencv-deep-learning-module-samples

[49] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[50] Gunes, H., & Piccardi, M. (2008). Automatic temporal segment detection and affect recognition from face and body display. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39*(1), 64-84.

[51] Chen, S., Tian, Y., Liu, Q., & Metaxas, D. N. (2013). Recognizing expressions from face and body gesture by temporal normalized motion and appearance features. *Image and Vision Computing, 31*(2), 175-185.