Emotion Recognition based on Facial Expressions

Maria Elena Lechuga Redondo

A thesis submitted for the degree of Master of Science in Artificial Intelligence

Supervisor: Dra. Alba García
School of Computer Science and Electronic Engineering
University of Essex

August 2018

Abstract

Detecting expressions in people's faces is highly demanded in computer vision fields, due to its several useful applications. Notorious psychologists differentiated between two types of facial expressions: macro-expressions which can be easily perceived by a human eye, and micro-expressions which on the contrary, only an expert eye could detect given their limited duration. This project, aims to develop a system capable of detecting seven possible macro-expressions (Anger, Happiness, Sadness, Fear, Surprise, Neutral and Disgust) just by looking at people's faces in real time streams. In order to accomplish that, several state-of-the-art techniques will be studied and analysed, along with different preprocessing techniques to cope with images or videos obtained in uncontrolled situations.

Key words

Machine Learning, Facial Emotion Recognition, Macro-expression, Micro-expression, Descriptor, Classifier, Computer Vision.

Table of Contents

1. Introduction & Motivation

Developing social relationships is fundamental in order to be integrated in society, it is a fact that happens from children in schools to businessmen in companies. In addition, it is not only necessary to be capable of talking with others, but also to understand their needs, feelings, thoughts or intentions, as the key to establishing good communications resides on paying attention. Furthermore, several investigations have proven that people do not only transmit information by speaking or writing, but also through their facial expressions and body movements [1, 2].

Inquiring the facial expressions field, Paul Ekman, American psychologist and professor emeritus at the University of California, published in 1978 one of the first papers announcing that emotions are strictly related to facial expressions. He conducted studies in order to answer questions such as, can this expressions be detected or disassembled? How many expressions are there? Do they vary with culture, language, age, sex or personality? [3] As a result, it was found that the total number of facial expressions was not clear, even though, they were common in humans and not dependent on the nationality, age or sex. Further experiments performed along the years have backed up this theory, differentiating between two types of facial expressions: macro-expressions and micro-expression. On the one hand, macro-expressions usually occur over multiple regions of the face and are easily observed by a non-trained human eye. On the other hand, micro-expressions could vary between 1/3 to 1/25 seconds and on the contrary, arise over small facial areas and are very difficult to identify [4]. In addition, it has been defined that every person innately owns six common macro-expressions: happiness, sadness, surprise, fear, anger and disgust [3,4] plus the neutral expression, however the number of micro-expressions is unknown [3].

It could be appreciated that having a system capable of identifying people's emotions by just looking at their faces could cause a big impact in society. The first electronic devices were not smart, the only input they could receive was through pushing a button, as the system inside them was very simplistic. Over the years, all of them have changed by creating complex operative systems constituted by applications with user interfaces, allowing the user to interact with the device by using external peripherals like, a keyboard or a touchpad screen. Currently, there are some devices that allow the user to control them with their own voice. The goal of all of these improvements is to simplify the human-computer interaction to make their experience the most comfortable as if they were communicating with humans. One good example is Alexa [40], a virtual assistant developed by Amazon which helps the user interact with their home devices by just talking to "her". Currently, Alexa just receives input as voice format, but why stop there? Would it not be better to allow Alexa also to receive input as video format (giving her "eyes")? Maybe that way, Alexa would know when the user wants to turn on or off the lights, the TV or even call their mother without the intervention of words.

On top of that, new applications could arise that could intervene in several professional sectors of society. From schools, in order to detect the level of interest or concentration of the students in a classroom, to police stations, judicial proceedings or psychiatrists test to deceived

the veracity of the subject that is being evaluated, and medical scenarios, where doctors could detect suppressed feelings in patients to recognise when additional reassurance is needed [5]. Private sectors could also benefit, from business situations to workplace environments, couple relationships or even at home, when detecting glimpses of truth, happiness, perspicacity, or non-conformity becomes crucial. Another use could be in robotics, as humanoid robots could develop emotional intelligence and therefore build social relationships with humans, allowing them a better integration [2].

Hence, this project aims to develop a system capable of classifying the emotion that the subject is feeling into one of the seven macro-expressions defined by Paul Ekman, either receiving an image or a real time video of the subject. What is more, as the system tries to imitate a human in the most similar way, the user won't need to accomplish any requirement in order to be evaluated, ergo no specific distance or position from the camera are required.

Overview

This document is basically divided in five sections: *Related Work*, *Proposed Approach*, *Experiments*, *Overview of the Results* and *Conclusion and Future Work*.
Related Work section will set out the background for facial emotion recognition, identifying the possible approaches along with some real examples. Next, Proposed Approach will detail the structure of the model besides the database of images and several criterias taken into account for the final system. Experiments will show the different methods studied in order to improve the performance of the system. After that, all the results will be compared in the following section, Overview of the Results. Finally, Conclusions and Future work will emphasize the achievements of this project together with future experiments that could significantly improve it.
Additionally, the Definitions, Acronyms and Abbreviations section could be useful when familiarising with some of the vocabulary related with this topic. Appendix section will show parts of the code of the final system along with a user manual to try and test it in a real environment.

2. Related Work

Interest in automatic Facial Emotion Recognition (FER) has been increasing over the years along with the number of techniques to tackle this matter [6]. Nonetheless, there are two essential common steps in every approach: **offline training** and **online prediction** [7].

Offline training consists on applying a machine learning classifier to a consistent set of images in order to teach the system to recognize, in this case, the seven possible macro-expressions. Additionally, the training time could vary between hours or days depending on the size of the database and images, along with the quality of the computer's hardware. On the contrary, *online prediction* defines the process of accepting new images or videos to evaluate them through an already trained model, returning the final prediction in a short period of time, without exceeding milliseconds or seconds.

Both, the *offline training* and *online prediction* procedures share the majority of the model characteristics. However, the database is just present in the *offline phase*, when training and evaluating the model, becoming an essential step for the future learning of the model. Therefore, every FER system could be divided into two different procedures: Data Acquisition and Model Definition.

2.1. Data Acquisition

The database used to train the system will notoriously affect the performance of the final system, as the model will learn what is taught. Hence, the motto "Garbage in, garbage out" [8]. Therefore, it is highly important to dispose of a trustworthy dataset of images with their respective verified labels.

One of the first datasets that merged measurements of facial motions with expression recognition was created by two experts in robotics, Takeo Kanade and Yingli Tian, and one expert in psychology Jeffrey F. Cohn in 2000 [9]. They were the first in describing the problem "space" for facial expression analysis, which tries to take into account the following considerations. First, they highlighted the importance on the level of description of the data, as they believed that obtaining fine-grained images was essential to codifying a Facial Action Coding System (FACS) [9]. In addition, they exposed and justified the relevance of transitions among expressions. Since, it is thought that expressions are singular and begin and end from a neutral position, but actually it is the opposite. Expressions may be born from another non-neutral expression or a sequence of them [9]. Also, they distinguished between deliberate and spontaneous expressions, they mentioned the reliability of manually coded expressions, as it is not easy for a subject to involuntarily arise a micro-expression, it could be inaccurately labelled [9]. Moreover, they concluded that, the eliciting conditions, individual differences in the subjects, the head orientation or the scene complexity of the whole sequence of information is very determinant when a database is being built [9]. Similarly, any dataset needs also to be big enough and cover all the possible expressions and emotions that could be detected in people of different ages, ethnicities, positions or scenarios. In particular, they worked on two different datasets. Initially, they created the first in 2000, it was called Cohn-Kanade database (CK) [9] and due to its wide use in training and testing FER algorithms, they improved it by building the other in 2010, called Extended Cohn-Kanade database (CK+) [10]. The original database went from having 1917 to 2339 image sequences, 182 to 210 subjects of different ethnic backgrounds and facial conformations and the images acquisition was performed using S-Video cameras of approximately 425 lines per frame, and digitized at frame rate omitting odd fields, whereas the images of the CK+ database were recorded using two hardware synchronized Panasonic AG-7500 cameras [9, 10].

A study performed by Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, Matti Pietikäinen introduced a novel database called SMIC, which includes 164 micro-expressions video clips elicited from 16 participants [41]. They built this database considering the lack of spontaneity in popular FER databases (CK, MMI, JAFFE, RU-FACS, Multi-PIE and Oulu-CASIA) [41]. Supporting that creating a proper dataset is a complex and long procedure, reflecting not only the importance of its content but also its size.

2.2. Model Definition

Nowadays, Facial Emotion Recognition (FER) models can be categorized into two groups: conventional and non-conventional approaches.

2.2.1. Conventional approaches

Conventional approaches split the FER process in three steps: **face and facial component detection**, **feature extraction** and **expression classification** [6].
First, the system receives an image of the subject, then it detects the region of the face and subsequently, its facial components or landmarks (eyes, nose, mouth, etc). Afterwards, the second step takes care of storing the spatial and temporal attributes of the point identified in the first step. Finally, the third step predicts the emotion associated to the input image using an already trained machine learning algorithm.

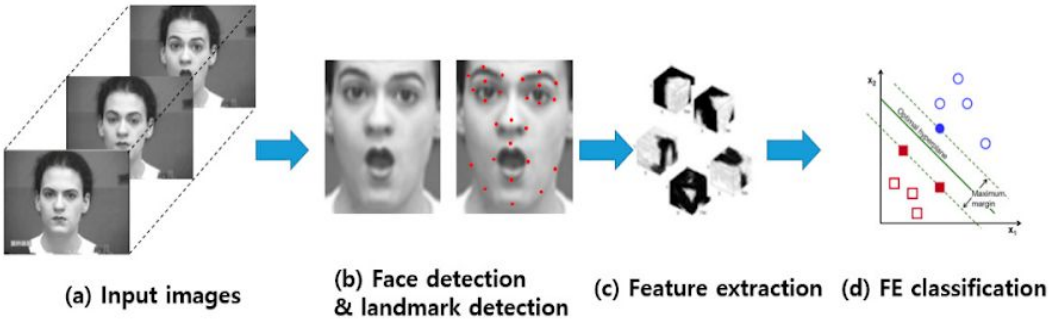


Figure 1: Four steps involved in conventional FER approaches, (a) shows the original images that the system will receive, (b) indicates the step explained in 2.2.1.1, (c) in 2.2.1.2 and (d) in 2.2.1.3 [6].

2.2.1.1. Face and Facial Component Detection

Once a database of images has been selected, it needs to be analyzed and treated in order to discard the irrelevant information comprehended in each one of the individual images, considering that the facial expressions of the subjects are the main focus.

Figure 2 shows two images belonging to the same database, where both subjects represent the same emotion: surprise. However, the differences between them are notorious (background, distance between the subject and the camera and physical appearances among subjects). It can be appreciated that if the system would receive these images with no previous processing, it could learn no relevant information regarding the facial expression and badly classify future images.

Figure 2: Two images extracted from the same database [CK+] where both subjects express the Surprise emotion. Nonetheless, the images present several differences regarding the background, distance from the camera and physical appearances between subjects.

In other words, a system trained with non "refined" images, similar to the ones illustrated by Figure 2, would probably store no useful information regarding the emotion. On the contrary, it would remember every pixel of the picture, every detail, like the line of the wall in the background, their clothes and gadgets, and even their face and body characteristics (hair, ears, eyebrows and so on). Therefore, the system would probably associate images of the same subjects (Figure 3 and Figure 4) rather than images of the same macro-expression (Figure 2), if no preprocessing or feature extraction (explained in the next section) techniques are applied.

Figure 3: First subject of Figue 2 [CK+ database], showing the Surprise and Sadness emotion in two individual images. A classifier trained with no preprocessed images or without performing any feature extraction would probably find more similarities between them than between images of different subjects producing the same emotion.

Figure 4: Second subject of Figue 2 [CK+ database], showing the Surprise and Sadness emotion in two individual images. A classifier trained with no preprocessed images or without performing any feature extraction would probably find more similarities between them than between images of different subjects producing the same emotion.

In addition, in the *online prediction* the images to be analysed will probably be wilder that the ones recorded in a studio, as it will be real scenarios where they for sure would have not be refined beforehand. That is to say, it is normal that the image could contain a lot of

useless information, such as, the landscape or body of the subject. Figure 5 shows an input example of the system, which would be analysed in order to identify the subject's emotion.

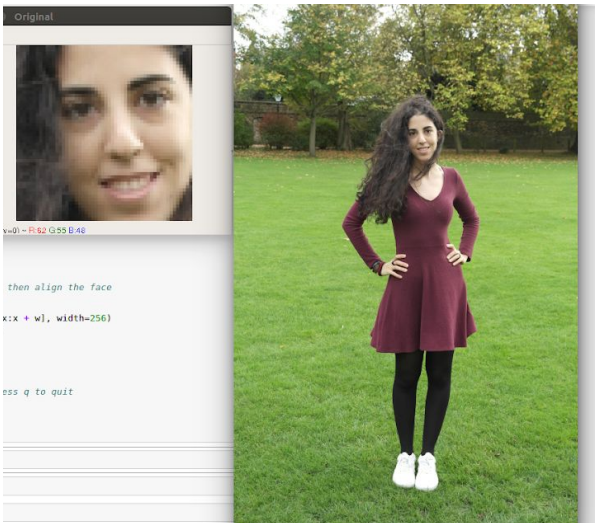


Figure 5: Picture taken outside of a studio in which the region of the subject's face is detected and cropped for its future FER prediction.

Thus, detecting the face among all the chaos in pictures happens to be one of the most fundamental steps to expect accurate predictions.

2.2.1.2. Feature Extraction

The way in which all the images are represented into a computer can be appreciated in Figure 7. They are stored as multidimensional arrays in which every position represents the Red Green Blue (RGB) additive color mixing of an specific pixel. On the other hand, every machine learning classifier receives input as numerical arrays. Even so, it is convenient to build an standard format input among all the images of the dataset, thus each array representing the image has to be normalised. In other words, every pixel value would be comprehended between 0 and 1. Figure 8 represents the image illustrated in Figure 6 after being normalised.

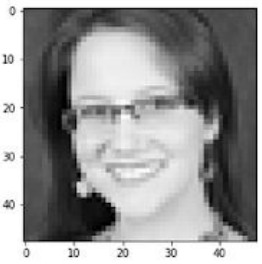


Figure 6: Visual image, size 48x48 pixels.

```
array([[ 77.,  78.,  79., ...,  74.,  77.,  76.],
       [ 83.,  84.,  82., ...,  75.,  79.,  77.],
       [ 84.,  87.,  84., ...,  74.,  79.,  79.],
       ...,
       [ 65.,  70.,  66., ...,  63.,  77.,  68.],
       [ 66.,  69.,  75., ...,  45.,  69.,  76.],
       [ 69.,  80.,  77., ..., 125.,  67.,  68.]], dtype=float32)
```

Figure 7: Numerical representation of the image plotted in Figure 6.

```
array([[0.3019608 , 0.30588236, 0.30980393, ..., 0.2901961 , 0.3019608 ,
        0.29803923],
       [0.3254902 , 0.32941177, 0.32156864, ..., 0.29411766, 0.30980393,
        0.3019608 ],
       [0.32941177, 0.34117648, 0.32941177, ..., 0.2901961 , 0.30980393,
        0.30980393],
       ...,
       [0.25490198, 0.27450982, 0.25882354, ..., 0.24705882, 0.3019608 ,
        0.23529412],
       [0.25882354, 0.27058825, 0.29411766, ..., 0.1764706 , 0.27058825,
        0.29803923],
       [0.27058825, 0.3137255 , 0.3019608 , ..., 0.49019608, 0.2627451 ,
        0.26666668]], dtype=float32)
```

Figure 8: Normalised representation of the image plotted in Figure 6.

Nevertheless, regarding FER scenarios not all the pixels of the image are relevant. In the previous section it was explained the importance of detecting the face and facial components. Namely, instead of sending to the model the array shown in Figure 8, feature extraction techniques aim in building a new array called descriptor, by storing only information regarding some relevant facial components along with some extra information (position, angle, etc - this will be decided by the creator of the feature extractor algorithm). The representation of each individual relevant component is defined as feature vector [11].
Coming back to the Figure 2 instance, in order to only store useful information about the emotion of the subject, it could be though to keep information about the position of certain zones of the subject's face, as shown in Figure 9. In this particular example, the descriptor of the images will be constituted by 13 feature vectors (blue points), each one representing a relevant zone of the face storing its coordinates. This is a mere example, to illustrate the way in which feature extractors are created.

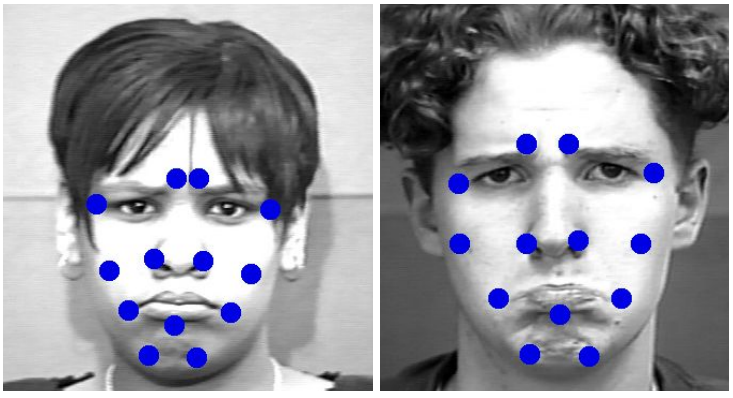


Figure 9: Graphic illustration of the relevant facial components chosen as example, to represent the emotion that the subjects are emitting. Each blue point will be represented as a feature vector and the set gathering all of them will represent the descriptor of the image.

The question now would be to determine the best descriptor, Which would be the optimal blue point to choose? Which would add the maximum quantity of relevant information and the less quantity of useless data?

Krystian Mikolajczyk and Cordelia Schmid asserted a study in 2010 about the performance of different descriptors which was published by the IEEE Transactions on Pattern Analysis and

Machine Intelligence [12]. This comparative study included the following ones: shape context, steerable filters, Scale Invariant Features Transform (SIFT), differential invariants, spin images, PCA-SIFT, complex filters, moment invariants and cross-correlation for different types of interest regions [12]. The experiment was concluded proving that the SIFT descriptor was the most efficient among the rest, using as the evaluation method, recall respect to precision [12].

On the other hand, Tomas Pfister, Xiaobai Li, Guoying Zhao and Matti Pietikäinen, researchers in the Department of Computer Science and Engineering at the University of Oulu (Finland), decided on developing their own descriptor [13]. They created a Temporal Interpolation Model (TIM) together with some state-of-the-art machine learning approaches to recognise spontaneous facial micro-expressions. TIM focuses in normalizing the face geometry by cropping it according to the eyes position, which are detected by using a Haar eye detector [13]. Then, the feature points are detected utilizing an Active Shape Model (ASM) deformation. They use graph embedding to interpolate images at arbitrary positions within a micro-expression. That strategy allows them to input a flexible number of frames to the descriptor, quite beneficial for very short expressions. In particular, they treated a video of a single micro-expression as a set of images sampled along a curve and then, they created a continuous function in a low-dimensional manifold [13]. The video is represented by a path graph and the vertices of the graph correspond to video frames. Also, they proved that TIM enables the system to accept images obtained by a 100 frames per seconds camera even with a 25 frames per seconds rate [13]. They were pioneers in successfully recognizing spontaneous micro-expressions with an accuracy higher than one obtained by a human.

In addition, Senya Polikovsky, Yoshinari Kameda and Yuichi Ohta, researchers at the University of Tsukub (Japan), developed another innovative descriptor [14]. On the opposite that in the TIM study, they used high quality cameras to record the face of the subjects, specifically 200 frames per second cameras [14]. Given that the structure and movements of the facial muscles are fairly complicated, recognising all at once could be a tedious task. Unlike Barlett [15] who applied the technique Gabor filter to all the facial area at the same time, they divided the face into 12 areas of interest [14]. In order to chose these 12 regions in which the face would be split, they followed the Facial Action Coding System (FACS) that decomposes facial expressions in terms of 46 component movements. See Figure 10.

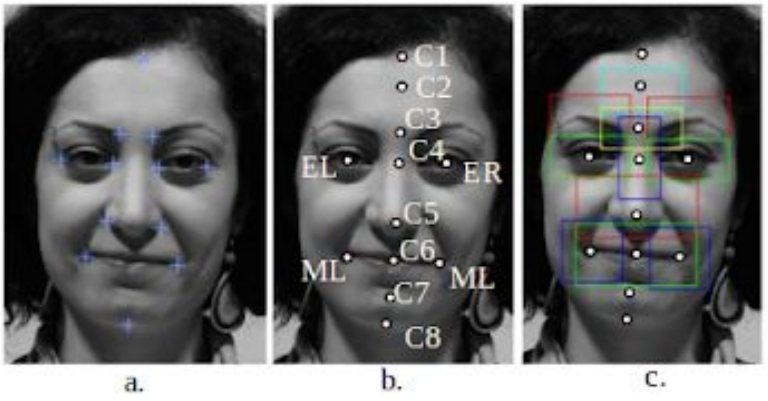


Figure 10: (a.) Represents the points of the first frame which are manually selected. Then, (b.) shows the centers of those points and finally, (c.) illustrates the twelve final facial regions represented as cubes [14].

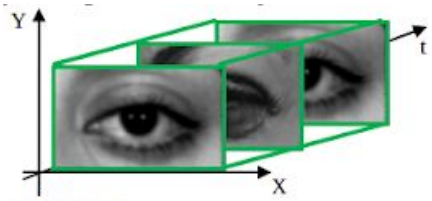


Figure 11: 3D cube that represents the region of the left eye, it will have X, Y and temporal dimensions [14].

Additionally, they highlighted the importance of the eyes size, as the size and location of other facial features are organized in proportion to them [14]. With this information, they built a table representing the name, size and location of every individual region. Figure 12 shows an example of it.

	Name	Centre Point	Size Height/Width [EyeUnits]
1	Forehead	C2	1 x 2
2	Left eyebrow	EL	2 x 2
3	Right eyebrow	EL	1 x 2
4	Left eye	ER	2 x 2
5	Right eye	ER	1 x 2
6	Between the eyes	C3	1 x 2
7	Upper Nose	C4	2 x 1
8	Lower Nose	C5	1.5 x 3
9	Mouth	C6	1.5 x 3
10	Left mouth corner	ML	1.5 x 1.5
11	Right mouth corner	MR	1.5 x 1.5
12	Chin	C7	1 X 1

Figure 12: Table comprehending all of the 12 selected regions, indicating their name, location and size.

In order to study the facial movements, they analysed a video sequence by calculating their partial derivatives, magnitudes and orientations, and then, they computed gradient orientation histograms for every frame, concatenating them to one feature vector and normalizing it afterwards [14]. Doing that, every motion between every frame in the video was represented by 32 *bins* histogram vectors [14]. They summarise their experiment by dividing the cubes in 8 groups which would be classified and predicted with a notorious precision [14]. This experiment is explained more deeply in the next section.

In general, it is important to notice that, many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction [16].

2.2.1.3. Expression Classification

This is the main part of the system, where a classification model is going to be defined, developed and trained in order to be capable of detecting facial expressions and predicting emotions over new data.

Nowadays, there are plenty of machine learning approaches to deal with image classification. Essentially, all of them could be divided into two broad categories: supervised learning, when the algorithm disposes of labelled input data and unsupervised learning, on the opposite. Additionally, the facial emotion recognition (FER) problem could be tackled in two ways. First, if the number of facial expression has been previously defined (like in the macro-expressions scenario), a supervised learning approach could be chosen, being then a multi-class classification problem with single label per class. In other words, there would be 7 possible classes, however, every image would be assigned just one of them. Secondly, if the number of facial expressions is unknown (like in the micro-expressions scenario), an unsupervised learning approach should be considered.

The creators of the TIM descriptor introduced before, used the following supervised classifiers to build the system and study its accuracy: Support Vector Machine (SVM), Multiple Kernel Learning (MKL) and Random Forest (RF) [13]. In their study, they performed three experiments: the first one was called "YorkDDT Corpus", the second one "SMIC Corpus" and the third one "Recognising Micro-expressions with Standard 25fps Frame Rate".
YorkDDT experiment was mainly focused in three phases which can be seen in Figure 13. Phase 1 consisted on differentiating micro-expressions from other facial activity. To that end, they randomly selected 18 images from sections free of micro-expressions (allowing macro-expressions if any). The best results were obtained by using the MKL classifier along with the TIM descriptor to 10 fps (as the videos were basically short, more frames added redundancy), achieving 83% accuracy [13]. Phase 2 focused in both, distinguishing false from truthful micro-expressions and also, detecting emotional and non-emotional gestures. One more time, MKL model was better than SVM and the TIM descriptor significantly increased its performance, obtaining 76.2% and 71.5% accuracy [13].

Phase	Classes	Method	Accuracy (%)
1	detection	SVM	65.0
1	detection	MKL	67.0
1	detection	MKL+TIM10	**83.0**
2	lie/truth	SVM	47.6
2	lie/truth	MKL	57.1
2	lie/truth	MKL+TIM10	**76.2**
2	emo/¬emo	SVM	69.5
2	emo/¬emo	MKL	**71.5**
2	emo/¬emo	MKL+TIM10	**71.5**

Figure 13: Table showing the different phases of the "YorkDDT" experiment, along with the supervised classifier used and the accuracy obtained[13].

SMIC experiment was also divided into two phases. One more time, Phase 1 focused on differentiating between micro-expression and other facial expressions, but introducing new classifiers and descriptors. Figure 14 shows the results of the experiment, situating RF classifier along the TIM descriptor to 10 frames per second (fps) as the best option, achieving 74.3% accuracy. Additionally, another experiment was carried out in Phase 2, it consisted on contradistinguish positive from negative micro-expressions. This time, MKL plus TIM to 10 fps was the optimal model, obtaining 71.4% accuracy.

Phase	Classes	Method	Accuracy (%)
1	detection	RF+TIM15	67.7
1	detection	SVM	70.3
1	detection	RF+TIM20	70.3
1	detection	MKL	71.4
1	detection	RF+TIM10	**74.3**
2	neg/pos	SVM	54.2
2	neg/pos	SVM+TIM15	59.8
2	neg/pos	MKL	60.2
2	neg/pos	MKL+TIM10	**71.4**

Figure 14: Table showing the different phases of the "SMIC" experiment, along with the supervised classifier used and the accuracy obtained[13].

The intention of the third experiment was to prove that micro-expressions could be identified by using images obtained through standard 25 fps video cameras without losing accuracy. Figure 15 demonstrates that again, RF to detect micro-expressions from others and MKL to differentiate positive from negative micro-expressions were the best classifiers, notwithstanding the best TIM descriptor went from 10 to 20 fps for Phase 1 and from 10 to 15 fps for Phase 2. The experts concluded that earning 4.9% more in Phase 1 and losing 6.3% in Phase 2, was not surprising. The process of detecting just requires a slight movement on the facial region, instead classifying the micro-expression type, even if it is just between two classes, requires detailed spatiotemporal data about the movement, therefore the frame rate is more relevant [13].

Phase	Classes	Method	Accuracy (%)
1	detection	RF+TIM10	58.5
1	detection	SVM+TIM10	65.0
1	detection	MKL+TIM10	70.3
1	detection	RF+TIM15	76.3
1	detection	RF+TIM20	**78.9**
2	neg/pos	SVM+TIM10	51.4
2	neg/pos	MKL+TIM10	60.0
2	neg/pos	MKL+TIM10	60.0
2	neg/pos	SVM+TIM15	62.8
2	neg/pos	MKL+TIM15	**64.9**

Figure 15: Table showing the different phases of the "Recognising Micro-expressions with Standard 25fps Frame Rate" experiment, along with the supervised classifier used and the accuracy obtained [13].

On the other hand, the researchers of the University of Tsuku [14], implemented a model in Matlab using "Piotr's Image & Video Toolbox" which allowed them to choose among the following parameters: the values of the 3D Gaussian smooth filter, the magnitude cut-off of the threshold and the size of the cubes [14].

The facial regions of the subjects were split into 8 groups: forehand, eyebrows, eyes, zone between the eyes, lower nose, mouth, mouth corners and chin. Being that they used K-means as the classifier, 8 clusters were automatically created, one per group. Figure 16 shows the accuracy of detecting the micro-expression represented on any of the 8 facial regions, which is notoriously high. Being the forehead the highest region with 95% of accuracy detection and the mouth corners along the chin zone the areas with less accuracy, regarding the beards of

some subjects [14]. Additionally, they created the following three stages: “Constrict”, “In-Action” and “Release” in order to measure the time of the micro-expressions [14]. Figure 16 also illustrate the accuracy of identifying each one of these concepts. “Constrict” comprehens the constriction of the muscle produced on the first stage of the micro-expression, and it goes from 31 to 43 frames lasting approximately 0.065 seconds. “In-Action” refers to the time that the action itself lasts, going from 44 to 73 frames (0.15s) and “Release” captures the final part of the micro-expression, reflecting the release of the muscle, going from 74 to 85 frames (0.055s) [14]. They believed that this time-measuring ability would be useful in psychology to measure hostile intent and damage behaviour detection [14].

Facial Cubes	Neutral	FACS AU	Constrict	In-Action	Release
a) Forehead	0.95	AU2	0.93	0.95	0.91
b)Eyebrows	0.93	AU4	0.84	0.83	0.9
	-	AU5	0.86	0.83	0.85
c) Eyes	0.92	AU4	0.84	0.83	0.84
	-	AU7	0.86	0.8	0.81
	-	AU43	0.85	0.85	0.84
d) Between the Eyes	0.9	AU4	0.93	0.9	0.84
e) Lower Nose	0.94	AU10	0.95	0.93	0.95
f) Mouth	0.88	AU12	0.81	0.79	0.85
	-	AU24	0.81	0.67	0.79
	-	AU26	0.83	0.77	0.8
g) Mouth corners	0.83	AU13	0.85	0.81	0.89
h) Chin	0.89	AU17	0.83	0.83	0.84

Figure 16: Table showing the experiment performed by Senya Polikovsky, Yoshinari Kameda and Yuichi Ohta [14].

Summarising, experts in recognising or identifying macro-expressions and/or micro-expressions decide on conventional approaches as they believe that implementing their own or choosing an state-of-the-art descriptor, will significantly improve the performance of the system.

2.2.2. Non-conventional approaches

On the other hand, non-conventional approaches simplify the use of models based on face-physics by enabling “end-to-end” learning [6]. These approaches are defined as **deep learning** and reduce for the programmer the number of steps of traditional FER models to one, receiving non-treated images and returning their equivalent prediction. This is possible due to the way in which deep learning operates, it internally performs the optimal feature extraction for the images, so, it is enough to indicate their type and resolution besides different customizable parameters (explained in the next subsections). At the same time, several deep learning approaches can be found, the most demanded in the Facial Emotion Recognition field are: **Convolutional Neural Networks (CNN)**, and **Hybrid Systems** combining CNN with spatio temporal Recursive Neural Networks (RNN). Figure 17 very generically describes the basic operations of a deep learning system. Nevertheless, it could be relevant to show the internal

operation of a simple standard Neural Network to achieve a better understanding of the situation.

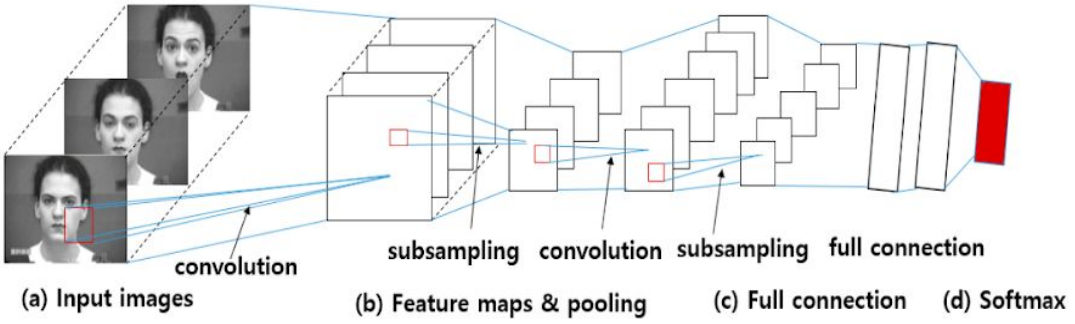


Figure 17: Steps involved in non-conventional FER approaches, (a) shows the original images that the system will receive, (b)(c)(d) indicates the internal steps performed by a deep learning architecture [6].

Figure 18 illustrates the structure of a simple neural network (NN). Every neural network is divided in three layers: the **Input layer**, which receives the input data and performs some operations over it, sending the results to the next layer, the **Hidden layer**. It operates a similar behaviour, receiving as input the output of the first layer and generating a new output, which will be sent to the third layer, the **Output layer**. It will receive that value, compute some operations and finally, it will generate the final output of the neural network. In this specific model, the NN takes in two inputs and generates one output.

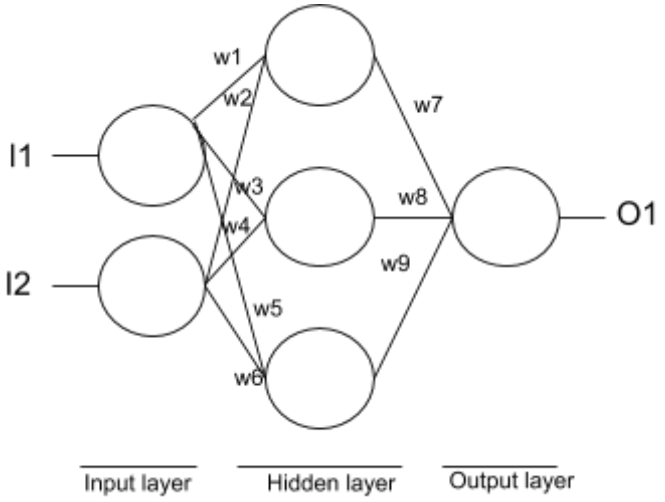


Figure 18: Representation of a simple neural network with 2 neurons in the input layer, 3 neurons in the hidden layer and 1 neurons in the output layer.

As a practical example, this model could be used to perform sum operations. I1 and I2 could be two random numbers and O1 the equivalent sum of both inputs. To do so, in *offline training*, large quantities of data similar to the one shown in Figure 19 would be needed, along with the backpropagation method which would allow the model to learn.

Backpropagation consists on establishing the value of the weights connecting neurons (w1, w2 …, w9). Initially, these weights would have a random value, which is updated with every iteration of a training pattern (row of the table shown in Figure 19). That is to say, for every training pattern, the NN would try to predict one specific output (O1) given two inputs (I1 and

I2). Then, it will calculate the root mean square error (RMSE) comparing the obtained to the real output values. Depending on the RMSE, the weights of the NN will be updated, this process will be repeated for a specific number of "Epochs". Finding that number is essential to establish the best weights of the final NN, as the equilibrium between learning a lot of information and not learning it with all possible detail is fundamental to hit future predictions. ("Epoch" defines when the backpropagation process has learnt over all the rows of the dataset). That way, in *online prediction*, the system could face quick decisions when calculating new sums.

I1	I2	O1
1	1	2
1	2	3
1	3	4

Figure 19: Practical example of possible input data for the model shown in Figure 18, creating calculator that sums two inputs.

Additionally, every neuron is connected to others. That means that every neuron will sum the output values of all the neurons in the previous layer, multiplied by their associated weights. Then, the result of that equation would be applied to an activation function, which will produce the output of that individual neuron.

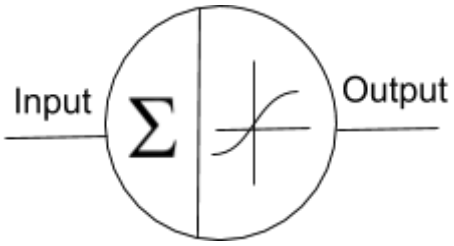


Figure 20: Graphic description of a neuron. It receives the sum of all their connections and it applies a specific activation function (the one in the picture is the sigmoid, being others like linear, softmax and so on) to that sum, producing an output.

Accordingly, deep learning architectures are kind of simple neural networks, but owning unknown/elevated quantities of hidden layers. Nevertheless, the internal operations between layers are essentially the same.

2.2.2.1. Convolutional Neural Networks (CNN)

A Convolutional Neural Network characterizes itself by owning three different types of "hidden" layers: **convolutional**, **pooling** and **fully connected** layers which can be seen in Figure 21.

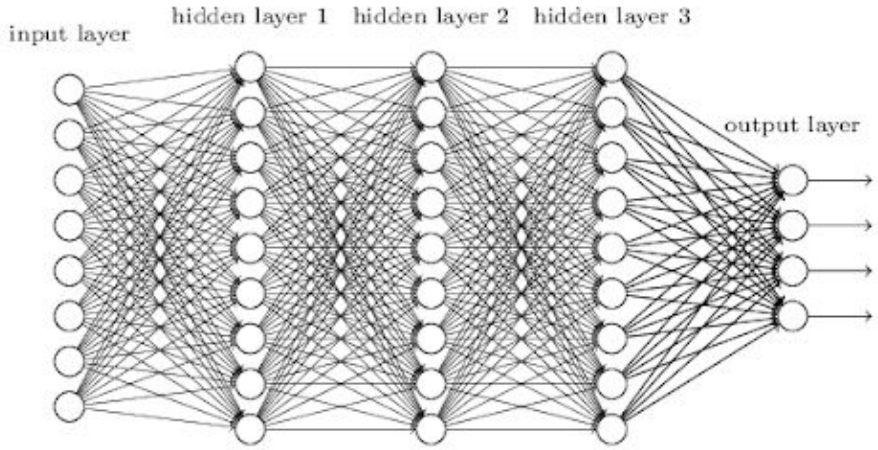


Figure 21: Convolutional Neural Network structure [23].

Convolutional layers are the first layers of the model and they are in charge of receiving the images and creating the corresponding descriptors. To do so, the input layer will create arrays containing localized areas of the original images. Since, instead of connecting every pixel of the image with every neuron of the next layer, several regions will arise, defined as filters [23]. Both, the number and size of filters, along with the number of pixels that the filter should be slid (defined as strides) to address the whole image, will be customizable.

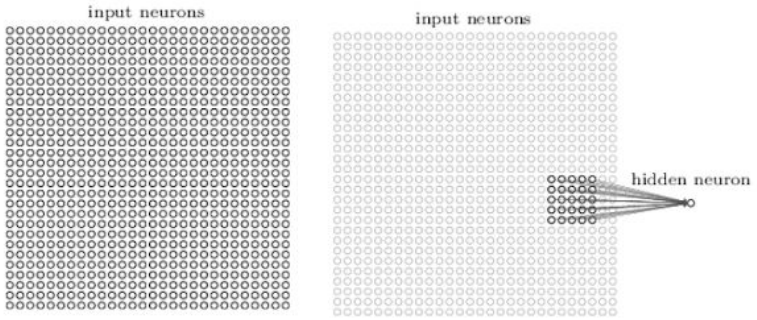


Figure 22: At the left, original image represented as 32x32 pixels array. At the right 5x5 pixels array defined as a filter of the image [23].

Subsequently, every filter will be analysed by a neuron of the *Convolutional* layer. This process is represented in Figure 23. Once the neuron have applied the activation function over the filter, it will return a number representing the filter. The set of all those numbers would be defined as feature map [23].

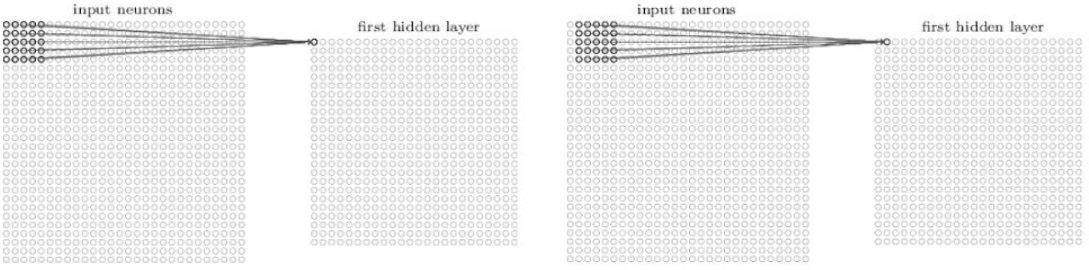


Figure 23: First filter represented by a 5x5 pixel array connected with the first neuron of the *Convolutional* layer. Then, second filter, also represented by a 5x5 pixel array but this time slid one pixel to the right, connected with the second neuron of the *Convolutional* layer[23].

Then, ***Pooling* layers** will act as simplifiers of the feature maps obtained by the *Convolutional* layers. It will define customizable regions over the feature map which each one will be reduce to one. Figure 24 shows an example of a 2x2 region, when performing the "max-pooling" operation. "Max-pooling" chooses the pixel with higher value among the four available in the 2x2 region. In this example the original feature map goes from 24x24 to 12x12 array [23].

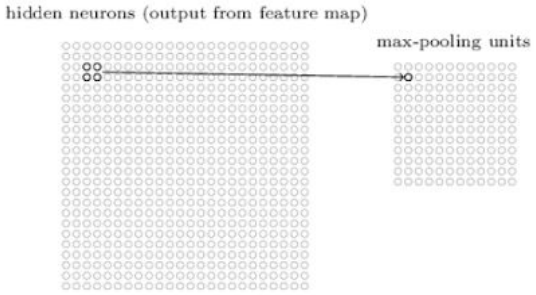


Figure 24: Feature map obtained by the *convolutional* layer, when choosing a 2x2 area to be performed a pooling operation. Specifically the "max-pooling" operation, reducing 4 pixels to 1, the one with higher value[23].

Finally, the ***Fully connected*** layer will connect every neuron of the *pooling* layers to their own neurons and the neurons of the last layer, the output layer [23]. The number of neurons of the output layer will be defined by the number of classes of the model.

Breuer and Kimmel, researchers of the department of Computer Science at the Israel Institute of Technology, employed several techniques in order to know the facial features that the CNN internally learns in every layer [24]. To do so, they performed some experiments using different datasets, in particular, CK+ [10], NovaEmotions and FER2013 [29].
The model defined for the experiments was constituted by the following structure. One *input layer*, which admitted both grayscale and color (RGB) images. Followed by 3 blocks, each one formed by one *convolution layer*, characterised by owning feature maps of dimensions 64, 128 and 256 pixels, with filters of size 5x5 pixels and the Rectified Linear Unit (Relu) activation function, and one *pooling layer* with 2x2 pool size. Ending with one *fully connected layer* with 512 hidden neurons, connected to the *output layer* formed by 8 neurons, one per macro-expression[24]. In addition, they applied dropout[25] operations between the fully connected layers and after the last convolution layer, in order to reduce overfitting (with probabilities of 0.5 and 0.25) [24]. Finally, they trained the model using the ADAM [26] optimizer with a learning rate of 1e − 3 and a decay rate of 1e − 5 and expanding the datasets mentioned before by combining random flips and affine transformations of the images [24].
Applying Zeiler and Springenber's methods [27], they were capable of visualizing the filters obtained by the model through *offline training*. Proving that the first layers produce lower level filters whereas layers closer to the output provide higher level filters, with human observable features [24]. Watch Figure 25.

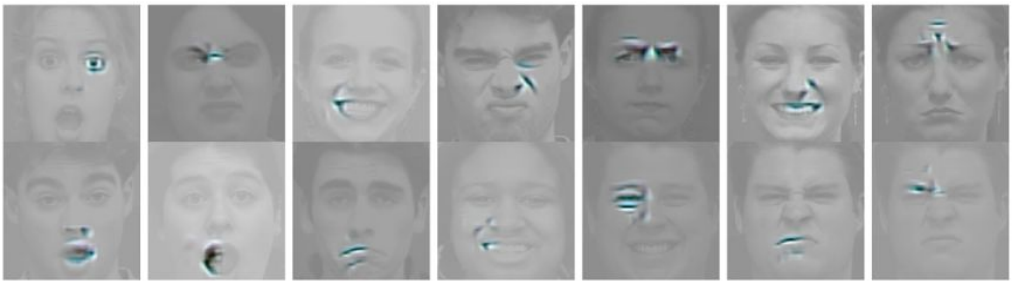


Figure 25: Graphic visualization of the filters generated in the layers closer to the output of the CNN, where facial features can be identified by an human eye and are quite similar to the FACS Action Units (AU) defined by Paul Ekman [24].

On the other hand, CNN architectures usually require elevated amounts of data to obtain proper future predictions [33]. Andre Teixeira Lopes, Edilson de Aguiar, Alberto F. De Souza and Thiago Oliveira-Santos, researchers at the Universidade Federal do Espírito Santo (Brazil), performed a study to tackle this problem [30]. They built a system combining CNN along with different image preprocessing techniques. For that purpose, they used three databases: Extended Cohn-Kanade (CK+) [10], Japanese Female Facial Expression (JAFFAE) [31] and Binghamton University 3D Facial Expression (BU-3DFE) [32], using some for training and others for testing. Figure 27 overviews the way in which the images are received by the CNN. In *offline training*, every image is slightly rotated by a 2D Gaussian distribution (σ = 3 pixels and μ = 0) to introduce random noise in the locations of the center of the eyes [30]. See Figure 26. That way, every image will produce 70 additional synthetic images, notoriously increasing the size of the database [30]. Once the database has been extended, before sending the images to the model, every single image will be processed (also every image of *online prediction* will follow this preprocessing) [30]. First, the picture will be aligned placing the subject's eyes in a straight line. Then, it will be cropped, tring to reduce the facial area to the minimum, removing not relevant information. Additionally, the size of the image will be reduced to 32x32 pixels. Finally, its light intensity will be standardized trying to shorten the physical and racial differences among the subjects [30].

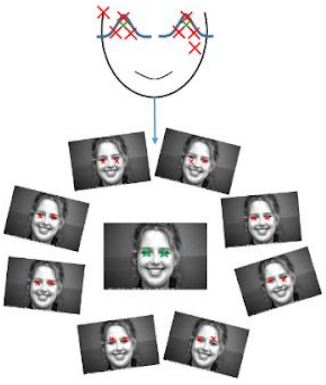


Figure 26: Synthetics images produced when rotating over a 2D Gaussian distribution [30].

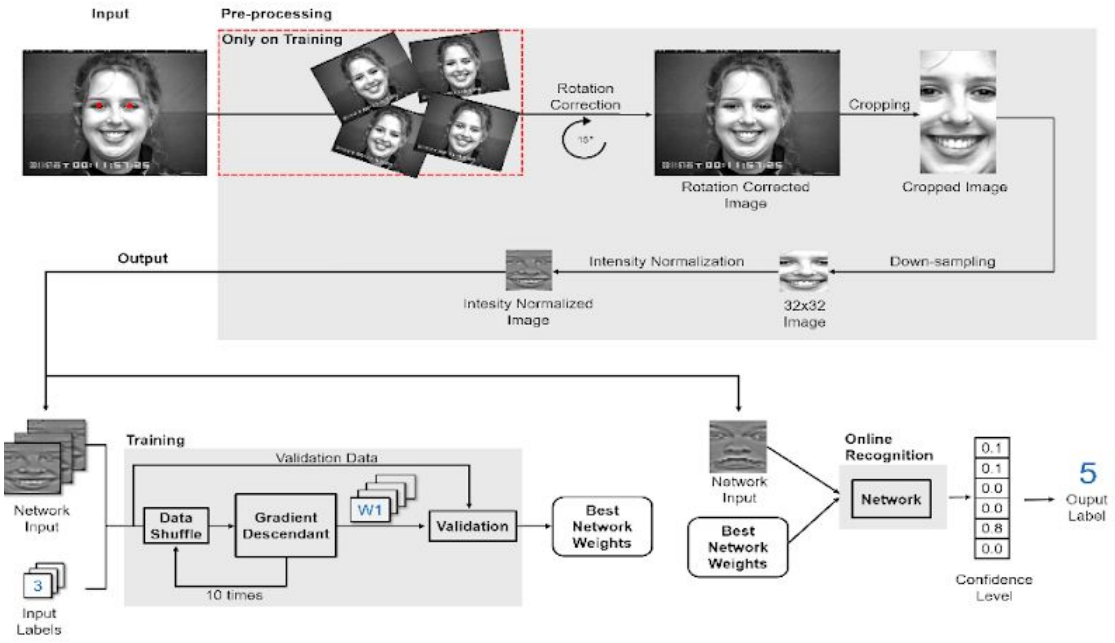


Figure 27: *Offline training* and *online prediction* of the model proposed by Andre Teixeira Lopes, Edilson de Aguiar, Alberto F. De Souza and Thiago Oliveira-Santos [30].

At the same time, the model they used is represented in Figure 28 [30]. The *input layer* of the CNN initially receives 32x32 pixels gray-scales images. Next, a *convolution layer* returns 32 feature maps (with size 28x28 pixels) created by picking filters of size 5x5. It is followed by a *pooling layer* which uses the max-pooling operation (with 2x2 pool's size), reducing the image to 14x14 pixels. Subsequently, the second *convolution layer* appears, generating 64 feature maps (with size 8x8 pixels) created by picking filters of 7x7. Followed by the second *pooling layer* with same parameters that the previous one. Then, the *fully connected* layer, with 256 neurons, establishes connections between all the neuron of the previous and *output layer*. The *output layer* will have seven neurons, one per macro-expression.

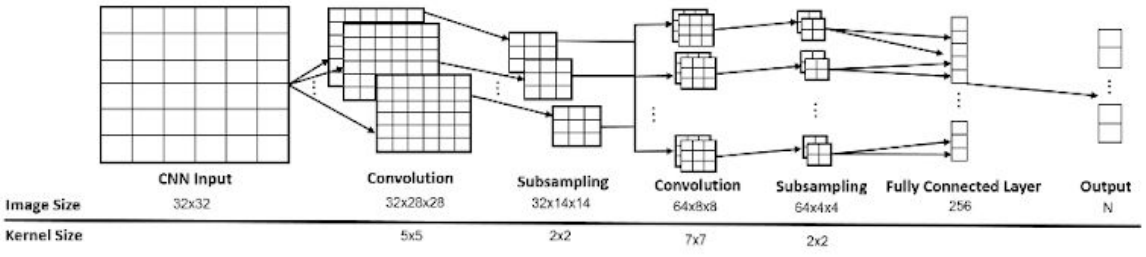


Figure 28: Structure of the model proposed by Andre Teixeira Lopes, Edilson de Aguiar, Alberto F. De Souza and Thiago Oliveira-Santos [30].

They obtained the best results, training with the CK+ dataset and the previously explained model, reaching 96.76% accuracy by predicting over 6 instead of 7 macro-expressions defined by Paul Ekman [30].

2.2.2.2. Hybrid Systems: Convolutional Neural Network (CNN) + Long Short Term Memory (LSTM)

LSTM is a special type of Recursive Neural Network (RNN), capable of learning long-term dependencies among sequences of images[28]. Unlike non-recursive neural networks, LSTM characterises itself by owning four iterating modules: the **cell state** , **input gate**, **forget gate** and **output gate** [6]. Each of these gates could be seen as "standard" neurons in a feed-forward neural network. Being that, they compute an activation (using an activation function) of a weighted sum, with the difference of representing the activations according to a time step.

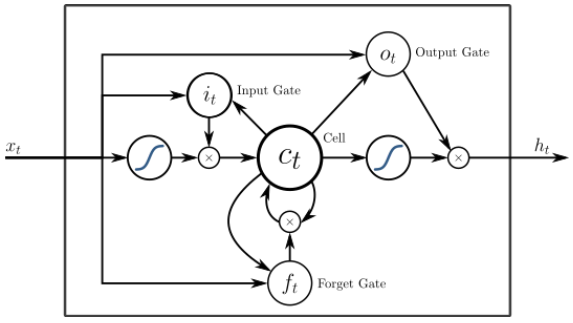


Figure 28: Peephole LSTM structure [28].

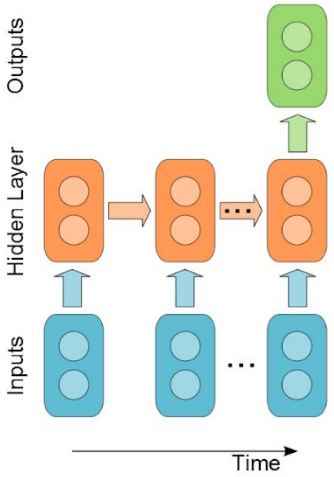


Figure 29: Hybrid system structure [34].

It has been proven that hybrid systems combining CNN with RNN, specially LTSM, produce notorious results when analysing videos [34][35][36]. That is to say, the hybrid model is capable of recognising and synthesizing temporal dynamics of facial expressions presented in sequences of images [6]. Internally, the CNN will determine the facial features present on individual static frames and the LTSM will establish temporal relations among them.

Ebrahimi Kahou, Michalski, Konda, Memisevic, and Pal, researchers of Frankfurt's and Montreal's universities, took part into a competition denominated Emotion recognition in the wild (EmotiW 2015) [34]. The goal was to create a system capable of identifying and classifying new "wild" images into their corresponding macro-expression. For that task, the Acted Facial Expressions in the Wild (AFEW) database [37] was facilitated. It gathers short video clips extracted from Hollywood movies, in particular 723 videos for training, 383 for validation and 539 for test. For that purpose, they developed a hybrid system combining RNN with CNN. Firstly, they trained the CNN to classify static images and then, they used RNN to predict a single emotion in the entire video clip. They performed several experiments modifying the size of the filters of the CNN and the temporal number of frames of the RNN. Obtaining 52%, 68% and 47% accuracy in train, validation and test. Reaching the best accuracy in the test set (52%) when mixing the train and validation set to train the model [34].

3. Proposed Approach

This project aims to create a system capable of identifying and classifying facial macro-expressions in natural daily environments. Specifically, this macro-expressions are the ones defined by Paul Ekman: Happiness, Sadness, Surprise, Fear, Anger, Disgust and Neutral. Also, the system will accept two different types of input: individual or online-video-stream images. As mentioned before, both input types will record wild and uncontrolled pictures, which means that the number of people in the image, the background, the distance, angle and rotation of the person with respect to the focus of the camera is flexible.

3.1. Database

For the reasons mentioned above, FER2013 is the chosen dataset to train the model proposed in this project, it was created in 2013 by Pierre Luc Carrier and Aaron Courville [29] by gathering internet images. Using the Google image search API, they established 184 emotion-related keywords (like blissful, enraged and so on) combined with gender, age and ethnicity words, achieving around 600 searching queries. The images obtained by the queries were processed by OpenCV in order to detect the region of the face to crop it afterwards. Then, human labelers assigned and verified labels to every single image, relating one of the 7 macro-expression to the associated image. In addition, human labelers also corrected the cropping if necessary, and filtered out duplicates. Finally, all the images were standardized by both, reducing their size to 48x48 pixels and converting their color type to grayscale [29].

The resulting dataset contains **35887 images**, of which 28709 are train samples, 3589 test samples and 3589 another alternative test samples. Additionally, considering the three sets, the number of images per class is not proportional, there are different number of images depending on the class, the exact quantity of images per class of the whole database is as follows: **547 "Disgust"**, **4002 "Surprise"**, **4953 "Anger"**, **5121 "Fear"**, **6077 "Sadness"**, **6198 "Neutral"** and **8989 "Happiness"** [29].

As it could be seen from Figure 30, the images of the dataset are not specially recorded in a studio for a FER task. The faces appear in different proximities, orientations and intensities. The subjects are different, in age, gender and ethnic, and even the quality of the pictures is not the same. Hence, dealing with a "wild" dataset could cause pros and cons. As the similarity increases according to real situations, it is more likely that the accuracy of the classifier decreases, being this a more complex scenario.

Figure 30: Batch of images of the FER2013 dataset. It can be appreciated their "wildness" and "naturalness".

3.2. Model definition

Trying to achieve state-of-the-art results by using non-conventional approaches, the model defined in this project implements a Convolutional Neural Network Structure (CNN), comprises of ten layers: 5 **Convolutional** layers, 3 **Pooling** layers and 2 **Fully Connected** layers.

Initially, the CNN receives grayscale images from the input layer which are sent to the first layer of convolutions. This first layer applies a convolution kernel of 5x5 pixels and the Rectified Linear Unit (Relu) activation function, returning 64 features maps. This layer is followed by a pooling layer which reduce every one of the 64 features maps by 5. It uses pool of size 5x5 pixels, strides of 2 pixels and the MaxPooling operation, which keeps the pixel with highest value among those 25 pixels of the kernel. Subsequently, two new convolution layers performs 64 convolutions with filters of 3x3 pixels, obtaining again 64 features maps per layer. The last of these convolution layers is followed by another pooling layer, with pools of 3x3 pixels, strides of 2 pixels and the AveragePooling operation, which computes the average among these 9 pixels keeping that value. One more time, two convolution layers similar to the ones before, but generating 128 instead of 64 feature maps are also followed by a pooling layer with the same parameters.

The outputs of theses layers are given to 2 fully connected hidden layers that have 1024 neurons and a dropout, which would reduce overfitting with probability of 0.2 [25]. Finally, all the outputs are connected with the output layer which is formed by 7 neurons, one per macro-expression.

All the decisions regarding the number of convolutions, size of the feature maps, filters and pools, number and type of the pooling operations and number of neurons in the fully connected layer have been considered by both, reading and comparing related papers [27,30] and empirically, training and testing with different parameters.

3.3. Training parameters & stopping criteria

Once the training set has been chosen and the model has been defined, the *offline training* phase needs to be performed. Every image of the training set will have an associated label indicating the macro-expression that the subject of the picture is mimicrying. At the same time, each single image will be sent to the defined classifier, CNN in this particular scenario, until reaching the optimal CNN's weights which will make the model capable of predicting over unseen images in *online prediction*. For that purpose, all the images comprehended in the dataset will be randomly split in batches. One batch will count as 1 "epoch", and then, the CNN will be trained a specific number of "Epochs". Finding that number becomes crucial when building this kind of system, as a very small number of "epochs" may cause underfitting, which means that the model have not learnt enough samples to perform proper future predictions, and a very high number of "epochs" may produce overfitting, which means that the CNN memorises the images of the training set in deep detail, storing not only information about the emotion but also other features, like the side and position of the facial components.

Therefore, performing several "epochs" experiments, it has been found that 19 is a good number of "epochs" along with 256 batches of images, for his particular dataset and model. Lower numbers of "epochs" showed that the obtained accuracy did not reach acceptable results considering the images of the training set were already seen by the classifier, and higher numbers showed that reaching certain percentage of accuracy the model plateaued. In addition, the optimizer is ADAM [26], the loss function (function in charge to minimize the error when calculating the new weights of the neuron's connections) is Categorical Crossentropy, as this is a multiclass classification problem where each example belongs to a single class, and the metric of the performance of the model is Accuracy, as we just want to know the percentage of classes which are classified correctly, independently if they are true positives, true negatives, false positives or false negatives.

4. Experiments & Results

In order to improve the performance of the model and given that deep learning approaches does not require of external feature extractors, different preprocessing techniques (Alignment, Crop, Light Intensity and combinations of them) have being considered to refined the input images at maximum level to make it easier for the CNN (detailed below).
Additionally, all the experiments were carried out using Linux (Ubuntu 16.04) as operative system, an Intel Core i7 7700 HQ as processor, a GeForce GTX 1050 Ti 4GB GDDR5 as GPU, Python as programming language, TensorFlow with Keras, CUDA, OpenCV and Matplotlib as libraries and Jupyter Notebook as programming environment. Also, all the experiments detailed below are produced by training the system 19 "epochs" with the parameters previously explained.

4.1. Original

The following results shown in Figure 31, 32 and 33 are obtained from evaluating the model described in the section before with no additional image preprocessing. Particularly, Figure 31 shows the error loss and accuracy of the system by predicting over the train and test sets. It can be seen that the model performs quite nicely when analysing images already seen (88.08%). Also, it is interesting to notice that the percentages obtained by the test set are quite favorable (56.42%), given the wildness of the images and the number of possible classes. Additionally, Figure 32 plots the confusion matrices regarding the same train and set sets, from which the number of images belonging to these sets can be obtained by summing all the rows and columns of its respective confusion matrix. Moreover, "True labels" axis indicates the number of images that contains each one of the seven emotions by summing all the columns, whereas "Predicted label" axis represents the number of images predicted with the respective emotion by summing all the rows of the corresponding emotion. Subsequently, the highlighted diagonal represents the images which are correctly classified, while the other positions show the incorrectly classified images. For example, the position of the left down corner of the train confusion matrix means that 95 images have been incorrectly classified by the CNN model, as they have been predicted with the Anger emotion, when their real associated macro-expression is Neutral. On the other hand, figure 33 shows the percentages of images correctly classified per class. It is noticeable that the macro-expressions with highest recognition rate are the Happiness and Surprise emotions, reaching 96% and 97% accuracy over the training set, and almost 80% and 78% accuracy over the test set.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.3386	88.0803%	1.8783	56.4224%

Figure 31: Accuracy regarding the training and test set respect to the 7 possible classes.

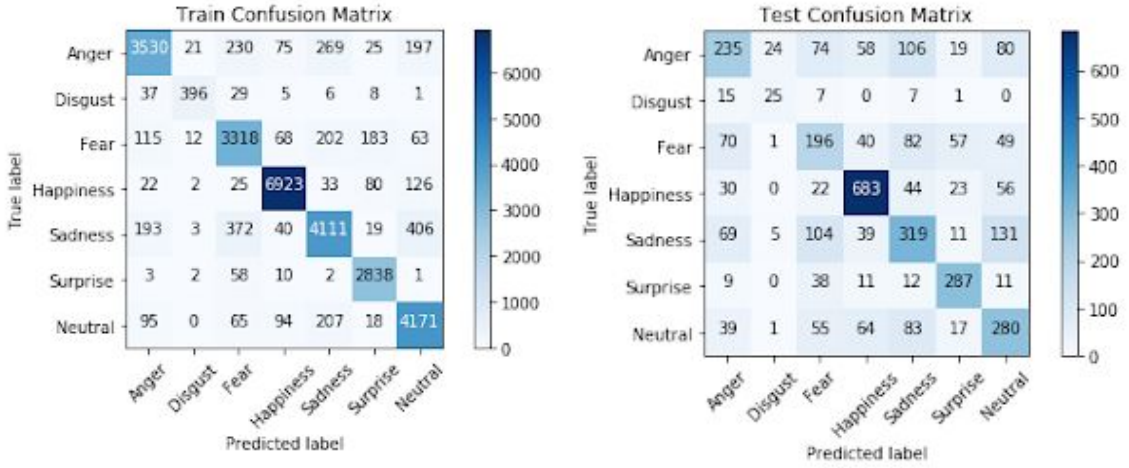


Figure 32: Confusion matrices of the train and test sets, predicting over the original model and reflecting the majority of the samples correctly classified.

Macro-expression	Train Accuracy (%)	Test Accuracy (%)
Anger	81.20	39.42
Disgust	82.15	45.45
Fear	83.76	39.59
Happiness	96.00	79.60
Sadness	79.91	47.05
Surprise	97.39	77.98
Neutral	89.69	51.94

Figure 33: Individual accuracy per class, outstanding the Happiness and Surprise emotions being the best classes identified by the model and the Fear emotion being the class with worst recognition rate.

4.2. Alignment

The images in the FER2013 database could vary in rotation among them [29]. It could be seen that, if the face is not aligned, not only the eyes are not in a straight position, but also the mouth, nose and other facial components. Due to CNN learns colors and positions of pixels, not dealing with these rotations which are not related with the macro-expression of the subject, could negatively affect the accuracy rate of the system [30].

To address this problem, using the CUDA library which allows real-time facial keypoints detection in uncontrolled environments, the center of the subject's eyes has been identified. Then, a rotation transformation is applied aligning the eyes with the horizontal axis of the image. Figure 34 shows a real example of the alignment process, the image of the left is the original and the image of the right is the result of applying the corresponding rotation.

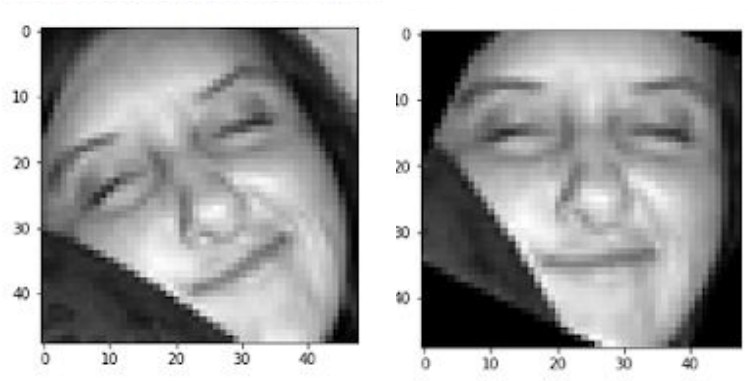


Figure 34: Original and final image, resulting from the "Alignment" preprocessing.

After running the experiment, it was found that aligning the images improved the original model performance by 5% and 2% over the train and test sets respectively. It is relevant to mention that the CUDA library is just capable of aligning the picture if it first, detects the face. As there are some images in which the face is not trivially illustrated, there is a percentage of

images which won't be aligned, exactly 30% of the whole dataset. Future experiments will deal with this issue. Subsequently, Figure 36 contains the individual accuracies per class over the test set. It can be notice that not all the macro-expressions have the same accuracy, fluctuating between 41.8% (Fear) and 81.78% (Happiness), which means the final system will predict better some emotions than others.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.2009	93.1624	2.1358	58.3449

Figure 35: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Test Accuracy (%)
Anger	45.40
Disgust	68.57
Fear	41.8
Happiness	81.78
Sadness	45.46
Surprise	74.05
Neutral	52.73

Figure 36: Individual accuracies per class over the test set.

Apart from the images which won't be aligned, there will be also other images which will present "black" lines, like the examples shown in Figure 37. The reason for the black background, resides in the alignment process, sometimes the image is totally twisted, so it may be autocomplete with black background in order to keep the same image size.

Figure 37: Batch of images after the "Alignment" preprocessing.

4.3. Crop

As explained in previous sections, the background of an image add a lot of useless information regarding the facial expression of the subject [30]. However, the background is not the only source of irrelevant data, there are also personal complements which can negatively affect the performance of the model, like the hairstyle, necklace, earing and so on. FER2013 database already presents images with barely background, even though the whole face of the subject could contain that kind of negative information, so, it might be convenient to crop the face from the mouth to the eyebrows. Nevertheless, some images are not aligned and cropping them could cause negative effects by losing relevant information, such as Figure 38. Section 4.5.2 presents the results of combining the Alignment and Cropping preprocessing techniques.

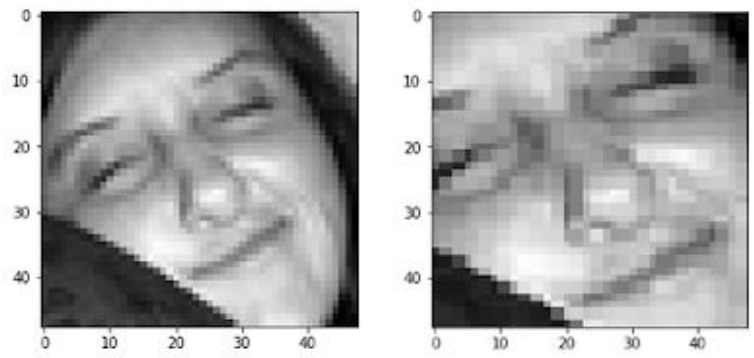


Figure 38: Original and final image, resulting from the "Crop" preprocessing. Cropping without aligning first could produce a negative effect, as relevant information could be lost. The left eye is cut by half.

Figure 39 shows the general accuracy of the Crop model. Probably, the accuracy of this model is inferior respect to the Original and the Alignment model because of the information lost when cropping. Even though, it still remains over the standards. In addition, Figure 40 situates the Fear and Anger emotions as the worst identified, keeping the Happiness and Surprise emotion, along the Disgust emotion as the best recognized.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.23774	91.7203	2.4243	51.90

Figure 39: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Test Accuracy (%)
Anger	38.54
Disgust	62.50
Fear	34.73
Happiness	69.01
Sadness	43.50

Surprise	69.26
Neutral	46.73

Figure 40: Individual accuracies per class, being Anger and Fear the classes with lowest recognition rate.

Figure 41 illustrates a batch example of the result of applying the "Crop" preprocessing to the original dataset, effectively it can be seen that several images appear with important facial components not clearly visible, like the images of the columns 1 and 7 of the second row.

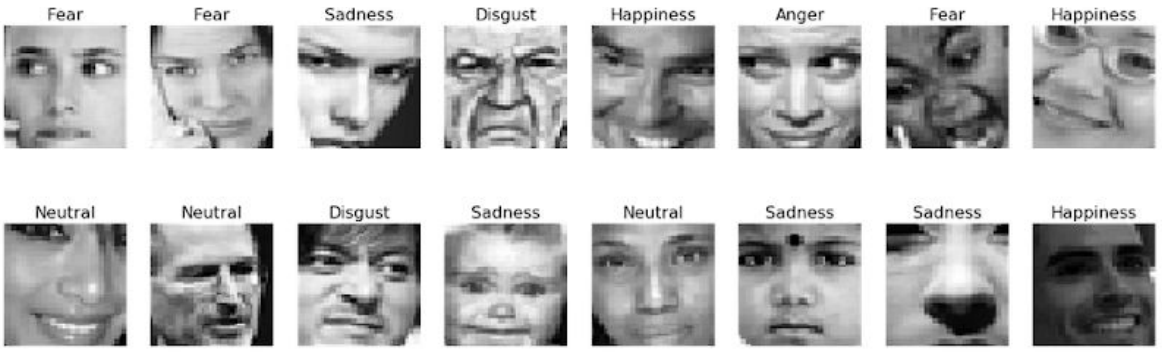


Figure 41: Batch of images after the "Cropping" preprocessing.

4.4. Light Intensity Normalization

The contrast and brightness of a picture can vary in images of the same person, even mimicrying the same facial expression, which implies bigger variations between descriptors [30]. Thus, to address this problem a method adapted from a bio-inspired technique has been applied in order to reduce and normalise the intensity of the light in images [30]. It is called "Contrastive equalization" and it consists in two phases: initially, the value of every pixel is extracted from a Gaussian-weighted average of the pixels neighbours. Then, every pixel is divided by the standard deviation of these neighbours. From Figure 42 it can be notice that not only the facial components are highlighted but also the personal accessories of the subject.

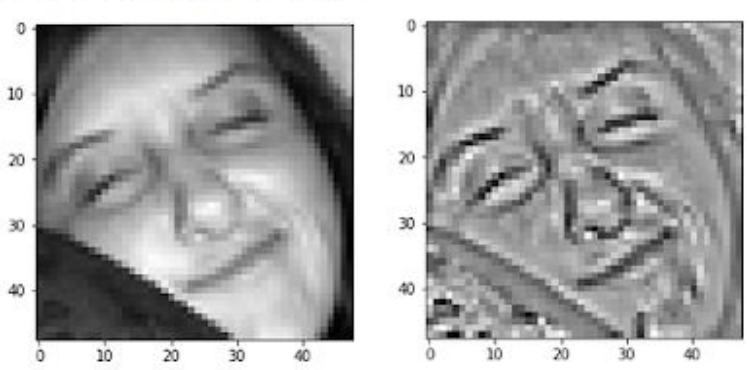


Figure 42: Original and final image, resulting from the "Light Intensity Normalization" preprocessing.

Figure 43 shows that the test accuracy is highly similar to the accuracy obtained by the original model, without producing any improvements. Additionally, Figure 44 illustrates that this time the class with highest accuracy is the Happiness with 76.13% followed by the Surprise with 66.73%, reaching its lower among the Original, Alignment or Cropping model. Other emotions,

slightly vary comparing to previous experiments. Finally, Figure 45 shows a tiny example of the images which will be received by the Convolutional Neural Network, in which it could be appreciated other components apart from the facial macro-expression.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.2190	92.22%	2.2287	56.28%

Figure 43: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Test Accuracy (%)
Anger	41.94
Disgust	64.44
Fear	42.02
Happiness	76.13
Sadness	44.21
Surprise	66.73
Neutral	51.84

Figure 44: Individual accuracies per class, being Happiness the class with highest recognition rate.

Figure 45: Batch of images after the "Light Intensity Normalization" preprocessing.

4.5. Not-face-detected Removed

The whole dataset is formed by 35,888 images, of which 28,709 are meant to train the model and 7,178 to test and evaluate it. Consequently, 8,587 images belonging to the training set (29.91%) and 1,093 images pertaining to the test set (30.45%) have been identified as images in which the face is not detected by the official OpenCV library. This means the subject may be in a very complex angle, position, proximity or even there is not subject at all. It might be

thought that getting rid of this kind of images could influence positively, not only in *online prediction* but also in *offline training*. Hence, the best previous preprocessing technique (Alignment) will be combined with others (Crop and Light Intensity Normalization) using a database without images in which the face has not been detected.

4.5.1. Alignment

Indeed learning from images filtered by OpenCV slightly improves the performance of the system which learns from the original database. Figure 46 shows the current accuracy, going from 93.16% to 97.01% in the train set and from 58.34% to 59.49% in the test set respect the original "Alignment" system. Figure 47 maintains the best and worst class performances, without highlighting any significant difference.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.0907	97.0132	2.7121	59.4951

Figure 46: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Test Accuracy (%)
Anger	45.94
Disgust	51.02
Fear	39.05
Happiness	79.22
Sadness	44.36
Surprise	78.47
Neutral	53.51

Figure 47: Individual accuracies per class over the test set.

4.5.2. Alignment + Crop

Observing Figure 48, it could be appreciated that aligning and cropping afterwards the original image is favorable, as the macro-expression becomes the main focus of the image. Figure 49 supports the theory of discarding faces in which the facial region has not been detected. It can be noticed that cropping an image of this kind, is not only unfavorable, but also, it loses all the information related to the emotion. Therefore, after getting rid of these trouble images, and cropping after aligning, the 4.5% and 3.4% accuracy increment respect the train and test sets of the simple Alignment model could be expected.

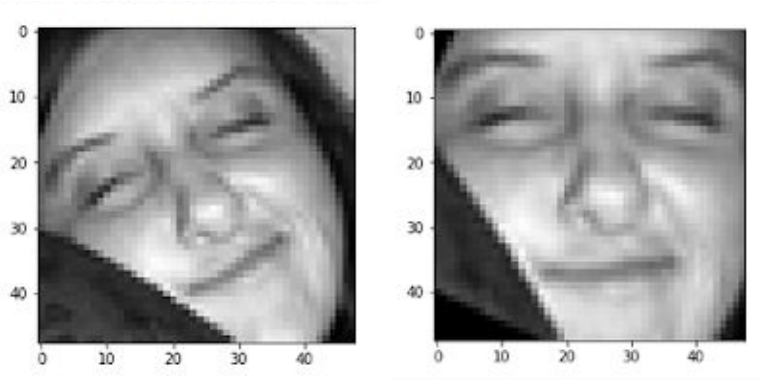


Figure 48: Original and final image, resulting from the "Alignment and Crop" preprocessing, when it is favorable.

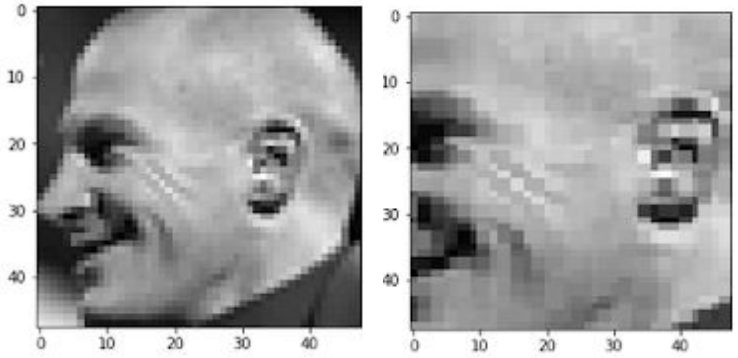


Figure 49: Original and final image, resulting from the "Alignment and Crop" preprocessing, when it is not favorable.

Figure 50 shows the accuracy obtained by this model, being 62.89% the highest accuracy over unseen images respect previous models. Additionally, Figure 51 shows that the behaviour of this model, in spite of being is similar to others (situating Disgust, Happiness and Surprise as the classes with the best accuracy), all of the emotions individually improve, being Sadness the only emotion under 50%, which advance the bright behaviour of others.

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.1149	96.08%	2.3668	62.89%

Figure 50: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Train(%)	Test(%)
Anger	94.79	50.00
Disgust	96.55	71.79
Fear	96.14	51.89
Happiness	98.76	81.50
Sadness	91.83	42.51

Surprise	97.71	75.25
Neutral	95.11	59.09

Figure 51: Individual accuracies per class, belonging to the dataset in which the images with not-face-detected have been removed.

Figure 42: Batch of images after the "Alignment+Crop" and "Not-face-recognized" preprocessing.

4.5.3. Alignment + Crop + Light Intensity Normalization

Strangely, Figure 54 shows that the performance of this models does not overcome the Alignment+Crop model. It seems that the light intensity normalization is not favorable for this kind of dataset. The research performed in the paper [30] proved that this "Contrastive equalization" was a clear indicator of the subjects facial expression, extraordinarily improving their results. Maybe, looking at Figure 54 and comparing it with Figure 53 or any image of Figure 57, it could be notice that in the FER2013 database, more subject's components are highlighted along with the macro-expression, which could be the negative influencer factor.

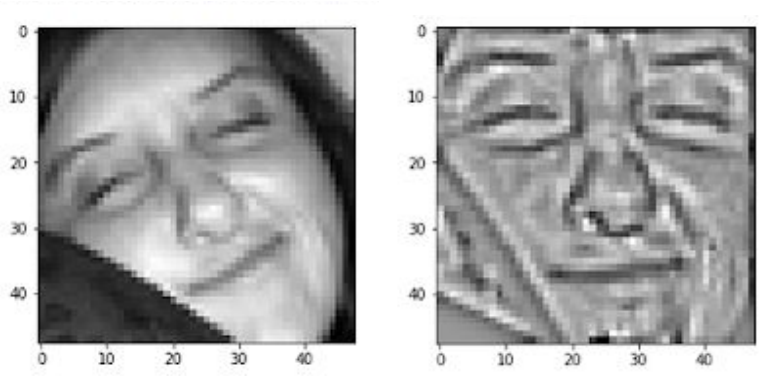


Figure 53: Original and final image, resulting from the "Alignment, Crop and Light Intensity" preprocessing.

From Figure 54, it can be appreciated that not only the quality of the image is better, but also the cropped area does not present external accessories, facilitating a better emphasis of the macro-expression.

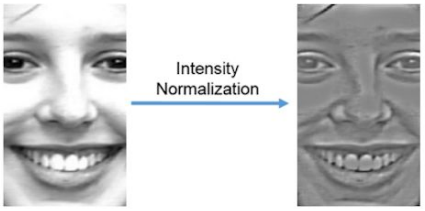


Figure 54: Original and final image, resulting from the "Contrastive equalization" process of the CK+ database [30].

Train Loss	Train Accuracy	Test Loss	Test Accuracy
0.3126	88.99	1.8416	57.29%

Figure 55: Accuracy regarding the training and test set respect to the 7 possible classes.

Macro-expression	Test Accuracy (%)
Anger	40.96
Disgust	41.17
Fear	38.79
Happiness	78.67
Sadness	41.63
Surprise	72.63
Neutral	50.93

Figure 56: Individual accuracy per class, keeping Happiness and Fear the classes with highest and lowest recognition rate.

Figure 57: Batch of images after the "Alignment, Crop and Light Intensity Normalization" preprocessing.

5. Overview of the Results

From Figure 58, it can be seen that already the Original system performs over the standards, acquiring 56.42% accuracy. Then, studying the experiments independently, it was discovered that the only preprocessing technique which works better than the original model is the Alignment approach, surpassing it by almost 2%. It is though that the Light Intensity Normalization model still remains the Original's model accuracy because of the wildness of the dataset, as not only the facial expression was remarked, but also external subject's accessories. In addition, the Crop model did not work up because of the proximities, positions or natural rotations of the faces, creating sometimes images in which the emotion was not longer visible even for a human eye. Therefore, the Alignment model was the chosen to perform three extra experiments. Consequently, it was noticed that the database contained extremely wild images not even recognized by the CUDA library (once because of the complexity of the picture and others because there was not subject at all), which could not favor the final performance of the system. Thus, the second round discarded any of these images for both, training and testing the model. Just by removing them, the accuracy improved 1.15% over the Alignment model. Then, after running the Alignment + Crop and the Alignment + Crop + LIN, it was found out that the LIN preprocessing affected negatively to this dataset, and on the contrary, cropping after alignment affected notoriously to the performance of the model, reaching the maximum accuracy of almost 63%, which is quite impressive given the number of possible predictions and the irregularity of the database.

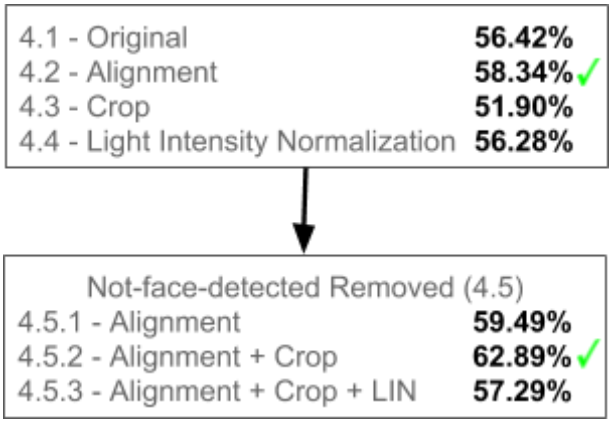


Figure 58: Summary of all the accuracies obtained for every one of the relevant experiments performed.

Additionally, studying the 37% of images wrongly classified by the Alignment + Crop model, it might be perceived that there are uncontrolled characteristics which are not related to the model, but to the database. Figure 59 illustrates a batch of 12 images picked randomly over the images not correctly classified by this particular model. Above every image there are two labels, the name at the left indicates the original label and the name at the right the label predicted by the model. Apparently, there are some images which its original label is not correct, or in which the emotion presents uncertainty. Over this circumstances the model

cannot do more than cut out these incongruencies in the database, as if not, the CNN will learn what it is taught [8].

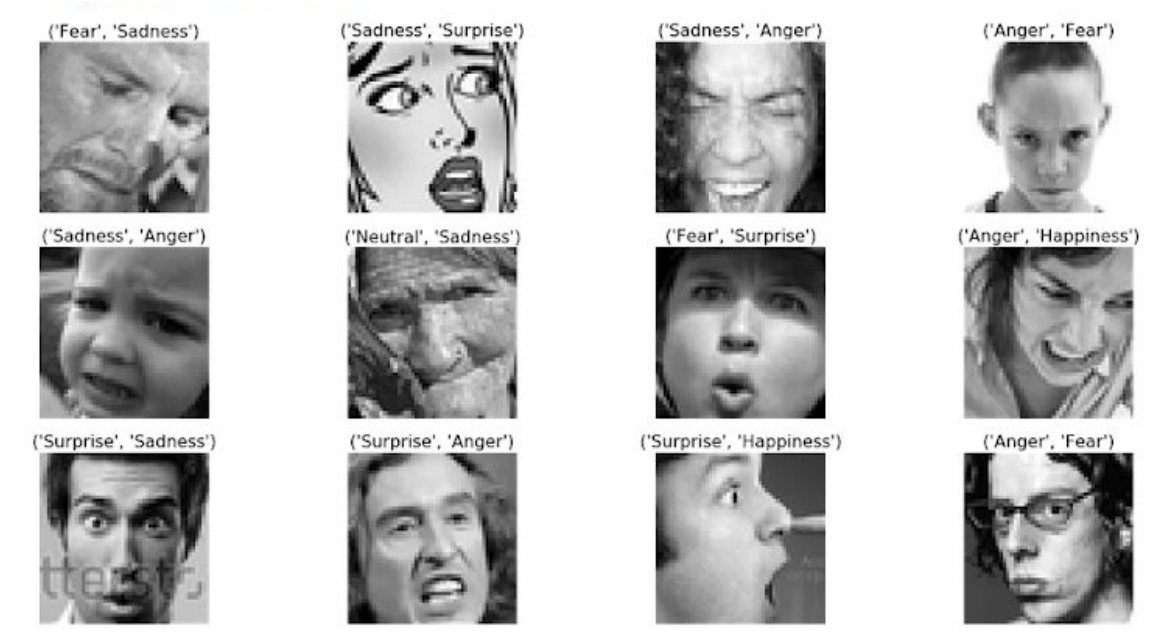


Figure 59: Batch of 12 images chosen randomly over the images incorrectly classified by the "Alignment + Crop" model. The label at the left is the original, whereas the label at the right is the predicted by the system.

In order to make a proper comparison among systems, they should share not only the same CNN model (with different parameters), but also the same prediction classes. Thus, the system obtained by the researchers at the Universidade Federal do Espírito Santo [30] is the chosen External Model to compare with. It is important to mention that both systems perform similar preprocessing techniques before sending the images to the CNN with the particularity that the External Model uses a controlled database and This Model does not. From Figure 60, it can be seen that this model behaves better than the other (External model) with no previous image preprocessing. In addition, the Alignment improvement is quite similar between them. The Crop experiment, produces a huge improvement over the External model and a noticeable decrease over This model. The Light Intensity Normalization (LIN) technique slightly overcomes the accuracy respect the Original approach in the External model, at the same time this model does not observe any favorable change, being the accuracy among models very similar. Then, even the External model obtains its best performances by mixing preprocessing techniques, it is significantly perceived that both model achieves their maximum accuracy by combining the Alignment and Crop methods.

Preprocessing	External model [30] - Accuracy	This model - Accuracy
Original	53.57%	56.42%
Alignment	61.55%	58.34%
Crop	71.67%	51.90%

Light Intensity Normalization (LIN)	57.00%	56.28%
Alignment + Crop	**87.86%**	**62.89%**
Alignment + Crop + LIN	86.67%	57.29%

Figure 60: Comparison of the performance of the model developed by the researchers of the Universidade Federal Do Espírito Santo [30] and this model.

Other approaches for facial emotion recognition have also focused on uncontrolled environment and spontaneous expressions, using different conventional and non-conventional models [38, 39] and concluding that dealing with this kind of images is still a challenging matter.

6. Conclusions and Future Work

As it has been seen throughout this project, the main goal resided in building an effective method for facial emotion recognition (FER) that operates in real time. It is important to highlight that the trained model is meant to work in real situations, for which the FER2013 database has been used, as it contains all type of wild and spontaneous images. Also, the model defined follows an non-conventional approach called Convolution Neural Network (CNN). Subsequently, a set of preprocessing operations for face normalization has been taken into account in order to decrease the need of controlled environments. Additionally, a deep study of the effects of these image preprocessing techniques has been performed, obtaining that the best approach aligns the subject's face, cropping it afterwards until keeping just the region from the eyebrows to the mouth, trying to remove irrelevant information like hair, earrings, etc, achieving almost 63% accuracy, which is quite remarkable considering the irregularity of the database and the number of possible macro-expressions.

Nevertheless, there is a huge amount of elements to improve and functionality to add, if more time is available. First, the macro-expression detection accuracy could be improved by researching spatio temporal techniques, which could cope with videos capturing the whole sequence of the macro-expression, learning from frame to frame instead from a single image. Additionally a new complex database could be created, not only to improve the accuracy of the already existent model, but also to unravel the field of micro-expression detection. As the number of micro-expressions is unknown, the first step would be to define combinations of relevant Facial Action Codifying Systems (FACS) in order to detect outstanding situations such as, differentiating between truth and lies, proud and repentance, peaceful and dangerous intentions and so on. Finally, a graphic interface could increase the use of this system in real daily life situations, approaching this project to people with no computer background, making possible a wider diffusion.

Definitions, Acronyms & Abbreviations

- GIGO: In computer science, “Garbage in, garbage out” means that flawed, or nonsense input data will produce nonsense output or "garbage" [8].
- FACS: Facial Action Coding System defines a system that tries to imitate a human observer to detect subtle changes in facial features [9]. It is the most widely used in the behavioural sciences [14].
- AU: Facial Action Unit, comprehends the individual or group of muscles that take part when producing a facial expression equivalent a particular emotion [6].
- SIFT: Scale Invariant Features Transform descriptor, takes an image and transforms it into a large collection of local feature vectors. Each of these feature vectors is invariant to any scaling, rotation or translation of the image [17].
- TIM: Temporal Interpolation Model, image descriptor created by Tomas Pfister, Xiaobai Li, Guoying Zhao and Matti Pietikäinen, machine vision group of the department of Computer Science and Engineering at the University of Oulu, Finland [13].
- SVM: Support Vector machine, are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis [18].
- MKL: Multiple Kernel Learning refers to a set of machine learning methods that use a predefined set of kernels and learn an optimal linear or non-linear combination of kernels as part of the algorithm [19].
- RF: Random Forest defines a set of learning methods for classification, regression and others, that operates by constructing a multitude of decision trees at training time. It corrects the decision trees habit of overfitting the training set [20].
- K-means: classifier that aims to partition n observations into k clusters, in which each observation belongs to the cluster with nearest mean.
- DBN: Deep Belief Network, is a generative graphical model, or alternatively a class of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer [21].
- CNN: Convolutional Neural Network, feedforward artificial neural network that has successfully been applied to analyzing visual imagery.
- RNN: Recursive Neural Network.
- LSTM: Long Short Term Memory, a type of RNN capable of learning long-term dependencies [28].

References

1. Prochazkova, E. and Kret, M.E., 2017. Connecting minds and sharing emotions through mimicry: A neurocognitive model of emotional contagion. *Neuroscience & Biobehavioral Reviews*, *80*, pp.99-114.
2. Sciutti, A. and Sandini, G., 2017. Interacting With Robots to Investigate the Bases of Social Interaction. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(12), pp.2295-2304.
3. Ekman, P., 1978. Facial Expression 1.
4. Shreve, M., Godavarthy, S., Goldgof, D. and Sarkar, S., 2011, March. Macro-and micro-expression spotting in long videos using spatio-temporal strain. In Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on (pp. 51-56). IEEE.
5. Ekman, P. and Friesen, W.V., 1982. Felt, false, and miserable smiles. Journal of nonverbal behavior, *6*(4), pp.238-252.
6. Ko, B.C., 2018. A Brief Review of Facial Emotion Recognition Based on Visual Information. *sensors*, *18*(2), p.401.
7. Liu, P., Han, S., Meng, Z. and Tong, Y., 2014. Facial expression recognition via a boosted deep belief network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1805-1812).
8. Babbage, C., Garbage in, garbage out.
9. Kanade, T., Tian, Y. and Cohn, J.F., 2000, March. Comprehensive database for facial expression analysis. In *fg* (p. 46). IEEE.
10. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z. and Matthews, I., 2010, June. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (pp. 94-101). IEEE.
11. Alpaydın, E., 2010. Introduction to machine learning (ed.).
12. Mikolajczyk, K. and Schmid, C., 2005. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, *27*(10), pp.1615-1630.
13. Pfister, T., Li, X., Zhao, G. and Pietikäinen, M., 2011, November. Recognising spontaneous facial micro-expressions. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 1449-1456). IEEE.
14. Polikovsky, S., Kameda, Y. and Ohta, Y., 2009. Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor.
15. Bartlett, M.S., Littlewort, G., Frank, M.G., Lainscsek, C., Fasel, I.R. and Movellan, J.R., 2006. Automatic recognition of facial actions in spontaneous expressions. Journal of multimedia, 1(6), pp.22-35.
16. Jeff Sieracki and Stuart Feffe. 2017. It's all about the features. [ONLINE] Available at: https://reality.ai/it-is-all-about-the-features/. [Accessed 27 July 2018].
17. Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), pp.91-110.

18. Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297
19. Chen, L., Duan, L. and Xu, D., 2013, June. Event recognition in videos by learning from heterogeneous web sources. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on (pp. 2666-2673). IEEE.
20. Ho, T.K., 1995, August. Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on (Vol. 1, pp. 278-282). IEEE.
21. Hinton, G.E., Osindero, S. and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural computation, 18(7), pp.1527-1554.
22. Liu, P., Han, S., Meng, Z. and Tong, Y., 2014. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1805-1812).
23. Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015
24. Breuer, R. and Kimmel, R., 2017. A deep learning perspective on the origin of facial expressions. *arXiv preprint arXiv:1705.01842*.
25. Dahl, G.E., Sainath, T.N. and Hinton, G.E., 2013, May. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (pp. 8609-8613). IEEE.
26. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
27. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.
28. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2017. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), pp.2222-2232.
29. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H. and 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.
30. Lopes, A.T., de Aguiar, E., De Souza, A.F. and Oliveira-Santos, T., 2017. Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, *61*, pp.610-628.
31. Lyons, M.J., Budynek, J. and Akamatsu, S., 1999. Automatic classification of single facial images. *IEEE transactions on pattern analysis and machine intelligence*, *21*(12), pp.1357-1362.
32. Yin, L., Wei, X., Sun, Y., Wang, J. and Rosato, M.J., 2006, April. A 3D facial expression database for facial behavior research. In *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on* (pp. 211-216). IEEE.
33. Simard, P.Y., Steinkraus, D. and Platt, J.C., 2003, August. Best practices for convolutional neural networks applied to visual document analysis. In *null* (p. 958). IEEE.
34. Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R. and Pal, C., 2015, November. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (pp. 467-474). ACM.

35. Hasani, B. and Mahoor, M.H., 2017, July. Facial expression recognition using enhanced deep 3D convolutional neural networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on* (pp. 2278-2288). IEEE.
36. Jain, D.K., Zhang, Z. and Huang, K., 2017. Multi angle optimal pattern-based deep learning for automatic facial expression recognition. *Pattern Recognition Letters*.
37. Dhall, A., Goecke, R., Lucey, S. and Gedeon, T., 2011. Acted facial expressions in the wild database. *Australian National University*, *Canberra*, *Australia*, *Technical Report TR-CS-11*, *2*, p.1.
38. El Meguid, M.K.A. and Levine, M.D., 2014. Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers. *IEEE Transactions on Affective Computing*, *5*(2), pp.141-154.
39. Liu, M., Li, S., Shan, S. and Chen, X., 2015. Au-inspired deep networks for facial expression feature learning. *Neurocomputing*, *159*, pp.126-136.
40. (19th August 2018) *Amazon Alexa*, Available at: *https://en.wikipedia.org/wiki/Amazon_Alexa* (Accessed: 20th August 2018).
41. Li, X., Pfister, T., Huang, X., Zhao, G. and Pietikäinen, M., 2013, April. A spontaneous micro-expression database: Inducement, collection and baseline. In *Automatic face and gesture recognition (fg), 2013 10th ieee international conference and workshops on* (pp. 1-6). IEEE.

Appendix

Referring the creation of the system and its future use, there are several attached files, showed in the table below.

Index	File Name	Functionality
1	Macro-expressions_CNN_[FER2013]_training&evaluation.ipynb	Training and evaluation of the model
2	Macro-expressions_predictor.ipynb	Final system
3	cascade_frontalface_default.xml	OpenCV file to recognize the face in pictures
4	haarcascade_frontalface_default.xml	OpenCV file to recognize the face in videos
5	fer2013.csv	Database
6	shape_predictor_68_face_landmarks.dat	Trained models for the face_recognition python library
7	models	Folder containing all the trained models detailed in the Experiments section
8	images-to-predict	Folder containing images obtained from internet, used to check out the system

Attached files along its name, functionality and Index name to refer them in the section.

User Manual

The code has been created and tested on Jupyter Notebook (Python 3), therefore it needs to be opened in the same programming environment in order to visualize it. This document will not show the code, as most of the functions are meant to cope with images, get information about the performance of the model and plot those results in different outstanding ways. Nevertheless, the Python definition of the final model using Keras, is shown below given its relevance.

```
def CNN_model_definition(rows, columns, channel):

        """
        Definition and creation of the Convolutional Neural Network model. It gathers 3
        convolution layers, 3 subsampling layers and 2 fully connected layers.

        :param: rows, number of pixels rows of the image.
        :param: columns, number of pixels columns of the image.
        :return: CNN model.

        """
        # CNN structure
        model = Sequential()

        # 1st Convolution layer
        model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(rows, columns, channel)))
        model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))

        # 2nd Convolution layer
        model.add(Conv2D(64, (3, 3), activation='relu'))
        model.add(Conv2D(64, (3, 3), activation='relu'))
        model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

        # 3rd Convolution layer
        model.add(Conv2D(128, (3, 3), activation='relu'))
        model.add(Conv2D(128, (3, 3), activation='relu'))
        model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

        model.add(Flatten())

        # Fully Connected layers
        model.add(Dense(1024, activation='relu'))
        model.add(Dropout(0.2))
        model.add(Dense(1024, activation='relu'))
        model.add(Dropout(0.2))

        # Output layer
        model.add(Dense(7, activation='softmax'))

        return model
```

Additionally, all the models acquired from the experiments previously explained have been stored in `Index7`. In order to test one, the name of the model should be indicated in the variable `model_file` of the file `Index1`, along with the following parameters:

- `FLAG_alignment`
- `FLAG_light`
- `FLAG_not_detected_face`

A value of “YES” will indicate that the preprocessing technique (Alignment, which includes Crop, Light Intensity Normalization and discarding images in which a face has not been detected) will be enabled, “NO” will disable these preprocessings. It is important to mention that given the size of the train, test and private test sets, the time to modify these images before evaluating them could reach 30 minutes. Moreover, the `FLAG_training` variable will decide if the model is being trained or just evaluated, if this variable is equal to “YES”, the `model_file` will be overwritten and the waiting time could reach several hours.

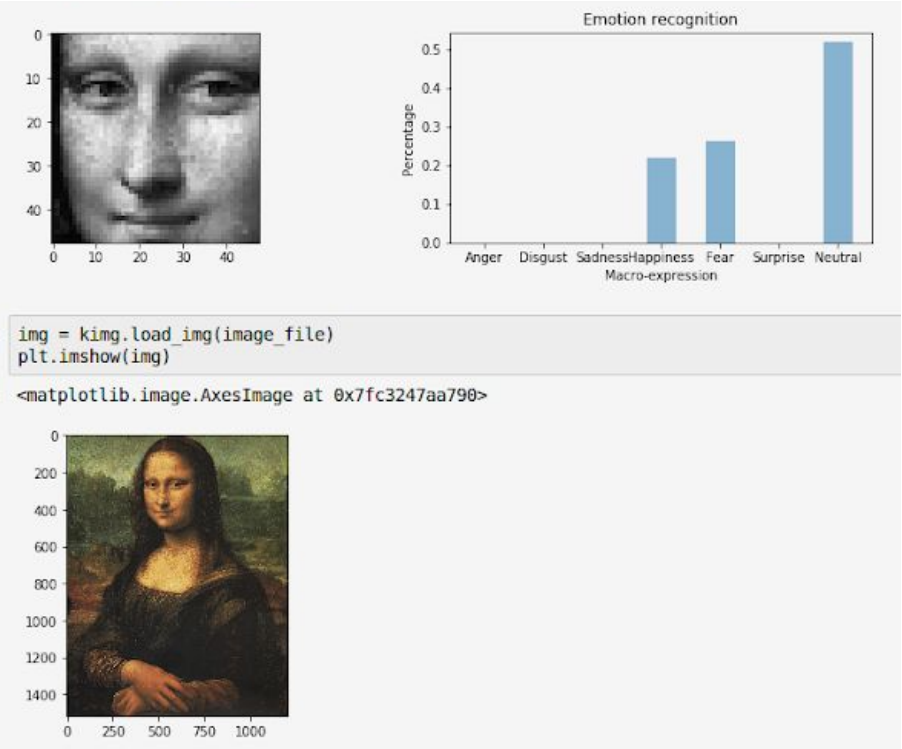
- `FLAG_training = NO`

On the other hand, to use the final system with both, personal images or with your own computer video camera, the `Index2` file will be used. It is convenient to store your image in `Index8` and change the name of the variable `image_file`.Given the Jupyter Notebook characteristics (allowing titles and text around the code), the indications are given in the same `Index2` file, so trying the system should be really easy going.

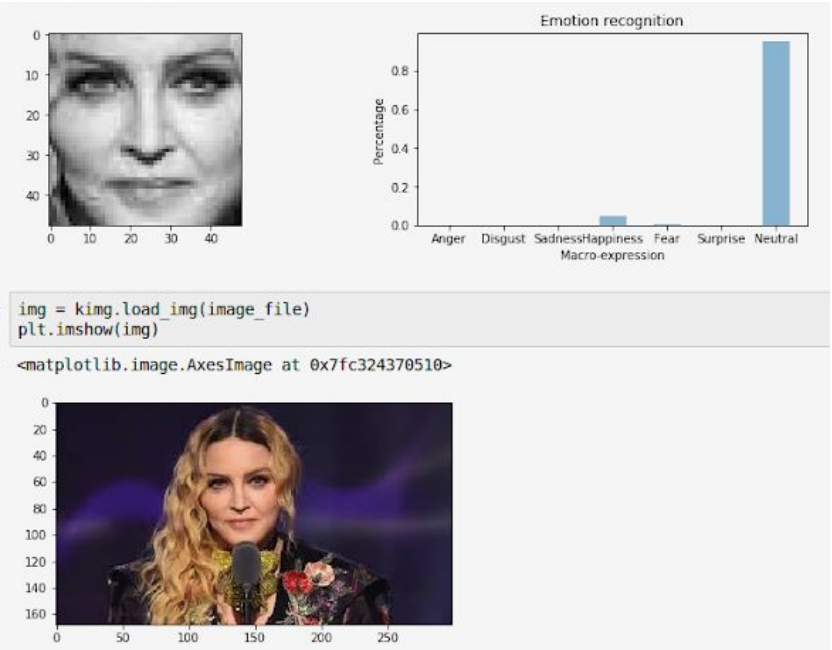
Prediction examples

Finally, the images of `Index8` have been tested by the system using the classifier with the best preprocessing technique (Alignment + Crop without using images in which a face was not detected) which achieved almost 63% accuracy. Throughout the images pictured below, it can be appreciated the cunning predictions obtained by the system, given the complexity and uncertainty of some of them.

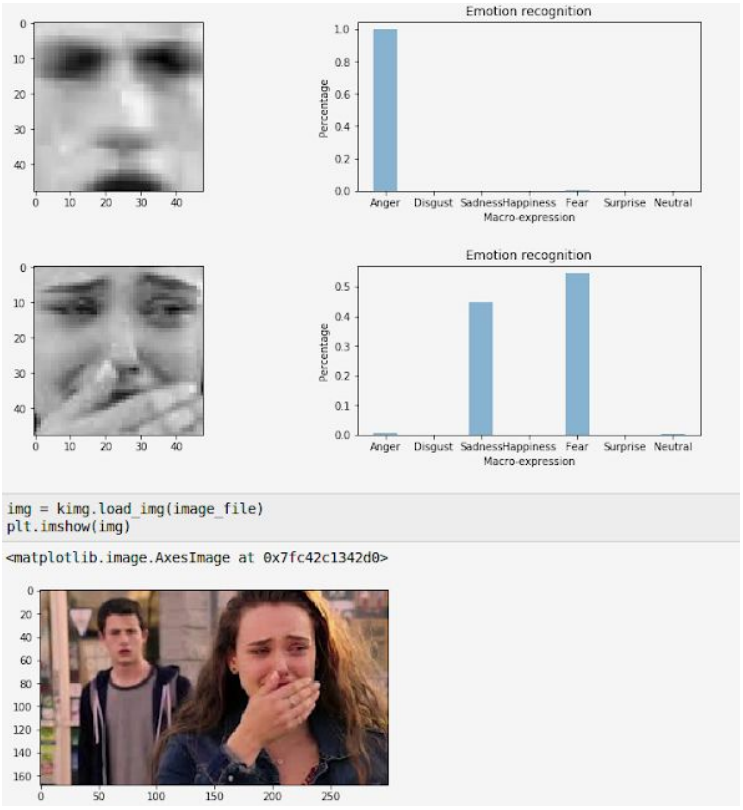
It can be seen that the system performs nicely identifying not only the main surprise emotion, but also glimpses of happiness.



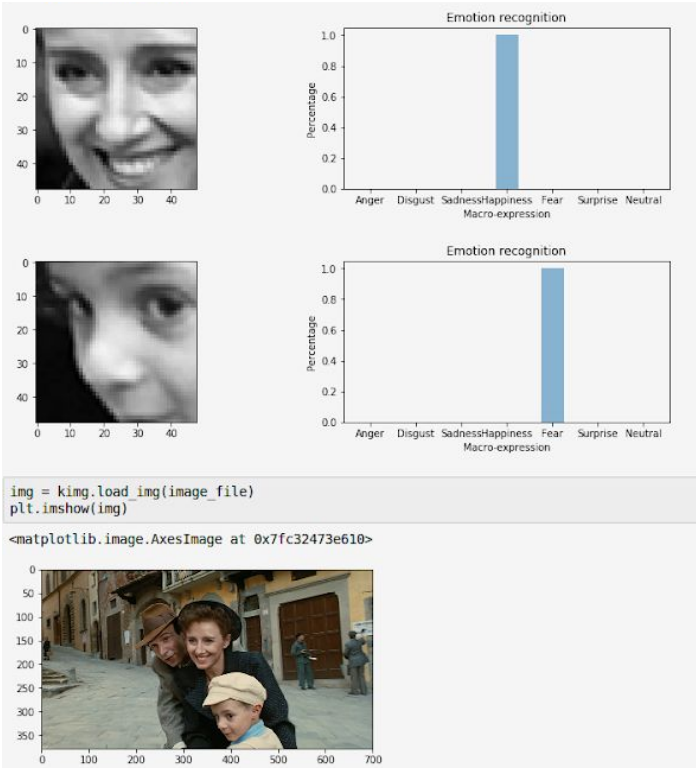
Mona Lisa (also known as La Gioconda) portrait made by Leonardo Da Vinci, which has always been highlighted by its mysterious smile. The system detects its neutral expression together some glimpses of happiness, but surprisingly, it also detects suppressed fear in her countenance. It is interesting to notice that, instead of looking at the original picture, observing the preprocessed grayscale image, that fear can also be perceived.



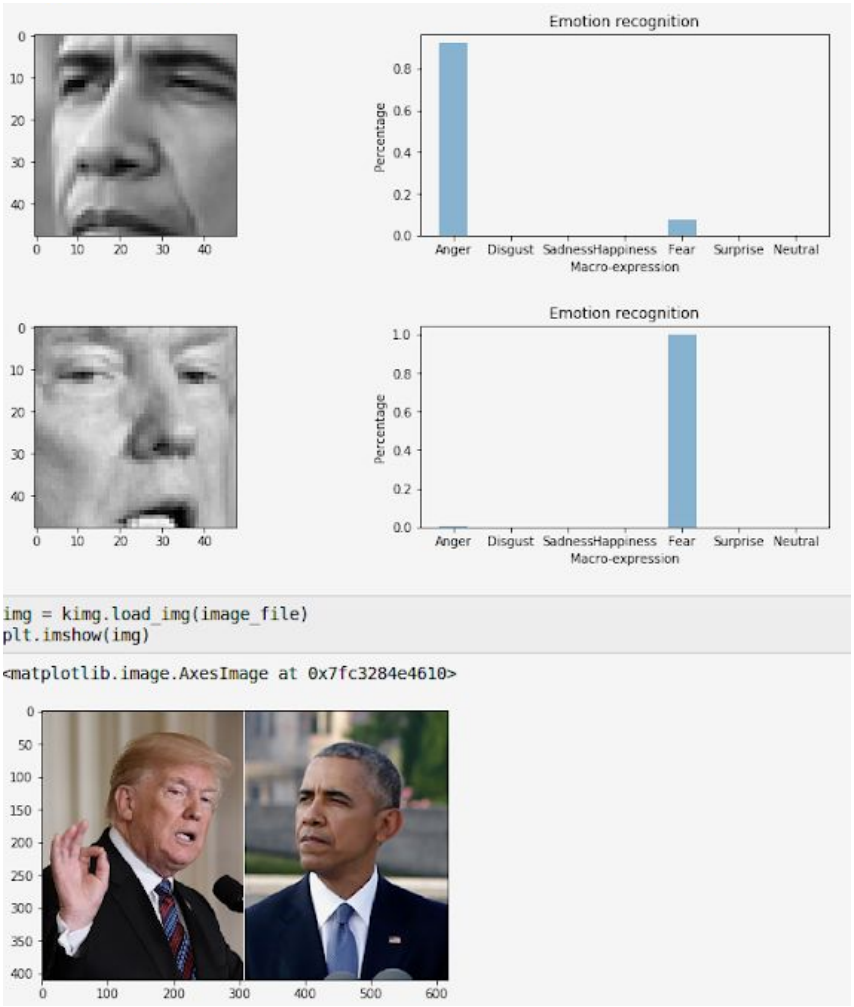
The neutral expression of Madonna when giving a heartfelt speech addressing sexism, misogyny, and feminism in the music industry in 2016. Although different from the Mona Lisa portrait, it is also recognized by the system, this time without perceiving any fear.



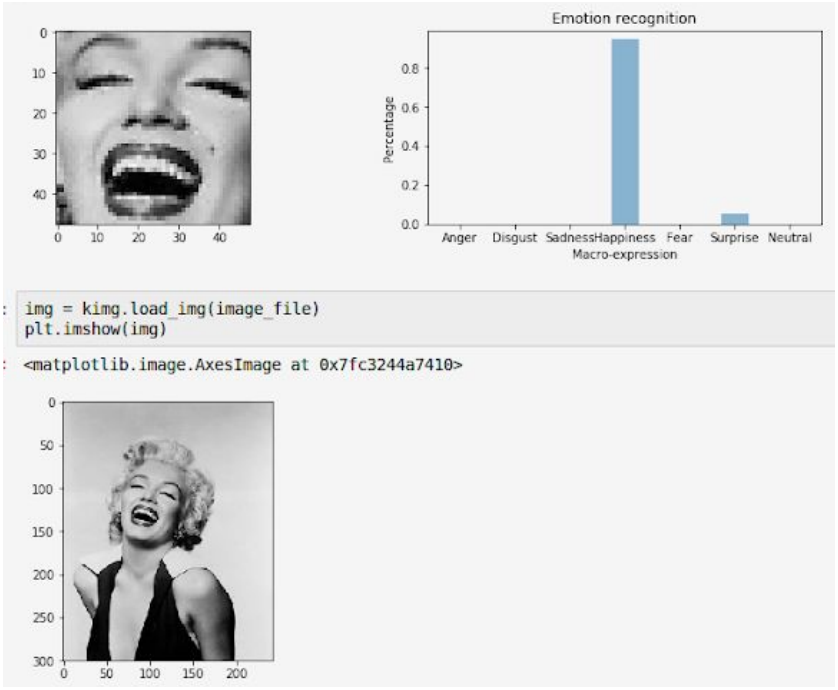
This image is extracted from a serie called “13 reasons why” where the girl in the picture commits suicide and records 13 voice tapes explaining the reasons. It can be seen how the system identifies her feelings before killing herself, when she is having an argument with a person very important to her.



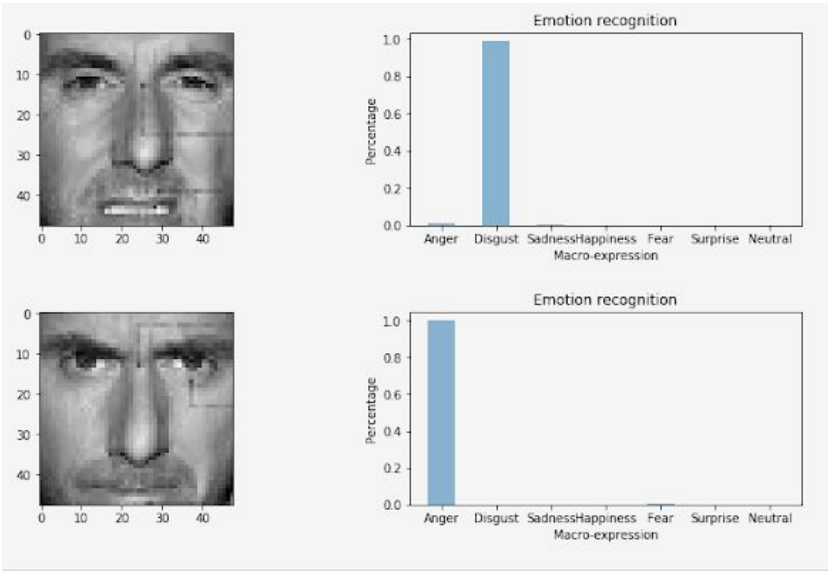
Additionally, this picture taken from “Life Is Beautiful” presents an example where one of the faces is not recognize. The other 2 are classified, maybe detecting fear in the boy’s face because they are riding a bike.



Subsequently, this is one real example of two presidents of United States, Donald Trump and Barack Obama, where their emotions are highly discrete.



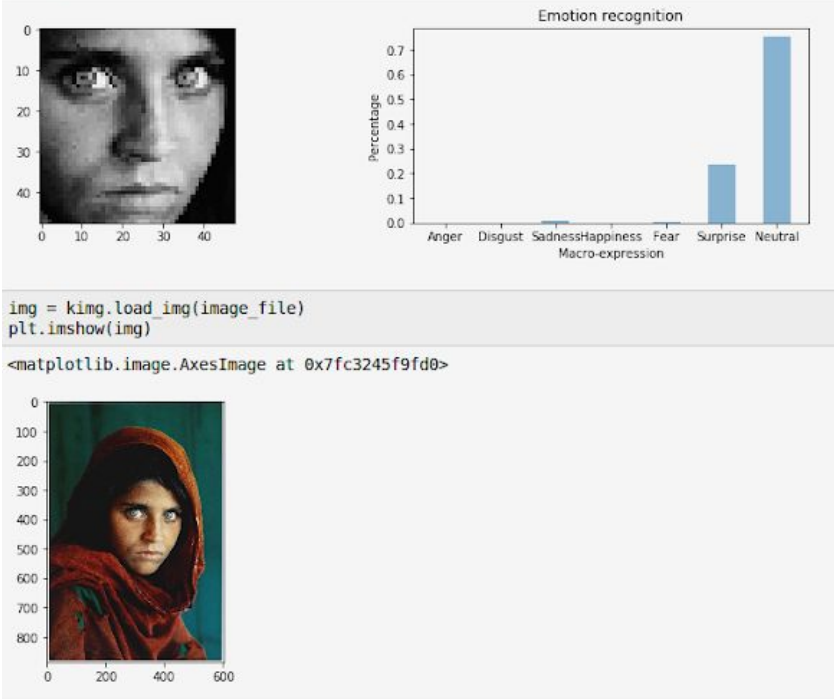
This picture of Marilyn Monroe clearly identify her feelings, as this is a very exaggerated actuation.




```
img = kimg.load_img(image_file)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fc31df7bcd0>
```

These two images are obtained from the serie “Lie to me” in which a company is in charge of detecting micro-expressions without computer intervenience. It can be seen that the system performs perfectly even when the expression is not obvious.




```
img = kimg.load_img(image_file)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fc3245f9fd0>
```


Picture taken by a National Geographic photographer to an Afghan girl called Sharbat Gula, the uncertainty of this picture makes difficult to know what emotion she is feeling even for a human eye.