TU WIEN

Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

MASTER THESIS:

Machine Learning and Deep Learning for Emotion Recognition

A Thesis submitted by Joan Sisquella Andrés for the Master's degree in Industrial Engineering

---

Supervised by:
Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. M. Vincze
Dipl.-Ing. M. Hirschmanner

Contents

List of Figures

List of Tables

Abstract

The ability of robots to recognize emotions will be vital for the development of service robotics in the coming years. However, we have not achieved the recognition of certain emotions in a sufficiently effective way yet. Therefore, we have not introduced in the market products using it. If emotion recognition is a very challenging task even for people, it is even more for robots, since they do not feel any emotion and can not empathize. Machine learning and deep learning are being used for emotion recognition. However, can Machine learning and Deep learning recognize these facial expressions efficiently?

This work presents a comparison of current state-of-the-art learning strategies that can handle data and adapt classical static approaches to deal with images sequence. Machine learning algorithms and deep learning versions of CNN are evaluated and compared using different datasets for universal emotion recognition, where the performances are shown, and the pros and cons are discussed.

Acknowledgements

I would first like to thank my thesis advisor Matthias Hirschmanner of the Vision for Robotics Laboratory at the Automation and Control Institute, TU Wien. He consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge professor Markus Vincze for allowing me to work in his department during this time of completion of the thesis.

Finally, I must express my very profound gratitude to my parents, to my sister, and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Chapter 1

Introduction

In recent years service robotics has improved their performance, there will surely be a revolution with service robotics in the coming years like the one that occurred with industrial robotics. But before, it is necessary that robots, especially humanoids, perceive our emotions to be able to adapt to our needs.

Artificial intelligence has had increasing involvement in any scope of human life. The technologies are adapted to the needs of the human being and artificial intelligence is what makes this adaptation between technology and humans possible.

These techniques are used in algorithms for the recognition of human emotions. When humans try to communicate with other people a very high percentage is represented by non-verbal communication. Many studies show that facial expressions have a connection with human emotions. The ability of human beings to detect and identify these emotions makes it possible for us to understand each other. The main objective of this part of artificial intelligence is to use learning techniques in order to to get the machine capable of identifying these emotions.

Emotions play a fundamental role in human cognition and they are an essential field in studies of cognitive sciences like neuroscience or psychol-

ogy. They also play a role in marketing or product management as detecting happiness from a customer that their needs are satisfied. More specifically, emotions can be estimated by facial inference since there is undoubtedly universal connections between emotions and facial expressions, independently of races, genders, ages or cultures.

Robots do not feel any emotion, they do not have empathy with humans, they can not recognize the emotions that a person is feeling compared to their own, as humans do. The information available are large matrices that represent the images captured and other additional information the might have like sensors or microphone. It is from there where we have to start working. For this reason, the recognition of emotions by computer is so complicated.

To achieve this, we must first discover what emotions the human being really feels, what technology is the most appropriate to capture them, and which models and algorithms are the most effective. It is for all these variables that this problem has not yet been sufficiently solved.

This thesis tests some state-of-the-art machine learning and deep learning models for emotion recognition using different datasets. Key aspects of affect expression through body language are introduced. Then, the pipeline for automatically recognizing emotions is explained and technical aspects of each component of such pipeline are discussed. Furthermore, the results are shown and it is concluded with discussions and potential future lines of research.

In this thesis, we evaluate some machine learning and deep learning models including Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Multilayer Perceptron (MLP), and Convolutional Neural Networks (CNNs). The evaluation focuses on the task of emotion recognition through facial expression in image and video and audio files have been used too. The datasets (Karolinka Directed Emotional Faces (KDEF), Japanese Female Facial Expression (JAFFE), Facial Expression Recognition (FER),

and Multimodal Emotion Recognition in Polish (MERiP)) were used for emotion classification providing data for six basic emotions.

This project shows that with the two techniques it has been able to correctly predict at least 80% of the images with the best of the machine learning techniques and reaching 89% in deep learning.

1.1 Thesis outline

For achieving the main thesis objectives, the thesis outline is the following:

- Explore and explain the main datasets and understand their advantages and constraints.
- Collecting different datasets for emotion recognition on images, video, and audio.
- Discover how to create a Convolutional Neural Network and choose one model for it and for the other machine learning algorithms.
- Create a dataset combining different ones and get it reliably labeled.
- Build different solutions for preprocessing the data for training the models. Considering different data modalities as features: face landmarks (i.e. geometry of the face), CNN features on the face itself (spatial features).
- Analysis, implementation, and evaluation of a machine learning system for the recognition of expressions based on Python, Sklearn, OpenCV and Dlib.
- Analysis, implementation, and evaluation of a deep learning system for the recognition of expressions with Keras on images, audio, and video.

Chapter 2

Related Work

In recent years, the automatic recognition of emotions has increased his importance at the business level. As a result, the academic works related to this field have increased in number and quality of the results. Below is a set of related works that have served as inspiration during the development of this thesis.

2.1 Models of the human body and emotion

Body language contains different types of nonverbal indicators: facial expression, gesticulation, eye movement, contact and use of personal space [1].

The face is the main source of information, but much information can also be obtained from the hands [2]. For example, Pease proved that people who use open-handed gestures are perceived in a better way. Head positioning contains lots of information, some of this is useful to convey the emotion that the person is transmitting. It has been shown that people are prone to talk more if the listener encourages them by nodding [3].

2.1.1 Models of emotions

Different ways of modeling emotions have been subject of debate for a long time and different models were proposed. The most relevant models can be

classified into three main categories: categorical, dimensional and componential.

Categorical models: Classifying emotions into a set of distinct classes that can be recognized and described easily in daily language are the most common. Paul Ekman proposed a model based on the assumption that humans universally express and recognize a set of discrete primary emotions which are happiness, sadness, fear, anger, disgust, and surprise [4].

Dimension models: Another way of modeling emotions is by using a set of latent dimensions. Basically, these dimensions are valence (how pleasant or unpleasant a feeling is), activation (how likely is the person to take action under the emotional state) and control (the sense of control over the emotion). Because these types of models are continuous, they can describe more complex emotions. Unfortunately, the richness of the space is more difficult to use for automatic recognition systems.

Componential models: According to this theory, more complex emotions are combinations of pairs of more basic emotions. The best example of componential models was proposed by Plutchik [5]. For example, love is considered to be a combination of joy and trust. These types of models are not commonly used in automatic emotion recognition.

Due to the goal of this to automatically recognize emotions, it is less effective to choose a model that deals with fuzzy classes. The most useful model is the categorical one due to simplicity. The aim of the categorical model is to classify emotions into a set of classes easy to recognize. The categorical model is not possible to recognize expressions at different intensities and it is also not possible to combine different emotions such as, for example, angrily surprised.

2.1.2 Models of the human body

There are two basic ways of representing the human body in an abstract form: based on the rigid composition of the human body (Part Based Models), or a kinematic construction of the human skeleton (Kinematic Models).

Part based models: The human body can be represented as a composition of trunk, limbs, and face; as well face is composed by eyes, nose, and mouth [6].

Kinematic models: Another way of modeling the human body is by defining a collection of interconnected joints also known as kinematic chain models. The models can be planar, in which case they are a projection in the image plane or depth information can be considered as well [7].

2.2 Automatic emotion analysis

This section shows some of the advances that other researchers have made in the field of facial recognition systems.

Many methods depend on the extraction of the facial region. This can be done through manual inference [8] or with automatic detection [9]. Many methods often use the Facial Action Coding System (FACS), which describes facial expression using the Units of Action (AU). An Action Unit is a facial action like "raise your eyebrow." Multiple activations of AUs describe an expression [10]. Being able to detect AUs correctly is a useful step since it allows to evaluate the level of activation of the corresponding emotion.

The next step in the evolution of facial detection systems was to diverge towards Neural Networks. Yu & Zhang introduced for its Facial Expression Recognition (FER) system based on convolutional neural networks. In this case, they disturbed the input images in different ways in the training sequence and the output was the response of each test image as an averaged

voting of responses from all the perturbed samples [11].

Kim, Roh and Lee [12] introduced a variation the architecture of the network and the parameters that make it initialize the weights of the model using weights trained in another database to later apply a re-training. In solving a seven-class problem (categorical classification) of static FER for the EmotiW2015, they achieve a test accuracy of 61.6%.

Another way for emotion recognition is the Multimodal systems that use different channels of information based on face, voice and body gesture at the same time. Because of that, they are useful elements of perceptual user interfaces, they can also have applications in human-machine interfaces, which are used in intelligent machines that understand and react to human emotions [14]. If the system is capable of combining the emotional and social aspects of the situations for making a decision based on the available cues, it can be a useful assistant for humans [15].

The use of deep learning methods has begun to grow significantly in recent years, as they have proven to be truly effective and have exceeded the approaches of prior state of the art approaches. Their particularity is that they take into account non-linear features.

Ranganathan et al. introduced the emoFBVP database of multimodal recordings [16]. Actors displayed 23 different emotions, each emotion with three different intensities. Next, they propose a Convolutional Deep Belief Model (CDBN) for emotion recognition using his own dataset. They demonstrate that their model gives better recognition accuracies when recognizing low intensity or subtle expressions of emotions when compared to state of the art methods

CNNs have the ability to extract information from a sequence of inputs. Ronghe [17] propose a hybrid CNN-RNN architecture for emotion analysis. Firstly, the model is trained to classify images into one of the seven emotions. For solving the movement between frames, the videos are preprocessed and transformed into a sequence of feature vectors which are used to train an SVM model.

This is combined with a Multilayer Perceptron that models the correlation between features of the emotions from the images and speech. This provides more parameters to the RNN, for classifying the emotion reaction on each frame in the video, a schematic is shown in Figure 2.1.

Another important advance has been to prevent networks of overfitting, changing the dropout parameter in the Fully-Connected layers [13].

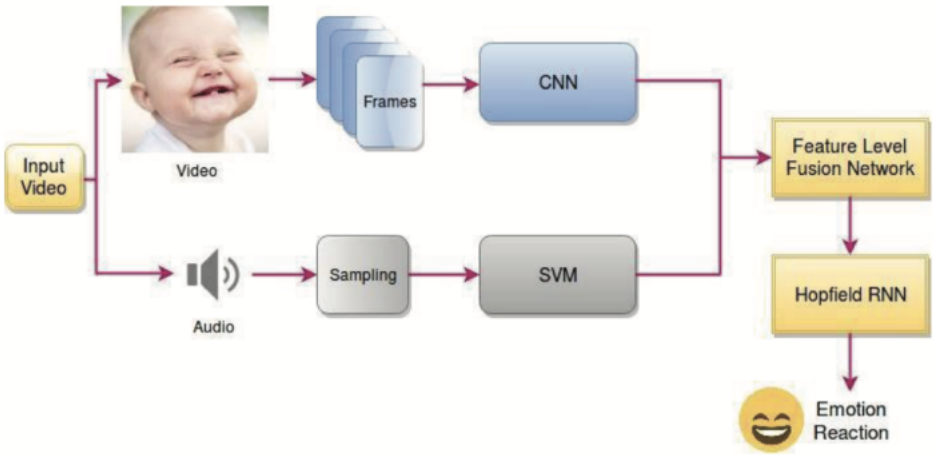

Figure 2.1: Scheme of the hybrid CNN-RNN model proposed by Ronghe et al. [17]

Chapter 3

Theoretical Framework

Machine learning is not only used for image recognition, but it also has applications as diverse as medical diagnosis, prediction systems, and financial services. Speech Recognition is a popular example of deep learning. In this part, the basis of these two concepts is explained.

3.1 Machine Learning

Machine learning aims to study the recognition of patterns and learning by computers. It makes it possible for the machines to learn without being explicitly programmed.

This discipline works with algorithms that can give relevant findings or conclusions obtained from a set of data, without the human being writing instructions or codes for this. The purpose of this discipline is that people and machines work hand in hand. Precisely this is what they do the algorithms, they allow the machines to execute tasks, both general as specific.

Learning process is done by classifiers. A classifier is an algorithm that, receiving as input certain information of an object, is able to indicate the category or class to which it belongs among a limited number of possible classes.

To evaluate four classifiers have been selected from some of the most representative families of algorithms:

3.1.1 Support Vector Machine [SVM]

Support Vector Machines are a set of supervised learning algorithms [18]. These methods are properly related to problems of classification and regression. Given a set of training examples (from samples), we can label the classes and train an SVM to build a model that predicts the class of a new sample. A SVM is a model that represents the sample points in space, separating the classes into spaces as wide as possible. This is done by a hyperplane. When the new samples are put in correspondence with the model, depending on the spaces to which they belong, they can be classified to one or the other class.

More clearly, this model represents the sample points in the space, separating the classes by a hyperplane in a space of very high dimensions that can be used on problems of classification or regression. A clear separation between the classes will allow a correct classification, a scheme is shown in Figure 3.1.

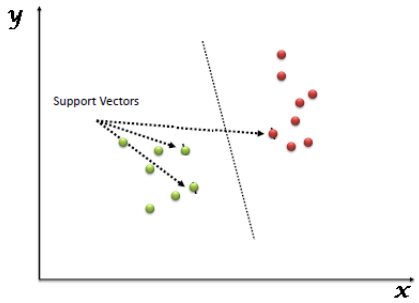

Figure 3.1: SVM feature selection[1].

[1]https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

3.1.2 Decision Tree [DT]

Learning based on decision trees is one of the predictive modeling approaches used in statistics, data mining and machine learning [19]. Tree models, where the target variable can take a finite set of values are called classification trees. In these tree structures, the leaves represent class labels and the branches represent the conjunctions of characteristics that lead to those class labels. Decision trees where the target variable can take continuous values are called regression trees.

The purpose of this classifier is to create a model that predicts the value of an objective by learning rules of simple decisions inferred from the characteristics of the data.

3.1.3 Random Forest [RF]

The essential idea of the random forest is to average many noisy but approximately impartial models, and therefore reduce the variation. Trees are ideal candidates for bagging since they can record complex interaction structures in the data, and if they grow sufficiently deep, they have a relatively low bias. Trees are notoriously noisy, they benefit greatly when averaging [20].

Each tree is constructed using the following algorithm:

- Being N the number of test cases, M is the number of variables in the classifier.
- Being m the number of input variables to be used to determine the decision in a given node; m must be much smaller than M.
- Choose a training set for this tree and use the rest of the test cases to estimate the error.
- For each node of the tree, randomly choose m variables on which to base the decision. Calculate the best partition of the training set from m variables.

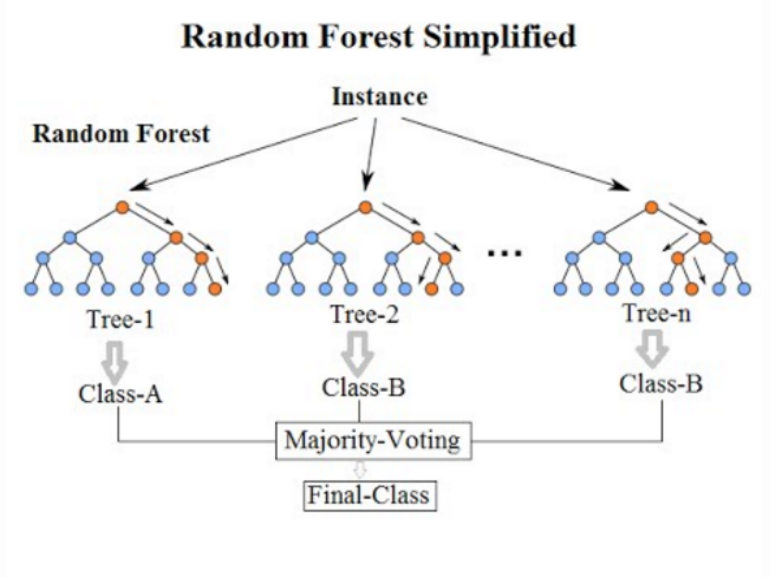

Figure 3.2: Scheme of RF performance[2].

For prediction, a new case is pushed down the tree. Then it is assigned the label of the terminal node where it ends. This process is iterated by all the trees in the assembly, and the label that gets the most incidents is reported as the prediction. So, the estimator fits a series of decision tree classifier in several sub-samples of the database and uses the average to improve predictive accuracy. A scheme of the performance is shown in Figure 3.2.

3.1.4 Multilayer Perceptron [MLP]

The multilayer perceptron is an artificial neural network formed by multiple layers, in such a way that it has the capacity to solve problems that are not linearly separable, which is the main limitation of the perceptron [21]. The Multilayer Perceptron can be totally or locally connected.

A multilayer perceptron has three layers of nodes: an input layer, a hid-

[2]https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

den layer, and an output layer. Except for the input node, each node uses a nonlinear activation function, which allows classification of data that is not linearly separated. MLP uses a supervised learning technique called backpropagation.

Backpropagation is a method that is used to calculate the gradient that is necessary to apply to the weights or coefficients of the nodes in the network. The algorithm is the following:

- Initialization of the model giving a random value to the weights.
- Calculation of the values of the output.
- Calculation of the loss function (obtained output less desired output).
- Backpropagation of the errors of the output towards the input.
- Recalculate the output values and repeat the algorithm until the stabilization of the values.

3.2 Deep Learning

Deep Learning is a branch of artificial intelligence that deals with emulating the way that humans use to obtain a certain kind of knowledge. It is a way of automatizing the predictive analysis. It is based on a biological model of the brain proposed by Nobel laureates Hubel and Wiesel in 1959. It is formed by layers and neurons that learn from the lower layers to the higher ones with the ability to extract abstract concepts.

Deep learning algorithms have a growing complexity hierarchy. Each algorithm applies a nonlinear transformation in his input and uses what it learns to create a statistical model as output. There are iterations until the output has an acceptable level of accuracy.

In deep learning, the model learns by itself discovering the relations between the variables. Humans do not intervene, the model is capable of

making the feature selection.

Sometimes it is complicated to extract high-level abstract features from raw data. Deep Learning solves this problem expressing these complex features in terms of combinations of simpler ones.

In this work, we focus on Convolutional Neural Networks (CNN), since this type of networks is very effective for image detection and classification tasks.

3.2.1 Convolutional Neural Network [CNN]

In this kind of deep learning, special neural networks known as convolutional are used. Like all neural networks they have input layers, hidden layers, and output layers, but this kind of networks has two types of hidden layers: convolution layers and pooling layers.

Convolution layers

These hidden layers are concerned with learning local patterns in small two-dimensional windows. They are layers that serve to detect visual characteristics in images such as lines, edges and drops color. These layers allow learn a property of the image in a specific point and are able to recognize it anywhere in one's own image.

Convolution layers are able to learn different elements more complexes according to what was learned in the previous layer. An example would be that a layer learns different types of lines that appear in the image and the next layer is able to learn elements of the image that are composed of different lines learned in the previous layer. The capacity for doing this allows these networks to learn in an efficient way visual concepts that are increasingly more complex.

The convolutional neural networks operate on 3 axes, width, height, and

the channel. In the input, the channel could be 1 if the input image is in a gray-scale or 3 for an RGB color image, one for each color (red, green and blue). Inside the network, the depth is usually higher than 3.

Not all input neurons will be connected to all neurons of this convolution layer. It is done through small localized areas of the space of the input neurons that store the pixels of the image.

The output is the result of applying the Kernel filter on the Original image. The result is the product and sum of the kernel matrix applied to the original image in every slide. The output size change depending on the kernel size. For example, Figure 3.3 shows a filter for detecting horizontal edges. Each output value is the result of the product and sum up of the original image with the filter, in the first slide:

$$3 + 1 + 2 - 1 - 7 - 5 = \textbf{-7} \tag{3.1}$$

The Output has a 4x4 size because in the 6x6 input image we can put the 3x3 filter in 4x4 positions. The matrix begins to travel through all the neurons in the upper left and it moves to the right. When you have finished a row, continue with the row from below.

Applying different filters is possible to detect different features.

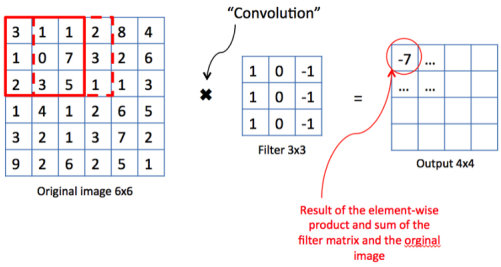

Figure 3.3: Convolution example[3].

[3]https://miro.medium.com/max/1400/1*7S266Kq-UCExS25iX_I_AQ.png

Pooling layers

This type of layers gathers the neurons in groups. This process obtains the most relevant properties of the input data and obviates superfluous data that can be misleading.

There are several forms of pooling. An "average pooling" that obtains the average value of the group of neurons or else "max pooling" that chooses the maximum value of each group.

In this layer, when the neurons are grouped, the entrance is reduced. It is reduced depending on the size of the groupings. For example, if the groups are a 2x2 size, at the output we will obtain a quarter of the size of input data. It can be seen in figure 3.4 that, even if this transformation technique, the spatial relationship is still maintained.

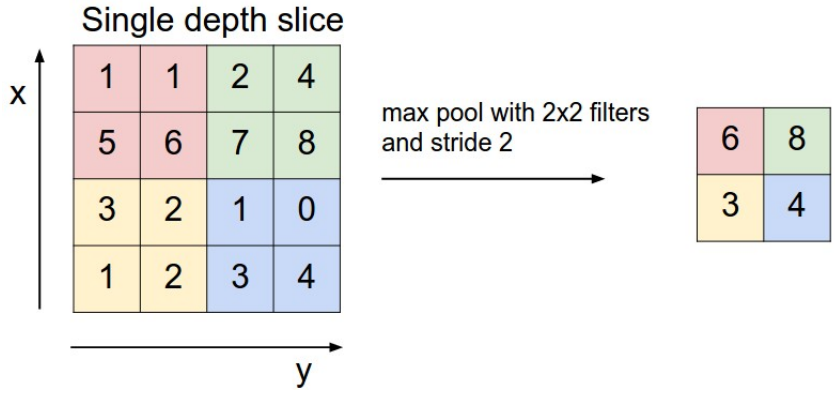

Figure 3.4: Pooling example[4].

[4]http://cs231n.github.io/assets/cnn/maxpool.jpeg

3.2.2 Reducing overfitting: Dropout

Deep learning architectures could have millions of parameters, each layer is connected by weight connections. These models with a lot of parameters can easily overfit to the training data. To help avoid this problem, there are some methods that help to reduce it.

One of them is Dropout, the term refers to dropping out units in a neural network as shown in Figure 3.5. Dropping units is removing neurons from the network, including their weights connections. Dropout uses a fixed probability to be retained and the choice of units to be dropped is random.

Neural networks that use dropout can be trained using an adaptive learning method. The difference is that for each training mini-batch, the forward and backpropagation are done in a thinned network that has to dropout units.

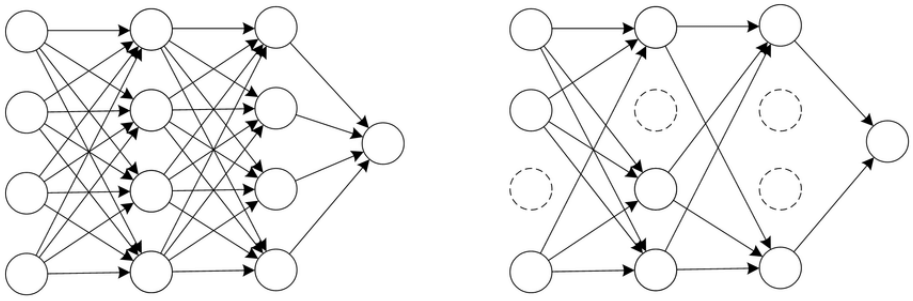

Figure 3.5: Scheme of dropout performance[5].

[5]https://www.researchgate.net/figure/Dropout-neural-network-model-a-is-a-standard-neural-network-b-is-the-same-network_fig3_309206911

3.2.3 Activation functions

In neural networks, the Activation Function of a node defines the output of a node depending on the input or set of inputs. For each neuron, the input is calculated with the activation functions and then the result is sent as an output. Depending on this output, the activation function decides if the connections of this neuron are triggered or not. In this work, two types of activation functions have been used: Rectified Linear Units and Sigmoid.

Rectified Linear Units

In recent years, Rectified Linear Units (ReLU) are the most used activation function because of their simplicity. ReLU has been demonstrated to make training faster compared to other activation functions. The ReLU activation function returns 0 if the output is less than 0 and it returns the same value of the input if the input is more than 0.

Although it may seem by its name, this activation function is not linear, since for all negative values are shifted to 0. The equation that represents it is the following.

$$f(x) = max(0, x) \tag{3.2}$$

And their graphic representation can be seen in Figure 3.6.

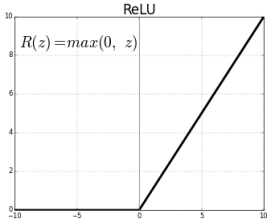

Figure 3.6: ReLU graphic representation[6].

[6]https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec

Sigmoid

The non-linear nature of the sigmoid function is ideal for neural networks. The sigmoid function transforms the entered values to a scale (0,1), where the high values have an asymptotic way to 1 and the very low values tend asymptotically to 0. His equation is the following:

$$Accuracy = \frac{1}{1 - e^{-x}} \tag{3.3}$$

The characteristics are:

- Slow convergence.
- It is not centered on zero.
- It is bounded between 0 and 1.
- Good performance in the last layer.

It is called this way because of his typical "S" shape, as can be seen in Figure 3.7.

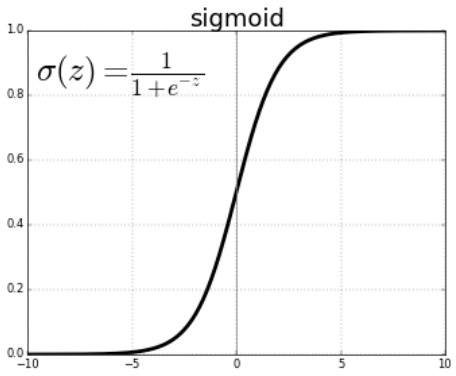

Figure 3.7: Sigmoid graphic representation[7].

[7] https://ml4a.github.io/images/figures/sigmoid.png

3.3 Software tools

For working with automatic learning techniques different tool has been used. The used tools are:

Python: programming language used to implement these techniques [22] . Version 2.7

Numpy: high-level mathematical functions library for operating with big vectors and matrices.

Pip: a tool used to install Python libraries in a more comfortable, faster and efficient way.

Dlib: Library used for facial points extraction of each face. These facial points have been used for working with machine learning techniques.

OpenCV : Free library of artificial vision developed by Intel.

Keras: Library of neural networks written in Python [23]. It has been used to apply deep learning techniques.

Compute Unified Device Architecture (CUDA): Computing Platform including a compiler and a set of development tools created by NVIDIA that allows the user to encode algorithms in NVIDIA GPU. Version 10.

Cudnn: Library that provides high-level implementations for standard routines such as convolution, grouping, normalization and activation layers.

Tensorflow & Tensorflow-gpu: Backend system that allows compiling code of deep neural networks. If we use the GPU extension we allow the computer to be able to process through the graphics processing unit of the graphics card.

Docker : Is a computer program that performs operating-system-level-virtualization [24]. Docker is used to running packages called containers. Containers are isolated from each other and bundle their own application, tools, libraries and configuration files. Containers are created from images that specify their precise contents.

3.4 Datasets

To be able to train the models it is necessary to have datasets with emotions well classified and enough data to be able to optimally train the models. In this section, we present the datasets that we have used during the execution of this work.

3.4.1 Karolinska Directed Emotional Faces [KDEF]

Karolinska Directed Emotional Faces was collected by scientists Ellen Goeleven, Rudi De Raedt, Lemke Leyman and Bruno Verschuere from Ghent University in Belgium [26]. It contains 4900 JPEG images showing 70 people (35 women and 35 men) displaying seven different facial expressions. Each expression is recorded from 5 different angles. All the participants were amateur actors between 20 and 30 years old. For the election of the individuals, mustaches, beards, eyeglasses, earrings, and make-up were exclusion criteria. Some samples can be seen in Figure 3.8.

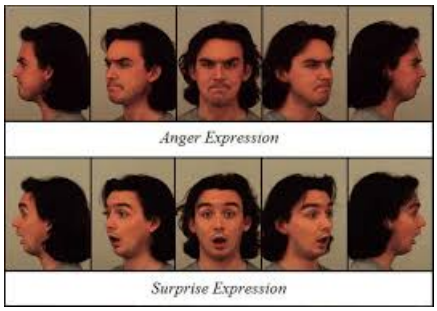

Figure 3.8: KDEF example [26]

3.4.2 Japanese Female Facial Expression [JAFFE]

The Japanese Female Facial Expression (JAFFE) Database that has two hundred and thirteen images of seven emotional expressions performed by ten Japanese female models [25]. The database was built by the Psychology Department at Kyushu University. It is possible to see some examples in Figure 3.9.

Figure 3.9: JAFFE example [25]

3.4.3 FER

The data consists of a 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is centered and occupies about the same amount of space in each image. The database was created using the Google image search API and faces have been automatically registered, each picture is categorized on the emotion shown in the facial expression into one of the seven categories of the Categorical model. Example in Figure 3.10.

Figure 3.10: FER example[8]

3.4.4 Multimodal Emotion Recognition in Polish [MERiP]

Polish emotional database composed of three modalities: facial expressions, body movement and gestures, and speech. The corpora contains recordings registered in studio conditions, acted out by 16 professional actors (8 male and 8 female)[27], Figure 3.11 is an example of a frame of the videos. The data is labeled with six basic emotions categories and the neutral one, according to Ekman's emotion categories.

Figure 3.11: MERiP example [27]

[8] https://mc.ai/face-expression-recognition-with-fastai-v1/

Chapter 4

Data preprocessing and model implementation

4.1 Machine learning on still images

In machine learning, the preprocessing of the data is as important as the algorithms since the input data directly affects the ability of the model to learn. Therefore, it is very important to choose correctly which data we need to train the models.

4.1.1 Face detection

A code provided by Adrian Rosebrock on his website that makes cropping and alignment has been used [28]. To extract faces from images the Dlib face detector is used. It provides the coordinates of a rectangular frame that fits the face. The face has to be detected from the input image and for that purpose, a HOG (histogram of oriented gradients) face detector is used. For every single pixel, the detector looks at the pixels surrounding it with the goal is to figure out changes of intensity between those pixels. Then the detector draws an arrow showing in which direction the image is getting darker and repeating the process for every single pixel in the image, it ends with a gradients image that shows the flow from light to dark across the entire

image.

After this process, it will divide the image into small squares (normally 16x16 pixels each) and count in each square how many gradients points in each major direction and draw an arrow of the strongest direction. The resulting is a much more simple representation that captures the basic structure of a pattern existing in our image. To find faces in an image there is a sliding window over the whole frame, and for each of these windows the HOG is calculated. The resulting HOG is then compared with a database of HOG patterns extracted from training data of faces.

4.1.2 Face points recollection

Using the code provided by Adrian Rosebrock, it is possible to detect 68 face points with their respective X and Y coordinates. See in Figure 4.1 for a sample of these points.

Figure 4.1: Localization in the human face of 68 facial coordinate points [28]

- 1-17 are points of the chin shape.
- 18-22 are points of the left eyebrow.
- 23-27 are points of the right eyebrow.
- 28-31 are points of the nose.
- 32-36 are points on the underside of the nose.
- 37-42 are points of the left eye.
- 43-48 are points of the right eye.
- 49-68 are points of the mouth.

These data have been collected in a CSV file, each line of this file corresponds to the X and Y coordinates of each face of the Dataset. Rows of this file correspond to each face of the Dataset and the columns correspond to the X and Y coordinates of all expression lines distributed in x0, y0, x1, y1, etc. Then, there is another file that indicates to which emotion each face corresponds. This file has the same number of rows as the previous one but it only has one column since each face only has one emotion.

4.2 Machine learning on Kinect motion capture data

MERiP Kinect database is a set of uncut files (not divided into single emotional expressions) recording single actor's five performances of a single emotional state. It is given the information which part of the file corresponds to each single emotional expressions. The first thing to do is to create a program to split these files into smaller ones with single emotional expressions.

As it's possible to see in Figure 4.2. each of these files with a single emotional expression contains the frame times stamp in which the information was obtained and the following positions of the body: Spine Base; Spine Mid; Neck; Head; Shoulder Left; Elbow Left; Wrist Left; Hand Left; Shoulder Right; Elbow Right; Wrist Right; Hand Right; Hip Left; Knee Left; Ankle Left; Foot Left; Hip Right; Knee Right; Ankle Right; Foot Right;

Spine Shoulder; Hand Tip Left; Thumb Left; Hand Tip Right and Thumb Right. For each part of the body, it is given the X, Y and Z position and the X, Y and Z rotation. With this information, it is possible to know the evolution of the different body position during the time.

Each of these single emotion files is labeled with their corresponding emotion classification to be able to apply the machine learning.

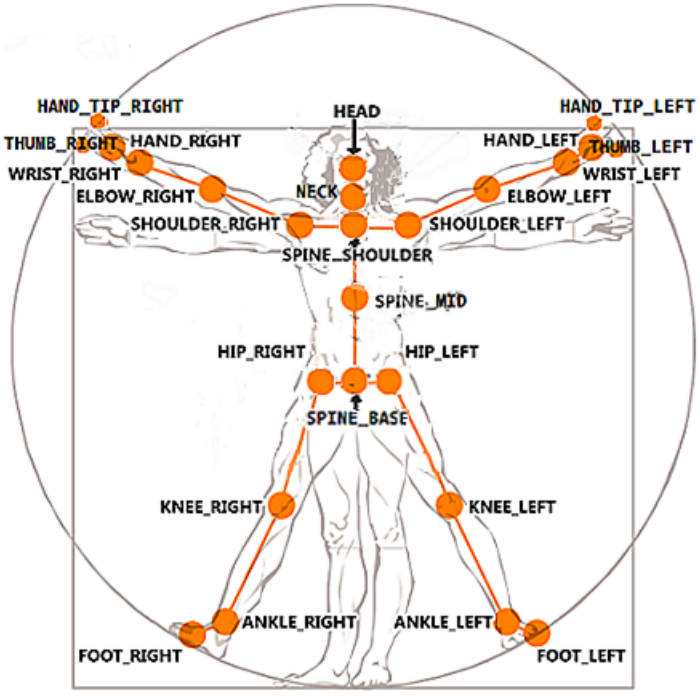

Figure 4.2: Scheme of Kinect skeleton[9]

[9]https://www.semanticscholar.org/paper/Accuracy-evaluation-of-the-Kinect-v2-sensor-during-Capecci-Ceravolo

4.3 Deep learning on still images

To test the models it is necessary to label the images with their corresponding feelings. In machine learning on still images, the data was provided within a csv file. Now, the input data is directly the complete image within a directory with the name of every emotion, the dataset is the same as in machine learning for still images, the Karolinska Directed Emotional Faces (KDEF). To validate it, we have made a partition of the dataset for separating the images that are to train and the images to validate. The structure of the directories is the following.

- /Training

  /Training/afraid
  /Training/angry
  /Training/disgusted
  /Training/happy
  /Training/neutral
  /Training/sad
  /Training/surprised

- /Validation

  /Validation/afraid
  /Validation/angry
  /Validation/disgusted
  /Validation/happy
  /Validation/neutral
  /Validation/sad
  /Validation/surprised

4.4 Deep learning on audio recognition

Sound is transmitted as waves. Sound waves are one-dimensional. At every moment in time, they have a single value based on the height of the wave. It is possible to feed these numbers right into a neural network. However, trying to recognize speech patterns by processing these samples directly is difficult. Instead, it is possible to make the problem easier by doing some pre-processing of the audio data.

The chosen Dataset has been the MERiP Dataset, which contains 315 different sound-files performing the 7 different categorical emotions: afraid, angry, disgusted, happy, neutral, sad, surprised. The 315 sound-files are not directly given in the dataset, it contains long videos of people performing the emotions several times and also contains descriptors that relate in which sequences of the videos the different emotions are represented. So first of all the videos have to be split and converted to audio wav files, these actions have been programmed in python.

Wav files are just a vector with amplitudes in each time interval. The procedure of representing a word in a vector or matrix form is called embedding. One dimensional vector is easy to visualize, however, speech recognition rarely works with one dimension amplitude data. In our case we would follow spectrogram method (to be more precise log-spectrograms as these are better to visualize)[30].

The spectrogram consists of taking a certain number of samples by means of a temporal window, with a specific size, depending on the type of analysis that is made of the signal, harmonic or resonant, the window must have a certain size. Then the calculation of the frequency content of the samples put in a window is made, and they are represented in a graph in three dimensions.

The three-dimensional graph can be represented in different ways, but the usual way to find it is to represent the time on the abscissa axis, representing

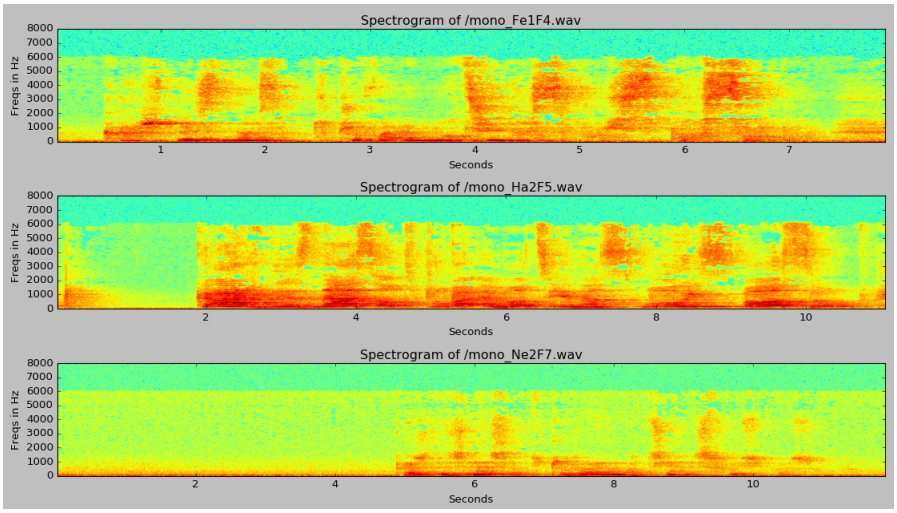

Figure 4.3: Log spectogram of some used audio files

the frequencies on the ordinate axis and a representation of the energy in dB in the three-dimensional plane. This is represented with a range of colors that indicate the variation in energy. An example can be seen in Figure 4.3.

4.5 Deep learning on video recognition

This section discusses the steps performed for pre-processing on video. The model does not accept videos as input; the videos must be converted by extracting frames and using them as input to the models.

The deep learning is done over the MERiP emotion video files, but this database is not ready to train it directly since it contains uncut video file (not divided into single emotional expressions) recording of single actor's five performances of a single emotional state.

Before training this database it is needed to create a program able to split the video files in smaller ones with every single emotional state. As the deep learning architecture works with each frame of the video, the video files have not been split into smaller video files, for each emotion, a folder

has been created and each of these folders contains the frames of each single emotion performance.

As a result, the input that has the following structure, there are to folders one for training and the other for validation, and inside each folder there one folder for every emotion, same as in deep learning on still images. Then inside the emotion folder there is a folder for each video, and inside this folder there are all the frames corresponding to the video.

4.6 Deep learning's model

As already explained, deep learning works with neural networks, where each network has a number of layers.

Images of all sizes enter into the classifier, but a readjustment is made to a size of 128x128. These images are composed of 3 channels for each color (RGB) so the size of the matrix representing each image will be 128x128x3.

If the number of neurons in a layer is very low, the model has no ability to detect patterns, therefore, in each layer there must be a sufficient number of neurons to detect these patterns. However, if we increase the number of neurons too much, the accuracy no longer increases and we only increase the calculation time unnecessarily. Therefore, to choose the number of neurons in the model we have taken as a starting point the examples shown in the Keras documentation [31].

The model starts with a hidden layer of 32 neurons, then the next layer is 64 neurons, this has been done to give depth to the model and be able to identify more complex features, then there is a third hidden layer of 128 neurons, this has been made with the same purpose, all these layers use the ReLU activation function, this choice has been made to make the neuronal network faster.

The model has at the end two more layers: a hidden layer of 64 neurons with ReLU activation (more neurons in this layer mean more complex neural network) and in the end, it has an output layer with equal neurons as classes we have, in this case, 7 emotions classification need 7 neurons. For this layer, the activation layer is the Sigmoid, it is slower than Relu but is more appropriate for the output layer of the neural network, as already explained above.

Figure 4.4 shows a schematic of the neural network.

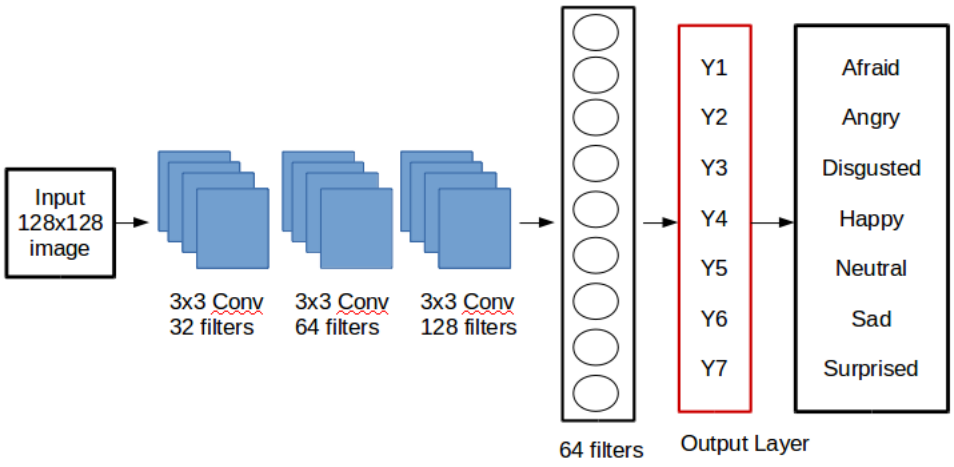

Figure 4.4: Schematic of the neural network

Chapter 5

Experimental results

In this chapter, the results of the different experiments will be presented. However, before it is important to note that practically all of the images are people acting, only the FER database is done with spontaneous expressions. Recognizing spontaneous expressions is more complex. Therefore, the results of the accuracy for a real application of these learning architectures would undoubtedly be worse.

5.1 Metrics for evaluation

In order to evaluate if the learning algorithms are really efficient a different kind of metrics is used. For example, the confusion matrices and the classification accuracy of our system are used. These two methods have been used since they are the ones most used by other researchers.

Accuracy: Parameter that provides how successful is a model when predicting.

$$Accuracy = \frac{Correctly \quad detected \quad emotions}{Total \quad emotions} * 100 \tag{5.1}$$

Confusion matrix: Tool that allows seeing how many predictions have been made, which has been successful and which has wrong. The values of the matrices will be shown as a percentage.

Losses: Parameter that allows observing which of the samples has been lost because of wrong predictions. It is the degree of error between measure calculated at the output and the desired response to the output.

5.2 Machine Learning on still images

These systems were trained using the KDEF database that contains static images, with the same amount for each emotion and subject. Input data is the same for each classifier, it contains the X and Y coordinates of each face of the Database. Another file is used in which for every face has indicated what emotion corresponds.

As explained before, in this work four different algorithms for machine learning are studied: Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF) and Multilayer Perceptron (MLP). The purpose of this work is comparing them in their accuracy for recognizing emotions in facial images. We also want to study how they are affected by the fact that the database has very similar images (only frontal images) or substantially different images (images in 5 different angles: full left profile, half left profile, frontal, half right profile, full right profile).

Frontal KDEF database contains 980 images, 784 images were used for training and 196 were used for validation, so for each emotion, there were 112 pictures for training and 28 for validation. The other database which contains images from five different angles contains 4900 images, 3920 used for training and 980 for validation, each emotion is trained with 560 images and validated with 140. A summary is shown in table 5.1.

Once the classifiers have been chosen and the way to train them too, we

Table 5.1: KDEF dataset for machine learning

	Total	**Train**	**Emotion**	**Test**	**Emotion**
Frontal	980	784	112	196	28
Diff. angles	4900	3920	560	980	140

evaluate them. Table 5.2 shows the accuracy that is has been obtained with different classifiers and with only frontal images or with different degrees.

Table 5.2: Accuracy summary for machine learning on still images.

	SVM	**DT**	**RF**	**MLP**
Frontal	81.22	64.22	66.12	79.59
Different angles	77.03	55.5	54.39	74.69

It is clearly observed that there is a substantial difference between the algorithms in terms of their effectiveness. There are two algorithms that are around 80% accuracy, Support Vector Machines, and Multilayer Perceptron, while Decision Tree and Random Forest are around 65% accuracy.

These results are due to the fact that SVM and MLP are more complex algorithms capable of taking into account the complexity of the characteristics. In contrast, the performance of DT and RF is based on a sequence of simple decisions that are not optimal for the recognition of something as complex as emotions.

It is also evident that the fact of introducing images in different angles makes it more difficult to recognize emotions and therefore this causes the accuracy to decrease in all the algorithms, but it is also clear that it does not affect them in the same way. While the accuracy decreases 5.16% and 6.16% for SVM and MLP respectively, for RF and DT it decreases by 13.58% and 17.74%.

These algorithms also have things in common for all of them. The emo-

tion that they recognize best is happiness and the ones they recognize the least is sadness, in general, this is because they confuse it with neutral emotion or anger. All this can be verified by looking at the tables 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9 and 5.10 that show the confusion matrices for the 4 algorithms with the two different databases.

Table 5.3: Accuracy for SVM on frontal images.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	75.71	4.29	0.00	2.86	1.43	7.14	8.57
angry	8.57	82.86	0.00	0.00	2.86	5.71	0.00
disgusted	5.71	10.00	77.14	0.00	0.00	7.14	0.00
happy	2.86	0.00	2.86	94.29	0.00	0.00	0.00
neutral	1.43	2.86	0.00	1.43	82.86	10.00	1.43
sad	7.14	4.29	7.14	0.00	11.43	67.14	2.86
surprised	7.14	0.00	0.00	0.00	2.86	1.43	88.57

Table 5.4: Accuracy for RF on frontal images.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	41.43	5.71	2.86	4.29	17.14	1.43	27.14
angry	1.43	84.29	8.57	0.00	0.00	1.43	4.29
disgusted	2.86	11.43	70.00	4.29	10.00	0.00	1.43
happy	1.43	0.00	1.43	95.71	0.00	0.00	1.43
neutral	2.86	11.43	1.43	0.00	82.86	1.43	0.00
sad	7.14	10.00	1.43	11.43	55.71	5.71	2.86
surprised	10.00	0.00	0.00	0.00	7.14	0.00	82.86

Table 5.5: Accuracy for DT on frontal images.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	55.71	1.43	2.86	1.43	11.43	8.57	18.57
angry	4.29	67.14	14.29	2.86	4.29	5.71	1.43
disgusted	1.43	10.00	62.86	11.43	4.29	10.00	0.00
happy	4.29	0.00	8.57	82.86	1.43	2.86	0.00
neutral	10.00	2.86	0.00	1.43	57.14	28.57	1.43
sad	4.29	7.14	11.43	5.71	21.43	47.14	2.86
surprised	14.29	0.00	0.00	1.43	2.86	4.29	77.14

Table 5.6: Accuracy for MLP on frontal images.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	72.86	5.71	1.43	1.43	1.43	5.71	11.43
angry	5.71	77.14	4.29	0.00	4.29	7.14	1.43
disgusted	4.29	8.57	81.43	0.00	0.00	5.71	0.00
happy	4.29	1.43	1.43	92.86	0.00	0.00	0.00
neutral	1.43	4.29	0.00	1.43	80.00	11.43	1.43
sad	1.43	7.14	4.29	0.00	14.29	71.43	1.43
surprised	14.29	0.00	0.00	0.00	4.29	0.00	81.43

Table 5.7: Accuracy for SVM with different angles.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	66.67	2.38	2.98	2.38	4.17	7.14	14.29
angry	3.57	76.19	9.52	0.60	4.76	4.76	0.60
disgusted	3.01	9.04	81.93	0.00	2.41	3.61	0.00
happy	4.85	0.61	0.61	89.70	1.21	1.82	1.21
neutral	1.80	4.79	2.40	0.60	77.84	11.38	1.20
sad	8.77	6.43	5.26	1.17	17.54	60.82	0.00
surprised	11.52	0.00	0.00	0.00	1.82	0.61	86.06

Table 5.8: Accuracy for RF with different angles.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	11.90	8.33	2.98	10.12	33.93	1.19	31.55
angry	0.00	77.98	13.10	0.00	4.76	0.00	4.17
disgusted	1.81	23.49	48.19	12.65	12.05	1.20	0.60
happy	0.61	0.61	6.67	81.82	5.45	0.00	4.85
neutral	0.00	10.78	3.59	5.39	76.05	2.40	1.80
sad	0.58	12.28	11.11	11.11	56.14	2.34	6.43
surprised	4.24	1.21	0.00	1.21	10.91	0.00	82.42

Table 5.9: Accuracy for DT with different angles.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	38.69	5.95	4.76	6.55	6.55	15.48	22.02
angry	5.36	67.26	8.93	2.98	5.36	10.12	0.00
disgusted	5.42	14.46	50.00	8.43	5.42	16.27	0.00
happy	5.45	2.42	9.09	74.55	2.42	4.24	1.82
neutral	11.98	0.60	3.59	0.60	53.29	25.75	4.19
sad	14.62	5.85	10.53	4.09	24.56	38.01	2.34
surprised	22.42	1.21	0.61	4.24	2.42	2.42	66.67

Table 5.10: Accuracy for MLP with different angles.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	62.50	4.17	3.57	2.38	5.36	7.74	14.29
angry	4.17	74.40	8.33	0.60	5.95	5.95	0.60
disgusted	2.41	10.84	76.51	0.60	3.01	6.02	0.60
happy	2.42	0.61	2.42	87.27	1.82	1.21	4.26
neutral	1.80	4.09	2.40	0.00	78.44	11.98	0.60
sad	9.36	4.09	8.77	1.75	12.28	63.74	0.00
surprised	15.76	0.61	0.00	0.61	1.21	1.82	80.00

5.3 Machine Learning MERiP kinetic

To evaluate the effectiveness of machine learning algorithms with dynamic information, the MERIP database has been used. Each element of the database contains the position of one of the "joint positions" that the Kinect sensor returns in addition to the moment of time in which they have been taken, so its movement during the time can be studied.

The dataset contains 315 emotion performances, for each performance Kinect captures "join points" constantly. In total there are 37355 frames, 80% was used for training, 29884 frames, and the other 20% for validation, 7471. In table 5.11 is shown.

Table 5.11: Database for machine learning on Kinetic.

	Videos	**Frames**	**Train**	**Test**
Database	315	37355	29884	7471

As can be seen in Table 5.12, the effectiveness of the different algorithms is substantially different from the still images. This is because the input information is totally different. In the first database, the input were points of the face of people who were static. In this case, the entrance are joint positions of the body of moving people.

Table 5.12: Accuracy summary for machine learning on Kinetic position.

	SVM	**DT**	**RF**	**MLP**
Accuracy	72.83	70.8	46.17	80.57

In this case, the highest accuracy, 80%, is achieved with the Multilayer Perceptron. Then, Support Vector Machines and Decision Tree are around 70%. Random Forest gives the worst accuracy with only 46%, very far from the others.

As can be seen in the tables 5.13, 5.14, 5.15, 5.16 the highest accuracy is given for the neutral emotion, probably because it is the least expressive and therefore in which there is less movement.

For the rest of the emotions, the accuracy percentage is quite stable, there is not much variability. This may be due to the fact that the input is points of the space more separated from each other, for example in the previous case, where many points of the face were extracted, it was easy to detect a smile and for that reason, the emotion of happiness was easier to detect.

For this case, the position is not as important as movement, and as all emotions involve movement, the results for all of them are similar.

Table 5.13: Accuracy for SVM on Kinetic positions.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	73.43	2.86	10.87	4.53	0.05	3.86	4.39
angry	3.10	68.46	4.69	12.54	0.23	1.40	9.58
disgusted	9.58	5.22	76.23	2.31	0.16	1.40	5.10
happy	7.37	11.65	4.26	67.18	0.45	1.19	7.90
neutral	0.52	0.64	0.07	0.33	95.79	2.50	0.14
sad	8.31	5.14	4.71	3.83	3.83	70.46	4.45
surprised	5.94	11.87	5.72	11.63	2.68	3.88	58.28

Table 5.14: Accuracy for DT on Kinetic positions.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	29.52	0.04	48.64	4.64	3.55	8.41	5.20
angry	7.50	2.35	36.02	26.65	3.65	6.25	17.58
disgusted	1.36	0.00	89.57	1.10	3.52	3.32	1.13
happy	8.47	0.19	23.45	53.98	5.25	3.31	5.36
neutral	0.00	0.00	3.09	0.00	91.87	4.18	0.86
sad	9.68	0.00	38.40	0.00	12.67	35.56	3.70
surprised	20.58	0.13	31.39	8.57	11.34	7.67	20.32

Table 5.15: Accuracy for RF on Kinetic positions.

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	63.56	7.81	15.37	5.15	0.60	3.39	4.13
angry	3.38	64.79	5.23	16.67	0.69	2.15	7.10
disgusted	11.25	9.75	68.16	3.81	0.16	4.29	2.58
happy	2.86	11.48	6.17	67.14	1.84	4.11	6.40
neutral	0.78	0.50	0.00	1.69	93.42	2.66	0.95
sad	6.67	4.05	4.39	4.76	3.29	71.90	4.93
surprised	7.84	7.13	5.36	6.85	2.99	3.19	66.64

Table 5.16: Accuracy for MLP on Kinetics positions

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	75.32	2.17	12.76	4.26	0.00	1.41	4.08
angry	2.02	80.23	2.85	11.44	0.15	0.63	2.69
disgusted	10.58	2.82	78.48	4.63	0.80	0.46	2.22
happy	2.14	11.29	5.81	75.89	0.21	0.34	4.32
neutral	0.14	0.17	0.00	0.38	98.93	0.17	0.21
sad	3.81	6.20	3.40	4.19	2.17	80.47	4.30
surprised	2.57	6.20	5.10	6.03	1.68	3.74	74.68

5.4 Deep Learning for still images recognition

In deep learning for still images recognition the same dataset as in machine learning for still images has been used, KDEF. The main difference is that with machine learning the input is the face points derived from the image, and in this section, the input is directly the complete image.

5.4.1 KDEF

As it was expected, the best accuracy, 87.14% (the complete results can be seen in figure 5.18, is obtained with deep learning on still images using only one database, because the images are more homogeneous. To train this model the KDEF dataset have been used, just the same as used with machine learning for still images. The summary is shown in Table 5.17. This has been done with the aim of subsequently being able to compare the results.

Table 5.17: KDEF database for still images

	Total	**Train**	**Emotion**	**Test**	**Emotion**
Frontal	980	770	110	210	30
Diff. angles	4900	3850	550	1050	150

To study the effect of changing the number of epochs and the size of the batch, we have repeatedly trained this database by changing these values. It has been observed that the size of the batch does not significantly affect the accuracy, it only modifies the training time. As for the epochs, they do not affect the accuracy either as long as there are enough epochs to learn, from 40 epochs the results are maintained, the results are shown in figure 5.18 correspond to 75 epochs.

5.4.2 JAFFE & KDEF

To study the effect caused by the combination of several databases for the training of deep learning models, it has been decided to expand the initial database with the JAFFE database, the new size could be seen in table 5.19. These two databases are quite similar, with the particularity that JAFFE

Table 5.18: Accuracy for deep learning for still images recognition KDEF

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	93.33	0.00	0.00	0.00	3.33	0.00	3.33
angry	6.67	80.00	3.33	0.00	0.00	10.00	0.00
disgusted	3.37	6.67	83.33	0.00	0.00	6.67	0.00
happy	3.33	0.00	0.00	96.67	0.00	0.00	0.00
neutral	0.00	0.00	0.00	0.00	86.67	13.33	0.00
sad	6.67	3.33	0.00	0.00	3.33	86.67	0.00
surprised	16.66	0.00	0.00	0.00	0.00	0.00	83.33

contains gray-scale images of Japanese women, but the positions and angles in which the photographs have been taken are similar to KDEF.

Table 5.19: KDEF & JAFFE database for still images

	Total	**Train**	**Emotion**	**Test**	**Emotion**
Frontal	1225	945	135	280	40

After training the model, it is observed that the accuracy decreases a bit but not excessively, the resulting accuracy is 79.96%. Figure 5.20 shows the confusion matrix for 75 epochs, this number of epochs has been chosen to later be able to compare the different datasets.

Table 5.20: Accuracy for deep learning for still images recognition KDEF & JAFFE

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	82.50	0.00	5.00	2.50	2.50	0.00	10.00
angry	0.00	72.50	12.50	0.00	0.00	7.50	0.00
disgusted	5.00	7.50	85.00	0.00	0.00	2.50	0.00
happy	5.13	0.00	0.00	89.74	5.13	0.00	0.00
neutral	10.0	2.50	2.50	0.00	80.00	0.00	0.00
sad	12.50	2.50	7.50	2.50	12.50	62.50.00	0.00
surprised	12.50	0.00	0.00	0.00	0.00	0.00	87.50

5.4.3 KDEF & FER

Another dataset has been created from two datasets with totally different characteristics, FER dataset has a big size, so the database is expanded widely, as could be seen in Table 5.21. FER dataset is done with gray-scale spontaneous expressions labeled in the categorical model. Unlike the previous datasets, for the same emotion, the images differ greatly from one another. This makes it much harder to train and that the accuracy drop significantly to a 48.53%, the confusion matrix can be seen in Table 5.22.

Table 5.21: KDEF & FER database for still images

	Total	**Train**	**Emotion**	**Test**	**Emotion**
Frontal	5770	4360	623	1410	200

These results are more similar to those that would be obtained in a real application because the images are spontaneous and have not been acted by professional actors.

Table 5.22: Accuracy for deep learning for still images recognition KDEF & FER

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	32.42	17.03	0.00	11.54	12.64	12.64	13.74
angry	8.67	47.98	2.31	6.94	17.34	15.03	1.73
disgusted	13.41	28.46	22.36	9.76	9.76	13.82	2.44
happy	2.76	6.30	0.00	72.05	9.84	6.69	2.36
neutral	7.54	9.55	0.50	8.04	57.79	13.57	3.02
sad	13.43	14.93	0.50	11.94	15.92	41.29	1.99
surprised	11.61	6.45	0.00	5.81	7.74	2.58	65.81

5.5 Deep Learning for audio recognition

In this section, we report the recognition accuracy of using the classifier on MERiP database which contains 315 audio files. To evaluate the classification error 7-cross-validation-test were used. As shown in Table 5.23, the accuracy obtained for audio emotion recognition are significantly worse than for image and video.

The accuracy has been 32.17 %, if the classification was totally random, the expected result for 7 emotions would be 14.3 %, (1/7). Consequently, it can be said that the classifier works even if it does not have very good results. It should also be noted that the database contains audio files where the phrase reproduced is exactly the same for the 7 emotions, that makes the differentiation very complicated since only the intensity changes.

Table 5.23: Accuracy for deep learning for audio recognition

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	33.33	0.00	8.33	16.67	33.33	8.33	0.00
angry	6.67	13.33	0.00	53.33	20.00	6.67	0.00
disgusted	26.67	0.00	6.67	6.67	53.33	0.00	6.67
happy	12.50	6.25	6.25	68.75	6.25	0.00	0.00
neutral	10.00	0.00	0.00	0.00	80.00	10.00	0.00
sad	23.08	0.00	7.69	0.00	46.15	23.08	0.00
surprised	21.43	0.00	0.00	0.00	64.29	14.29	0.00

5.6 Deep Learning for video recognition

The database used for video recognition contains 315 videos, the same ones as the audio recognition. These 315 videos are represented in 2788 frames, 2228 used for training and 560 for validation, as it is shown in Table 5.24.

Table 5.24: Database for video recognition.

	Videos	**Frames**	**Train**	**Test**
Database	315	2788	2228	560

The deep learning model for the recognition of emotions on video has obtained an accuracy of 81.00%. The confusion matrix is shown in table 5.25. The model is more capable of classifying neutral, disgusted and surprised emotions while it has less accuracy predicting the other emotions. Especially angry emotion with only 59.41%.

It is important to note that for this training the MERiP database has been used, the same as in the deep learning audio recognition. This has been done with the purpose of combining audio and video recognition to create a multimodal one.

Table 5.25: Accuracy for deep learning for video recognition

	afraid	angry	disg.	happy	neutr.	sad	surpr.
afraid	79.78	2.25	6.74	1.12	1.12	0.00	8.99
angry	21.78	59.41	15.84	0.00	1.98	0.99	0.00
disgusted	5.26	1.05	91.58	1.05	0.00	0.00	1.05
happy	3.70	13.58	6.17	71.60	0.00	3.70	1.23
neutral	0.00	0.00	0.00	0.00	96.25	1.25	2.50
sad	3.51	0.00	0.00	0.00	14.04	78.95	3.51
surprised	7.02	0.00	0.00	0.00	3.51	0.00	89.47

5.7 Discussion

5.7.1 Machine Learning vs Deep Learning

Comparing the results, figure 5.26, obtained with the deep learning model with those obtained with machine learning it is easy to see that deep learning works significantly better than machine learning, especially when compared to Random Forest and Decision Tree.

It is important to note that exactly the same data set has been used for all the experiments. The difference is that in machine learning 68 relevant points have been extracted from the face of the actors, while in deep learning the complete images have been used. It should also be noted that SVM and MLP achieve quite good results with a much lower calculation time than the deep learning model.

Table 5.26: Accuracy summary for machine learning and deep learning

	SVM	**DT**	**RF**	**MLP**	**Deep Learning**
Accuracy	81.22	64.22	66.12	79.59	87.14

5.7.2 Acted vs Spontaneous database

The difference between using a database with actors or another with spontaneous images is huge, the same deep learning model has been trained with three different datasets, KDEF dataset and the dataset created combining KDEF and JAFFE contains only actors performing emotions, instead, FER dataset contains labeled spontaneous expressions. While in databases where actors have overacted the emotions are consigned some accuracy of 87.14% and 79.96% respectively, the database constructed with random images of the internet does not reach 50% accuracy.

It should also be noted that the images of FER are really difficult to classify even for a human since they have been labeled by an algorithm. For the same emotion label, there are really different facial expressions as shown previously in the Dataset section.

Table 5.27: Accuracy summary for acted and spontaneous databases

	KDEF	**KDEF&JAFFE**	**KDEF&FER**
Accuracy	87.14	79.96	48.53

Chapter 6

Conclusion

In this thesis, machine learning and deep learning models are evaluated, including SVM, RF, DT, MLP, and CNNs. The evaluation clearly demonstrates the superiority of using deep learning and concretely convolutional neural networks for image classification.

The basic model was further tested with audio and video recognition, getting good results with the video but worse than expected with the audio.

A process of study of the bibliography related with emotional recognition has been carried out, which proposes models that constitute the state of the art in all the problems treated, and some of these models have been implemented with the aim of being used in future works for tasks of recognition of emotions by computer vision.

From a technical point of view, this work has served to improve in the management of the combination libraries of Keras-Tensorflow. These valuable tools, open-source and with great growth in reliability and potential, have been used for the implementation of the network architectures exposed throughout the investigation.

6.1 Future work

One of the most obvious continuations of this work is the creation of a multimodal model that combines the video and audio inputs, and therefore significantly improves the results.

The next step would be to save the parameters of the model to be able to use it in an application in real-time. For example, this model could be added to a humanoid robot for being able to detect the feelings of the people with whom the robot interacts and improve the performance as a service robot, getting to empathize with humans and trying to make them feel better.

As explained in the discussion section, surely if we applied the models created in a real application, its effectiveness would be deficient, since the models have been created with acted databases and the categories of emotions are very rigid. Probably for a future work the model needs to be tested against different types of data, changing the background and light conditions too.

Also, it could be investigated if applying another emotion classification, such as dimensional, which takes into account the intensity of the emotions, could be more effective for recognizing the most intense one and not erring in detecting the most subtle ones.

Bibliography

[1] H. Ruthrof. *The body in language* Bloomsbury Publishing, 2015.

[2] B. Pease, A. Pease. *The Definitive Book of Body Language: how to read others' attitudes by their gestures* Hachette UK, 2016.

[3] B. Pease, A. Pease. *The definitive book of body language* Bantam, 2004.

[4] P. Ekman. *Universal and cultural differences in facial expression of emotion.* Nebr. Sym. Motiv.19 (1971) 207–283.

[5] R. Plutchik. *The nature of emotions human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice.* American scientist 89 (4) (2001)344–350

[6] P. F. Felzenszwalb, D. McAllester. *Object detection grammars.* CCV Workshops, 2011, p. 691.

[7] M. A. Fischler, R. A. Elschlager. *The representation and matching of pictorial structures* IEEE Transactions on computers 100 (1)(1973) 67–92.

[8] K. Kotsia and I. Pitas *Facial expression recognition in image sequences using geometric deformation features and support vector machines.* Image Processing, IEEE Transactions on, vol. 16, no. 1, pp. 172–187, Jan 2007

[9] K. Anderson and P. W. Mcowan. *A real-time automated system for recognition of human facial expressions* EEE Trans. Syst., Man, Cybern. B, Cybern, pp. 96–105, 2006

[10] B. V. Kumar. *Face expression recognition and analysis: the state of the art* Course Paper, Visual Interfaces to Computer, 200

[11] Z. Yu and C. Zhang. *Image based static facial expression recognition with multiple deep network learning* ICMI Proceeding

[12] S.-Y. D. Bo-Kyeong Kim, Jihyeon Roh and S.-Y. Lee. *Hierarchical committee of deep convolutional neural networks for robust facial expression recognition* Journal on Multimodal User Interfaces, pages 1–17, 2015.

[13] N Srivastava, GE Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting.* Journal of Machine Learning Research 15 (1), 1929-1958.

[14] R. W. Picard, E. Vyzas, J. Healey. *Toward machine emotional intelligence: Analysis of affective physiological state.* TPAMI 23 (10) (2001) 1175–1191.

[15] B. Reeves, C. Nass. *How people treat computers, television, and new media like real people and places.* CSLI Publications and Cambridge university press Cambridge, UK, 1996.

[16] H. Ranganathan, S. Chakraborty, and S. Panchanathan. *Multimodal Emotion Recognition using Deep Learning Architectures.* 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016

[17] N. Ronghe, S. Nakashe, A. Pawar, and S. Bobde. *Emotion recognition and reaction pre- diction in videos* 2017

[18] Evgeniou, Theodoros & Pontil, Massimiliano. *Support Vector Machines: Theory and Applications.* In 2049. 249-257. 10.1007/3-540-44673-7_12 (2001)

[19] Ronny Kohavi, Ronny Kohavi. *Data mining tasks and methods: Classification: decision-tree discovery* Handbook of data mining and knowledge discovery Pages 267-276. Oxford University Press, Inc. New York, NY, USA ©2002

[20] M Denil, D Matheson, N De Freitas. *Narrowing the Gap: Random ForestsIn Theory and In Practice* in Proceedings of The 31st International Conference on Machine Learning, pp. 665–673.(2014)

[21] Hassan Ramchoun, Mohammed Amine Janati Idrissi, Youssef Ghanou, Mohamed Ettaouil. *Multilayer Perceptron: Architecture, Optimization and Training.* International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº1.

[22] Python. February, 2019.
`https://www.python.org/`

[23] Keras. February, 2019.
`https://keras.io/`

[24] Docker. April, 2019.
`https://www.docker.com/`

[25] Michael J Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. *The Japanese female facial expression (JAFFE) database.* In Proceedings of third international conference on automatic face and gesture recognition, pages 14–16, 1998.

[26] Manuel G Calvo and Daniel Lundqvist. *Facial expressions of emotion (kdef): Identification under different display-duration conditions.* Behavior research methods, 40(1):109–115, 2008.

[27] Tomasz Sapiński, Dorota Kamińska, Adam Pelikant, Cagri Ozcinar, Egils Avots, Gholamreza Anbarjafari. *Multimodal Database of Emotional Speech, Video and Gestures.* Part of the Lecture Notes in Computer Science book series (LNCS, volume 11188).

[28] Adrian Rosebrock. *Face alignment with opencv and python, 2017.*
`https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/`

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* Journal of Machine Learning Research, 15:1929–1958, 2014.

[30] Honglak Lee, Peter Pham, Yan Largman, Andrew Y. Ng. *Unsupervised feature learning for audio classification using convolutional deep belief networks* Part of: Advances in Neural Information Processing Systems 22 (NIPS 2009)

[31] Keras Documentation. March, 2019.
`https://keras.io/examples/`