Universitat Autònoma
de Barcelona

escola d'enginyeria

ANDROID™ REAL-TIME RECOGNITION SYSTEM OF JOY, DISGUST AND NEUTRAL EXPRESSIONS

In partial fulfillment for the degree of *Master in Multimedia Ambient Intelligence*
by Jordi Hernàndez i Prat and supervised by Dr. Enric Martí Gòdia

Bellaterra, October 2013

To my family, the dogs and the cats

Acknowledgments

I would like to express my gratitude to all the people, teachers, colleagues, friends or strangers, from whom I have had the chance to learn any thing, good or bad, and specially to my advisor Enric Martí, who has helped me to focus on what is really important during a research process.

I also want to express my love to all my family, for being always on my side.

Abstract

The recognition of expressions is a fundamental stage in the Emotion Recognition area, which at the same time, is a sort of Grial among those researchers who aim developing natural human interfaces. Unfortunately, a simple and fast recognition - in a natural way, not only on a lab environment - is today difficult: heavy algorithms and powerful work stations are still required.

This research aims to provide a first study of a new solution: a real-time expressions recognition system, enhanced by an image-processing and pattern- recognition algorithm, and which is ready to be implemented in millions of mobile or home devices. This solution can be a way of improving other existing systems, and also be part of future Human-Computer interfaces with among others, medical, research or entertainment purposes. The prototype Application, running on an Android™ *Smartphone* - which processes real-time images - is able to recognize disgust, joy and neutral human expressions.

Keywords: Expression recognition, Real-time, Android™, Natural Human Computer Interface, Emotion Detection, Face Detection, Image Processing, Computer Vision, Affective Computing

Contents

1 Introduction **1**
- 1.1 Problems to solve . . . 5
  - 1.1.1 The difficulties of emotion and expression recognition . . . 5
  - 1.1.2 Technology limits : Android capabilities . . . 6
- 1.2 Project main goals and Research Method . . . 7
  - 1.2.1 Previous research . . . 9
  - 1.2.2 System overview . . . 10

2 Implemented new technique of recognition **13**
- 2.1 Image capturing . . . 14
- 2.2 Image processing . . . 16
- 2.3 The recognition technique: The Lemniscate's trip . . . 19
  - 2.3.1 The way of recognizing expressions: joy, disgust and neutral . . . 19

3 Results **25**
- 3.1 Experiments and drawbacks . . . 26
  - 3.1.1 Disgust - non smiling - expressions . . . 28
  - 3.1.2 Neutral expressions . . . 31
  - 3.1.3 Joy - smiling - expressions . . . 33
- 3.2 The Mona Lisa Smile . . . 35

4 Conclusions and Future Work **37**

Chapter 1

Introduction

Nature flies from the infinite, for the infinite is unending or imperfect, and Nature ever seeks amend.

Aristotle, Greek philosopher and a scientist (384BC - 322BC)

The saying goes that emotions may unveil hidden chambers of an hypothetical human soul, just like those shadows - as it is said in Plato's tale - that some men tried to decipher from the inside of a dark cave. And just like them, from the huge area of affect machine-recognition, we are still facing the challenge to understand the blinks of a Reality that is far from being explained yet. This Master Thesis suggests a step in the darkness, focusing on the study of a simple shadow - the expressions - as a first step in the emotions recognition field. One said that a dove might think that flying would be better without the wind, while it is - the wind - that makes the flight possible. So that, being next to the beloved ignorance - and holding her brave hand - seems to be a nice chance to get out of the cave and, who knows, finally shining on the shadow that falls between the emotion and the response.

This Master Thesis project was originated by a personal interest in the research, depiction and understanding of human emotions; its manifestations through technological and artistic ways; and the possibility that computer machines can be able to recognize affects. In addition, this personal interest goes further, aiming to demonstrate that animals also share with humans a big part of this set of emotions - which comes from the same life situations, such as feed needs, defense or mating - but that promotes on them, slightly different physical reactions.

The human expressions are the core part of this project. Its meaning - as part of human emotions - has been an important study area since hundreds years ago. In nowadays,

this topic is near to Psychology or Neuroscience, but since long ago it has been part of many other disciplines. For instance, artists - in their aim to portrait the emotional dialog among human individuals - have revealed the concordance among physical reactions and the emotions which lay behind, as Charles Bell explained in 1872 [3]. It is important to take into account that an expression is much more than a static reaction, but also a part of an emotional dialog - not only verbal or physical. It means that individuals can adapt or modify expressions according to their environment's requirements. Darwin clearly emphasized on that when he researched about the expression's evolution in parallel with human kind evolution [4]. A three-stages updating of this dialog's skills is developed during human individuals' life: babies have a sort of innate basic map of reactions - some studies prove that this is commonly known among all the humans [6]. A set of expressions that individuals modify according of what they see - babies do it by imitating adults - and finally, depending on their social environment. This is part of human's need to successfully communicating with the rest, and mainly, to adapt the dialog to particular interests. This updating process is *semi-automatic* and *quasi-unconscious*, although sometimes the adaption of reactions is deliberate. In sum, a human emotion - in a social context - is a bunch of layers which need to be analysed:

- The particular individual sense of emotion - e.g. one is feeling sad according to some causes that not always make all others feel sad,
- The physical reaction - e.g skin's chemical variances, heart beating, head movement, body transpiration or, and mainly, expressions of the face,
- The social context where the individual is acting - several cultural aspects must be considered in order to decrypt emotions' specific nuances during a communication process,

This aspects drive us to a concept : the impression, very usual among those interested in painting movements, and mainly connected with French impressionists, who tried to frame all of this variables in order to capture an entire emotion - and they succeeded in it. As *human machines*, we can infer from a Degas portrait much more than the sketch of a situation. The combination of light, movement, colours, shapes and shades - placed in a social context, a dirty French bar, which even evokes much more - gives us all the necessary to go deeper and understand almost all about what the characters are feeling. Maybe, in the future, it will be possible to develop a computer machine which could take the pulse of the portrait in Figure 1.1, meanwhile, it is necessary to work harder on the analyse of expressions. At this moment, one could try to twist the parallelism with art movements and talk about the expressionists artists: Perhaps were the latter trying to capture just an expression? Yes and no. In fact, the expressionists painters were acting and creating as humans are used to do. By adapting some basic known codes - as physical expressions - they constructed a complete message about emotions and feelings.

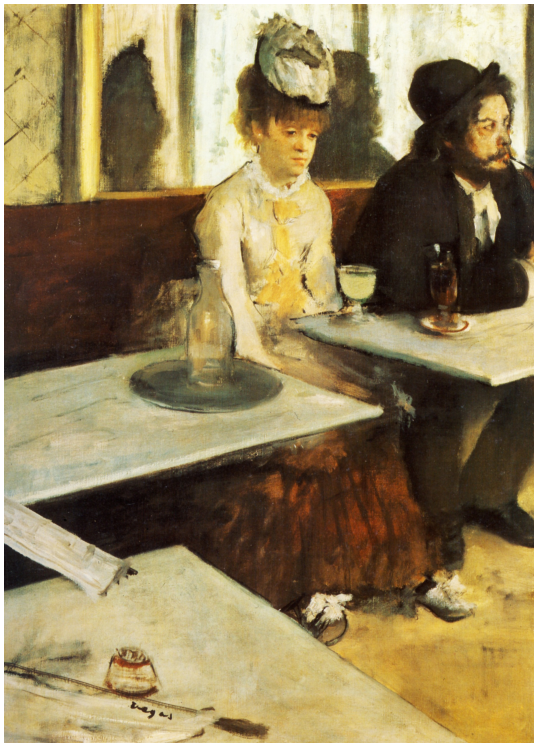


Figure 1.1: Edgar Degas, "The Absinthe Drinker", 1876

In Figure 1.2, Munch depicts a similar situation that Degas in Figure 1.1. But in Munch's masterpiece, it is easier to identify the typical, simple and effective traits of a disgusted expression. The artist has strongly remarked it on the sad face of the woman. But this is not just a naive portrait, which is using the same rudiments commonly known by the child painters. According to the explained above, Munch is offering us a complete message where emotions, context and answers, are powerfully mixed and overlapped. In fact, Degas and Munch are exposing almost exactly the same situation and the same kind of emotions, but in almost opposite formal ways. And however, the *Human Machine* is interpreting both portraits rightly, and extracting the same kind of information about the woman and man depicted on the painting. Paul Ekman [5] also remarked this disambiguation among expression and emotion terms, when he wrote that an *Expression is a central feature of emotion, not simply an outer manifestation of an internal phenomena*.
General Usage Computational devices - for the time being - are only able to extract information in a more simple or naive manner. Somehow, just like children do when they start learning drawing and paintings. Fortunately, several state of the art researches have been developed on laboratory computers, as the extraordinary case of Hoque et al.[9], who succeeded in demonstrating than computers - provided with specific algorithms, such Google™-NevenVision™, and huge databases - can go even further than common human perception. In fact, they demonstrate that a machine is able to infer richer information from voice and images than a human is. Just this last statement pushes the reader to a ocean of possibilities but unfortunately, this project's purposes are short-term oriented.

Figure 1.2: Edvard Munch "Ashes", 1894

As it has said, big computational features are still needed in order to capture and manage enough data to reach important results. This scenario implies waiting for several years until it is possible to embed this technology on mobile devices. And what if it is wanted to start now using devices such as glasses or watches, in order to understand some expressions - even emotions - from the users? And what about the sensors's data which is already available? Could be possible mixing all of this in order to get some valuable information?
The main goal of this research is to propose a real-time mobile solution - a combination of software and hardware - which can detect basic expressions such as joy - smile - , disgust or neutral. A solution easy to update at the same time that technology is developing. In brief, the aim here would be that a mobile device could be able to recognize the basic expressions which are drawn in figure 1.3.

Finally, as a conclusion of this introduction point, it is necessary to state that this research is far from being just an experiment or a *technological-artistic* hypothesis. The improvement of Human Computer Interfaces (HCI) can be an astonishing milestone in the development of medical treatments - e.g. those oriented to patients with communications difficulties, even when these drawbacks are critical. At the same time, recovering therapies can be designed according to the particular patients's needs; and thank to the monitoring home devices - e.g. smartphones - doctors could be able to gather more accurate data about their patients' actual health. But medical possibilities are just a small part of what stays beyond of a real-time emotions recognition by machines.

Figure 1.3: Natalia Marín "Gava i Rres", 2011

1.1 Problems to solve

This section points two key areas: the difficulty related with affect recognition, and the actual technological constraints.

1.1.1 The difficulties of emotion and expression recognition

Basically, these are the critical points of emotion and expression recognition by machines:

- The meaning of the traditional emotion prototypes is not clear enough - e.g. a smile not always means happiness. Usually, a smile also appears when someone is crying. More research work is necessary in this field,
- Laboratory working conditions and strong resources - as powerful computers - are still required. Is not still possible to research onto real-life environments,
- *Multimodal* systems are necessary to determine emotions - it is necessary to count with different sources of information about the invididuals' reactions,

The emotions understanding is not easy even for humans. Although it has been demonstrated that machines are sometimes smarter in this area [9], it is difficult to develop an application which can manage all the necessary data to recognize emotions. Besides, there are still dark spaces on this research field, per instance, Hoque et al. [9] found that a delighted or frustrated smile is not always related with human preexisting cliches - sometimes these are absolutely opposed. Somehow, remembering Darwin and what he exposes in his *Principle of Antithesis* [8].
The contradiction among existing cliches and real emotions expression, can be the result of the individuals' desire of camouflaging their real emotions. In addition, just one-mode data capturing system - e.g. camera/face, micro/voice - seems to be not enough. It is not neither for humans, who need more than glimpses for understanding other human's reactions - context references, voice's style or body gesture, are also necessary. Some studies,

as the survey of affect recognition methods by Zeng et al. [14], remark the need to combine multiple sources of information in order to polish the final results obtained by the machines. Besides, to improve the final data, it is important working out of the laboratory and sampling information directly from users' day life.
It is now possible to state - at the starting point of this research - that a mobile device, a bunch of in-motion sensors - as it is depicted on Figure 1.5 - is a good candidate to offer us *multimodal* data across humans' day life, although *mobile's brains* are still too slow to process it fast.
In sum, the development of one mobile interface can be useful to improve the affect recognition - as a way to get much more data on-real time, and outside from the laboratory conditions - although these devices are not powerful enough to manage the information provided by *multimodal* systems. So it is necessary a short-term solution that allows to easily work with mobiles, primarily, with the goal to gather real data and to enhance existing techniques, such as those who work with images databases - e.g *eigenfaces, fishfaces* - or other type of biometrics. The research here implemented has focused on developing a technique - compiled on an Android™ application - that can recognize when someone is smiling, or by contrast, one is showing a disgust or neutral expression.

1.1.2 Technology limits : Android capabilities

Android™, its development tool-set, and the worldwide community strongly focused on that[1], gives to the researcher an amazing environment to work. However, there are some problems to face:

- Real performance of Android's Application Programming Interfaces (APIs),
- The devices' technological limits, such as those of the smartphones,

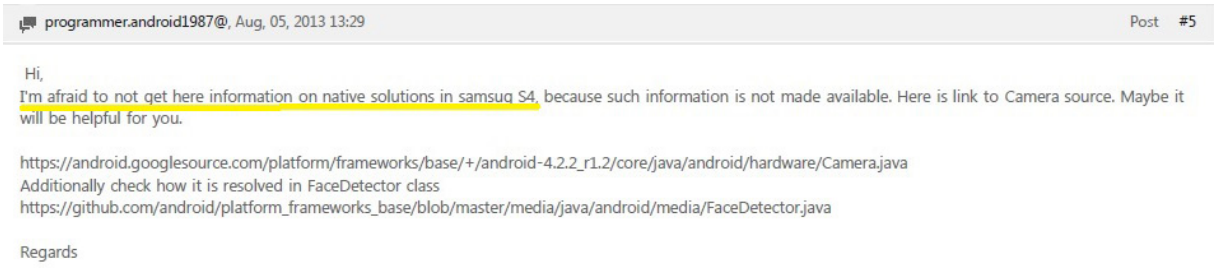
Android™ API 14 is featuring the *Camera.Face* class. A top programming solution which can work over real-time images - on *preview* mode - and which is able to give information about:

1. When a human face is on the camera's view,
2. The bounds of the face on a rectangle shape,
3. The coordinates of the right and left eye,
4. The coordinates of the center of the mouth,

This set of features gives the developer a wide range of options. Once the face is detected a *callback* is triggered, and after, it is possible to start working with face's rectangular bounds, but above all, the center of the mouth's coordinates offer the chance to explore related expressions, such as the smile. However, this top-of-range body of data is not fully operative. Per instance, Samsung™'s Smartphone model: S4, can not manage it rightly.

[1] Websites such as www.android.com or www.stackoverflow are a nice example

After asking about in *Samsung Developers Forum* - a corporative tool of Samsung - the *webmanagers* don't offer a clear answer, as it is shown on Figure 1.4. In addition, the native *Face Detection* option is not too much accurate. The device offers an approximation of the face position, but with significant area variances among consecutive samples. This can be an important problem when using the rectangle region as starting point to work is considered.


programmer.android1987@, Aug, 05, 2013 13:29 Post #5

Hi,
I'm afraid to not get here information on native solutions in samsuq S4, because such information is not made available. Here is link to Camera source. Maybe it will be helpful for you.

https://android.googlesource.com/platform/frameworks/base/+/android-4.2.2_r1.2/core/java/android/hardware/Camera.java
Additionally check how it is resolved in FaceDetector class
https://github.com/android/platform_frameworks_base/blob/master/media/java/android/media/FaceDetector.java

Regards

Figure 1.4: August 5, 2013 - Samsung Developers Forum answer

So that, it is also necessary solving the problems related with:

- A good approximation to determine the center of the mouth
- An efficient *workflow* sequence involving steps such as face detection process and data provided; camera preview; and particular aspects of the Samsung™ test device, such as its acquisition time limitations,
- A fast-recognition technique, suitable to work with the today and future Api14's features, and easily portable to more powerful devices [2],

1.2 Project main goals and Research Method

The main goals of this project - as it is explained in sections before - can be condensed in the statement:

> *To design and to develop a real-time expressions recognition technique and implementing it on a mobile application*

The next steps are necessary to accomplish the aim, and they are also part of the **Research Method** process:

- (a) Studying of existing techniques about recognition of face, emotions and expressions,
- (b) Studying the viability of adapting these solutions in a mobile device,

[2] Android technologies are probably going to improve fast during the next years. Somehow, guaranteeing an easy updating. Probably, next versions of the APIs will carry better performance routines

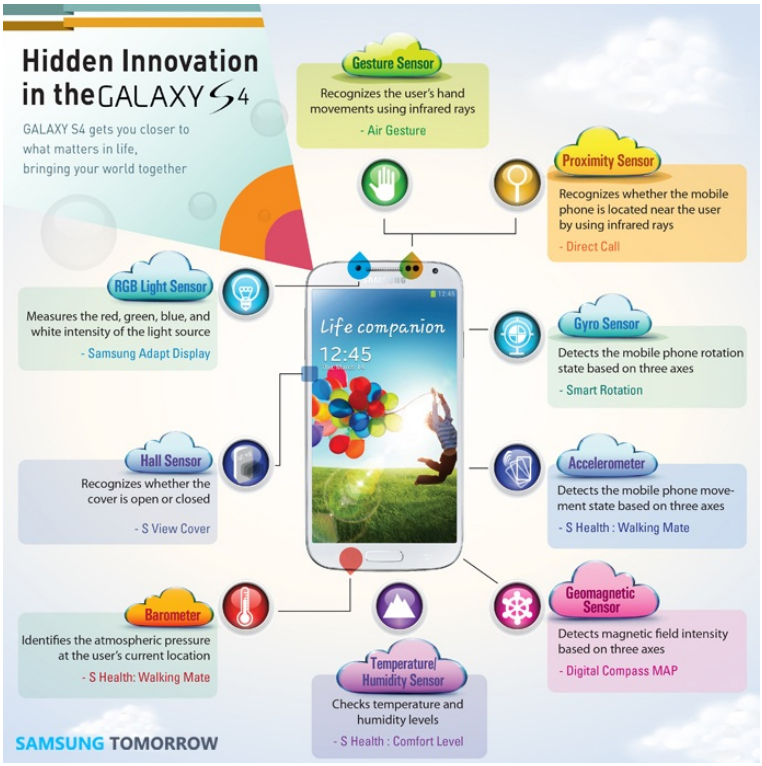


Figure 1.5: Samsung Galaxy S4 set of sensors

- (c) Developing of an image-processing technique which allows a real-time expressions recognition, focusing on joy, disgust and neutral expressions,
- (d) Implementing this technique in real market devices, such as Android™ Smartphones,
- (e) Testing the application's accuracy, concerning still and in-motion images,
- (f) Analysing how to enhance this technique and some possible future steps,

As starting point and according to the analyses of a part of the vast existing literature, it is necessary making a general and brief overview of the affect recognition problem - this is included in the introduction of this Chapter 1 (a). Secondly, as it is explained on Chapter 2 and Chapter 3, it is important to determine if Android™ devices are able to support a visual recognition application (b). To do that, some existing applications , and similar algorithms, have also been studied. After analysing them - e.g. OpenCV implementations [12] - it is time to develop a specific technique and its application oriented to Android devices (c). Some constraints are needed to be overcome at this moment, such as real technological capabilities of these devices which imply the need to define one as fast as possible technique (d) . All of this work, problems solved and remaining difficulties, is also explained on Chapter 2.
The algorithm must be tested in real world conditions (e). This is an interesting step collected on Chapter 3, that contains all the final results, the accuracy rates , as the

detected errors - e.g. some images' peculiarities which impede a final good recognition. The tests has been made over still and in-motion images. Finally (f), the study process ends with practical conclusions related with previous results, and with a brief exposition of possible next steps. All of this is not only focused on overcoming existing drawbacks related with the project but also related with the Affect Recognition field, as it is developed on Chapter 4.

1.2.1 Previous research

Really interesting workings have been developed about the Face Recognition problem, and also, about particular expressions and emotions recognition, per instance, those whose concern is the whole face, or just eyes, mouth or even just psychological reactions. OpenCV [12] - an image processing architecture - offers extraordinary opportunities in order to have a correct face-tracking, head's pose and movement analysis, eye tracking and face recognition. Google Inc. is leading a practical implementation of this techniques. Firstly, by addressing important research, like the works of Kim et al. [11] - who proposes a face tracker based on (HMM)Hidden Markov Models, and where the pose's features are obtained using the LDA (Linear Discriminant Analysis). Secondly, Google Inc., has acquired startups like *Neven Vision Inc.*, *DNNresearch Inc.*, or *PittPatt Inc.*, which allows Google™ to provide different powerful image-recognition technologies, such as included on *Picasa* [3] which, per instance, allows researchers to work with a group of face points, monitoring them, and extract valuable information during the whole face tracking process.

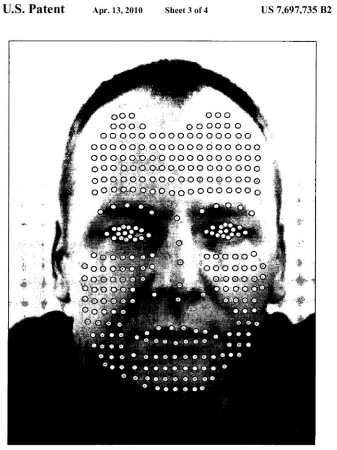


Figure 1.6: US Patent 7,697,735 B2 corresponding to the Image Based Multi-Biometric System invention by Neven et al.

Besides, Hoque et al. and Bailenson et al. use Neven Vision software package to specifically track faces during their studies - and focusing on particular points, as it is shown

[3] Wikipedia defines Picasa as an image organizer and image viewer for organizing and editing digital photos, plus an integrated photo-sharing website, originally created by a company named Lifescape (which at that time may have resided at Idealab) in 2002 and owned by Google since 2004.

on figure 1.6. Bailenson et al. have also developed a remarkable work on emotions detection over video images [2]. They deal with 53 face physical-features points and 15 psychological-features points, and they can correctly predict emotions on real people - not only from actors. To reach this goal they implement learning based algorithms, and introduce the measuring of some psychological reactions. An effective method which offers a lot of future possibilities to the multimodal emotion recognition systems.
Fortunately, developers have a lot of programming interfaces which make easier the implementation of applications oriented to Face Detection and Recognition. OpenCV is one of the top solutions to openly use, and that counts with a wide community of developers. Besides, other Java and C based solutions are available, such as JJIL (Jon's Java Imaging Library) created by Jon A.Webb - also an open source architecture - or Qualcomm™ Inc. solution *FastCV*, that runs on ARM-based processors, and is tuned for the Snapdragon™ chips included on latest Samsung™ devices - S2 versions and above. Remembering several goals of the project as the fast and simple processing, a solution easy to implement on as much mobile devices as possible, or a future updating, it is possible to conclude that anyone of the previous software architectures is suitable enough. However, at the time to make the final decision some other aspects are also relevant:

- The existing - and growing up - community of developers of Android™: it is amazing and highly remarkable the huge community of people who is actively testing Android™ and Java™ applications, its new features, versions and so on. In addition, the incredible generosity of anonymous developers offers everybody the chance to find snippets of code, testing them by yourself or interchanging questions and answers with hundreds of people. Android™ worldwide community is a gift for any researcher, student or anybody who likes programming.

- The strong presence of Google™, who rules part of this niche market, the research field and also the *Technology Planet*, is almost a guarantee of robustness of future Android™ APIs, the devices' compatibilities and much more.

Definitely, although a big variety - even more efficient - of architectures are available, the Android™ native SDK, supported by a gigantic community of developers and Google™, has been chosen in order to make this project's application possible. Chapter 2 includes more technical considerations about this decision.

1.2.2 System overview

A big sum of experiments about face and emotions recognition are based on learning processes, which usually work with image libraries. In order to categorize subsections of an image and its changes during a lapse time, some techniques on objects detection have been adapted and developed. Per instance, the HMM (Hidden Markov Models), adapted by Minyoung Kim et al. [11], or Haar Cascades - adapted by Viola and Jones [13]. The Haar-like technique is applied on OpenCV solutions as well. This systems are learning-based, and they need a database of classifiers to get more efficiency. The classifiers are updating as long as the system is learning. Bailenson et al. use Bayesian networks - in fact, HMM are

a simple type of this networks and also based on classifiers. At this moment, it is possible to remark important positives aspects and some constraints of this kind of solutions:

- The opportunity to implement learning systems: it allows to develop *auto-updating* applications which can acquire much better performance while users interact with them. Market applications, such as *Picasa*, illustrate this fact.
- It makes possible to combine physical features with psychological ones. This is a milestone in order to develop complete affect recognition systems.

However, this systems also involve some drawbacks:

- Both aspects above actually make implementations on mobile devices quite difficult,
- Such a big complexity is not fully necessary in order to develop simple human-computer interfaces,
- It is necessary a training time in order to setup it correctly,
- The need of an important database of classifiers can be a constraint. Per instance, it is necessary a database of eigenfaces - real images transformed by PCA (Principal Component Analysis) and related with face's variations or features. Figure 1.7 illustrates a group of these images obtained from the MIT website[4],

Figure 1.7: Part of the Rice University Eigenfaces Group @MIT website

[4] `http://fab.cba.mit.edu/classes/MIT/864.05/people/dgreensp/eigen/faces.html`

So that, in order to develop a tool easily adaptable to mobile devices, and whose performance - on real time - is suitable to develop human-computer interfaces - such one controlled by basic expressions -, it is necessary to make it simple. Maybe it sounds quite crude, but the fact of focusing on natural interfaces can be quite illustrative: if it is pretended that a human person is going to control an interface with just an expression - like a smile - it is basic that the machine is going to detect it as fast as possible. **The goal is to implement a system where just one change of expression - detected by a combination of two consecutive samples - is enough to obtain the control**, and this is about next chapter is dedicated to.

Chapter 2

Implemented new technique of recognition

The method here implemented is a technique to recognize three types of expression:

- Smiling - usually related with joy,
- Disgust - namely the inverted - U shape,
- Neutral,

This three-states system permits developing expressions-based interfaces, as well as start designing other applications to recognize emotions. The working diagram of the prototype is depicted on Figure 2.1.

The process three main steps are:

- 1. Image capturing
- 2. Image processing
- 3. Image analysis, which comprises the application of a new recognition technique

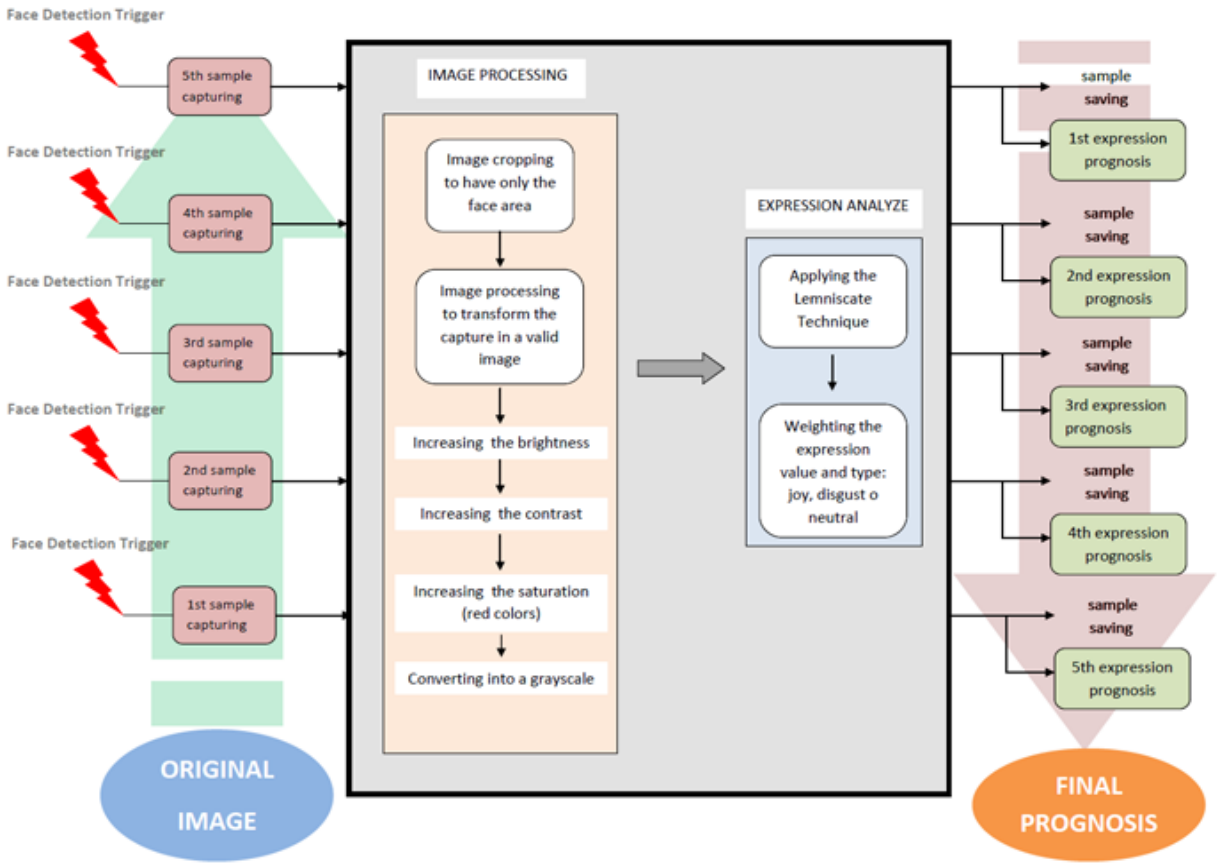


Figure 2.1: Block diagram of the process

2.1 Image capturing

The main goal of this step is capturing an image sample of a face, in order to further work with it. It is necessary to note that Android™ software includes two programming *classes* which provides *Face Detection* features [1]:

- FaceDetector.Face
- Camera.Face

The first one - implemented since the Api version one - is focused on work onto still images, such as bitmaps. But it is not prepared to deal with motion pictures. In addition, the

[1]The next software tools have been used during the programming process: Android Developers Tools, Build: v22.0.4-741630 which includes Eclipse. A Samsung Galaxy S4 smartphone, including Android 4.2.2, has been used to test the program versions.

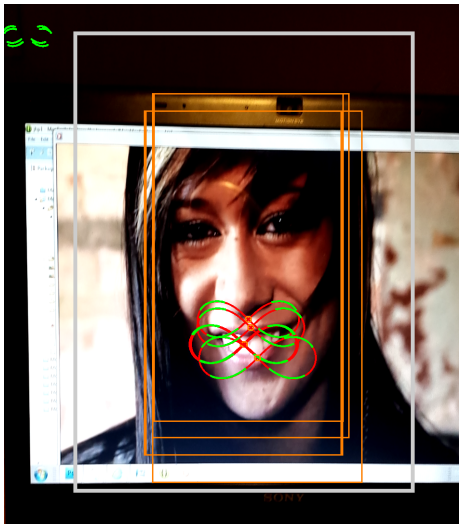


Figure 2.2: Application visual interface

faces' features provided are quite poor: just triggering on face detection, and providing the bounds of the face. As it is wanted to work with real-time moving images, it is necessary an enhanced class. Fortunately, *Camera.Face*, available since the Api14 version, allows to manage more interesting data [2]. Face's bounds, and the coordinates of the center of mouth, are valuable points to take into account during the next development.

Unfortunately, the practical research indicates that this features set is not still operative on Samsung S4 devices, as it is explained on section 1.1 - the frontal camera doesn't provide eyes and mouth coordinates. Probably, it will be solved soon, but as it is explained in next sections, by now it has been necessary to find an alternative way to determine the center of the mouth's coordinates. Another important point here is that the application visual design, and its interface, are not oriented to the final user. As it is shown on Figure 2.2, the interface consists on a simple camera-preview environment, with just a triggering button. This is because what is tested here is a recognition technique, and also its conditions to be implemented on a basic interface. Somehow, the application is only the window to check the technique features, a starting point to final-user oriented applications, but not something to put this afternoon on GooglePlay™.

Once the user has pushed the trigger button - just a shooting to start analysing the images - the application starts its automatic behaviour. Firstly, just waiting for a face. Java™ environments - Android is a combination of different languages, such as Java, C++ or HTML - offers the *Callback* actions, an automatic response to different *stimuli*. In this case, the detection of a face - getting over a predefined accuracy threshold *score*, of 55 out 100 - is enough to *shot the callback*. After, it is time for the application to gather useful information: the face bounds - in a rectangle - and unfortunately, not much more. However, it is possible now to work with inferred measures as: rectangle's position, width and height.

[2] As it is stated in Android's main website `http://developer.android.com/reference/android/hardware/Camera.Face.html`

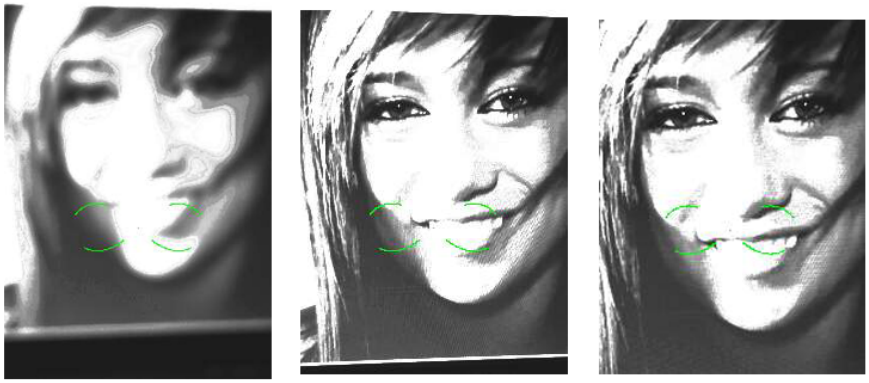


Figure 2.3: Consecutive but different captures of the same image

In sum, the applications is automatically able to complete the next image-capturing steps:

- Knowing when a face is on the preview image,
- Getting this face's limits in a rectangle area,
- Obtaining rectangle's measures according to the preview display,

Now, it is time to face the problem to analyse what is on the display. At least, it is known that there is a face there. The method here implies capturing visual information. Again, Android™ provides automatic routines - such as *TakePicture*, based also on callbacks' philosophy - that help to implement an automatic capture of the preview.

2.2 Image processing

As it is explained below, this step of the method partly remembers the eigenfaces creation process. The main goal now is to transform the captured image into a useful picture, which can be easily studied. In fact, what is pursued here is a sort of *lightweight-PCA transformation* but using only the basic tools which Android provides. The goal is to avoid complex mathematical operations which overload the processor and waste time. As a brief summary, it is possible to state that the application here is going to build its own eigenfaces, on real time, and according to a particular real face.

Once a face image is captured, cropping the image is necessary because just the face is important. To do that, the limits of the rectangle discussed above are now very useful information. To improve further processing time, is highly recommended to have a light-weight image file. Previously, the application has captured the picture with a quality of 60 out 100 of JPEG compression[3] and on a bitmap file. Besides, by cropping the useful data

[3]It means compressed using the JPEG compression algorithm

```
// setting values for every gamma channels
for(int i = 0; i < MAX_SIZE; ++i) {
    gammaR[i] = (int)Math.min(MAX_VALUE_INT,
            (int)((MAX_VALUE_DBL * Math.pow(i / MAX_VALUE_DBL, REVERSE / red)) + 0.5));
    gammaG[i] = (int)Math.min(MAX_VALUE_INT,
            (int)((MAX_VALUE_DBL * Math.pow(i / MAX_VALUE_DBL, REVERSE / green)) + 0.5));
    gammaB[i] = (int)Math.min(MAX_VALUE_INT,
            (int)((MAX_VALUE_DBL * Math.pow(i / MAX_VALUE_DBL, REVERSE / blue)) + 0.5));
}

// apply gamma table
for(int x = 0; x < width; ++x) {
    for(int y = 0; y < height; ++y) {
        // get pixel color
        pixel = src.getPixel(x, y);
        A = Color.alpha(pixel);
        // look up gamma
        R = gammaR[Color.red(pixel)];
        G = gammaG[Color.green(pixel)];
        B = gammaB[Color.blue(pixel)];
        // set new color to output bitmap
        bmOut.setPixel(x, y, Color.argb(A, R, G, B));
    }
```

Figure 2.4: Java language snippet of code which illustrates gamma-setting process

area, a better performance time is guaranteed. It is important to note that a good face detection is basic, because the next steps are closely related to this. Again, an Android's drawback needs to be overcome: the given rectangle area which contains the face is not always the same, even though always same video or picture data is analyzed - as Figure 2.3 shows. As it is depicted on this picture, the rectangles' areas are slightly different among consecutive samples - so that, the picture's cropped region is not exactly always the same. Sometimes, the differences - related with Android's native detection performance - are more accentuated. At this stage, it is necessary to solve a new drawback: the device is offering a sort of *corrupted data* - the mentioned wrong samples - which need to be correctly managed in next steps. However, these samples are providing also an interesting approach about where the face is located.

After cropping the image, and as it is shown on Figure 2.3, some other image settings have been changed in order to get a *private eigenface*. Specifically, the brightness, contrast and saturation color modification, as also a final conversion into grayscale. At this point, it is necessary to appreciate the useful contribution of Pete Houston and his Android examples of how manage all of this image features in an efficient way [4]. All these changes have been applied on image's pixels, and on every RGB(Red, Green and Blue) channel - the Alpha channel has remained unmodified during all the processing. Firstly, the brightness levels have been raised up to 90 out 100 on all RGB channels. Secondly, the contrast has been set up as Figure 2.4 shows: the gamma is modified, separately, on each RGB channel, by iteratively applying the equation 2.1 to all the bitmap pixels. Where variable **i** goes from $i = 0$ to $i = 255$ - with increments of 1 - , and *RGB* is a preset value - R, or G, or B considered separately - , depending on the corresponding channel. In this project, *(R,G,B)=(1,0.2,0.2)*. Finally, as it is also depicted on the code, the gamma value corresponding to each channel-

[4] Pete Houston's posts can be found on the site http://xjaphx.wordpress.com/

```
int index = 0;
// iteration through pixels
for(int y = 0; y < height; ++y) {
    for(int x = 0; x < width; ++x) {
        // get current index in 2D-matrix
        index = y * width + x;
        // convert to HSV
        Color.colorToHSV(pixels[index], HSV);
        // increase Saturation level
        HSV[1] *= level;
        HSV[1] = (float) Math.max(0.0, Math.min(HSV[1], 1.0));
        // take color back
        pixels[index] |= Color.HSVToColor(HSV);
    }
```

Figure 2.5: Java language snippet of code which illustrates saturation-setting process

```
// scan through every single pixel
for(int x = 0; x < width; ++x) {
    for(int y = 0; y < height; ++y) {
        // get one pixel color
        pixel = src.getPixel(x, y);
        // retrieve color of all channels
        A = Color.alpha(pixel);
        R = Color.red(pixel);
        G = Color.green(pixel);
        B = Color.blue(pixel);
        // take conversion up to one single value
        R = G = B = (int)(GS_RED * R + GS_GREEN * G + GS_BLUE * B);
        // set new pixel color to output bitmap
        bmOut.setPixel(x, y, Color.argb(A, R, G, B));
    }
}
```

Figure 2.6: Java language snippet of code which illustrates grayscale conversion process

color of the pixel is the result of selecting the minimum value among the previous equation result and 255. The constant value of MAX_VALUE_INT and MAX_VALUE_DBL, has been set up the RGB maximum level, that is 255.

$$x = 255i/255^{1/RGB} + 05 \tag{2.1}$$

Next steps implies working with the Saturation value of the image, which has been set up during a similar process than above, and where is important to note than the red colour levels have been incremented. Usually, the lips' tones are close of the red color, or have important amount of this. Therefore, rising the Red amount during the process, helps remarking the lips area - which is going to give really interesting information about the expression. Figure 2.5 shows the Java code of all the process. Again, an incremented iteration process is carried out through all the image's pixels. This time, the native function *colorToHSV* is used in order to convert HSV (Hue Saturation Value or Brightness) values to ARGB [5]. Pixels are multiplied by a level value - the desired amount of saturation -, in this case, *level=2*.

Finally, it is necessary transporting the resulting image to a *grayscale* version. As it is shown on figure 2.6, the RGB values have been multiplied by particular factors, in this case:

[5] As it is defined on www.android.com, colorToHSV converts HSV components to an ARGB color. Alpha set to 0xFF. hsv[0] is Hue [0 .. 360) hsv[1] is Saturation [0...1] hsv[2] is Value [0...1] If hsv values are out of range, they are pinned.

(R,G,B)=(0.299,0.587,0.114).
In sum, this is the image processing implementation, which is developed on the Java new classes tailored to this project: *imageCompressed, imageCropped*.

2.3 The recognition technique: The Lemniscate's trip

Until this stage, some problems have been solved and some drawbacks still need to be overcome. On one side, a *private EigenFace* is already available, but on the other hand, next problems need to be solved:

- The FaceDetection's results are different among the samples, even when the original data - picture or video - is the same
- How is possible recognizing expressions on the image processed?

A way for overcoming the first problem is by acquiring more samples. Finally, it is indicated that 5 consecutive images are used. It allows to apply the expression-recognition technique on 5 examples, and as more results are going to be gathered, these need to be classified according its accuracy. Per instance, a good sample of a smile, in case of being perfectly recognized - is weighted with a 1 - while a completely wrong recognition is not weighted. The intermediates - as possible smile or not-smile, are weighted with a 0.5.
This way of dealing with the data is partly is adopted because Android's today performance. With a better and constant face detection, the number of needed samples could be smaller - and closer to the model of expression-detector based on two-samples. Next section enfaces the core part of the project, a new proposal to recognize expressions.

2.3.1 The way of recognizing expressions: joy, disgust and neutral

This part is, probably, one of the more interesting of the project. Remembering the goals stated above, here is where the fast recognition is implemented. During this process, different sort of strategies are going to be applied: one, the analysis of the pixels, although it is known that having to work with pixels is not a very efficient processing - because of that, the mathematical transformations are usually applied and this fact changed, years ago, the old-fashioned methods. However, the proposal here is to study just only a particular line of few pixels, and mainly, on some *sensitive* regions.
To start working the coordinates of the center of the mouth are necessary. As it has been stated on chapters before, Samsung S4 devices are not providing this measures today. But at least, it is possible to know that the previously obtained rectangle includes a face. So that, and applying a geometrical approximation, it is possible to guess where the mouth's center is placed. As it is an approximation - including an error which is necessary to correct - it implies working with at least 5 samples. Maybe at the future, in case of a new S4 updating, it will be possible to work with more accurate coordinates on this device. At least, Android's future compatibilities may *ensure* that it will be possible on new devices, such as Google Glass™.

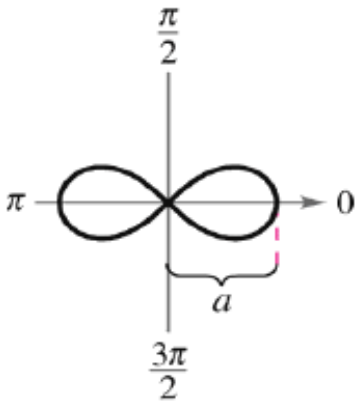


Figure 2.7: Lemniscate function representation

After the mouth's center is placed, it is necessary to establish the *a* value of the Lemniscate curve, since as it its depicted on Figure 2.7, this value is important in order to get the curve real size. Finally, the *a* value has been proportionally approached from the face's rectangle that the systems provides: *a= w/5*, where *w= rectangle's width* . Once the Lemniscate curve is placed, it is time to start analysing the image. As it has been explained above, the top methods architectures imply working with variances detection, and a big set of references to compare with. The strategy here implemented - and that is explained on this section - is quite different and a little bit unusual: *The Lemniscate's Trip*.

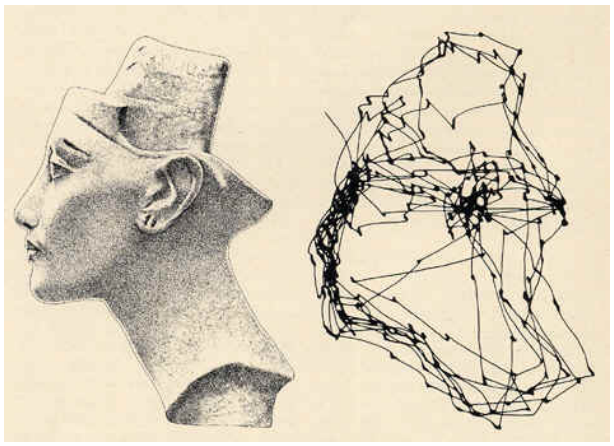


Figure 2.8: Image corresponding to Alfred Yarbus works about eyes movements. In this case, onto a picture of Queen Nefertiti's bust

The human gaze's path is usually moving around, but not in a chaotic way but according to interest areas, and jumping short movements: *the saccades*. In Figure 2.8 it is depicted a human gaze path and its saccades' jumps onto an image. However, it doesn't mean that a computer is going to need making a same kind of visual travel over an image, mainly, because machines do not know which are the areas of interest and probably it is necessary providing them some initial patterns. But in order to define a fast-simple technique, it is

possible to adapt some concepts related with human behaviour, per instance, establishing the mouth as an area of main interest. Definitely, the mouth is a *hot* region where focusing the study. Once the center of the mouth is determined, it is also possible to define a constant exploration trajectory. The mouth limits - marked by lips silhouettes and shapes - can be reduced into a particular figure: the infinite-shape or Lemniscate of Bernoulli - that is represented in Figure 2.7 . In addition, this shape can be easily transformed in a trigonometrical representation, like the Equations 2.2 and 2.3 represent.

$$x = \frac{a\sin(t)}{1+\sin^2(t)} \tag{2.2}$$

$$y = \frac{a\sin(t)cost(t)}{1+\sin^2(t)} \tag{2.3}$$

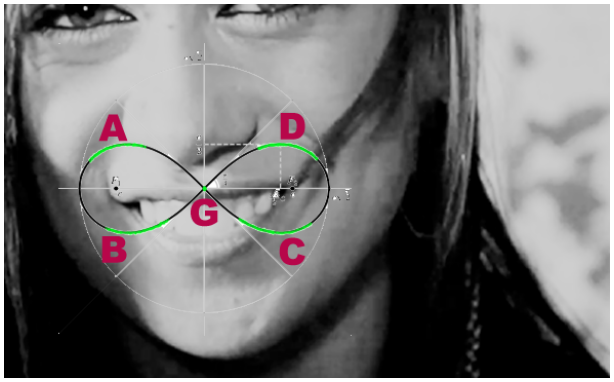


Figure 2.9: Lemniscate path over an image captured and processed

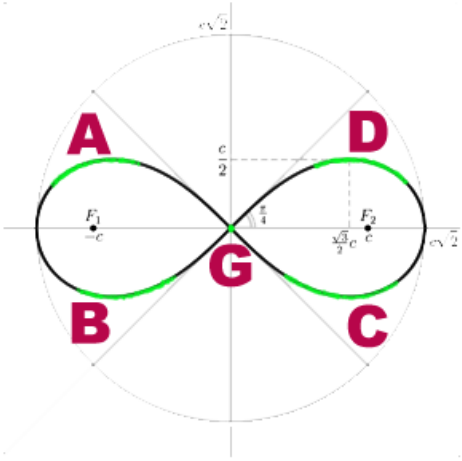


Figure 2.10: Particular areas of interest - green zones.

The method's application consists in the analyse, by following the path marked by the Lemniscate, of several particular regions of the mouth area - as it is shown on Figure 2.9.

As it also depicted on Figure 2.10, there are five *sensitive* - green - parts: *A, B, C, D and G*. These regions are fundamental in order to determine which is the face's expression. The work method consists in comparing the pixels comprised on the lemniscate thin path - just a one-pixel width curve. If these pixels have a value among dark grey and black - 70 types of shades are taken into account [6] - it is possible to conclude that part of the mouth or the lip's borders are overlapping the lemniscate trajectory. Depending on the overlapping combinations it is also possible to make particular hypothesis about the expression.

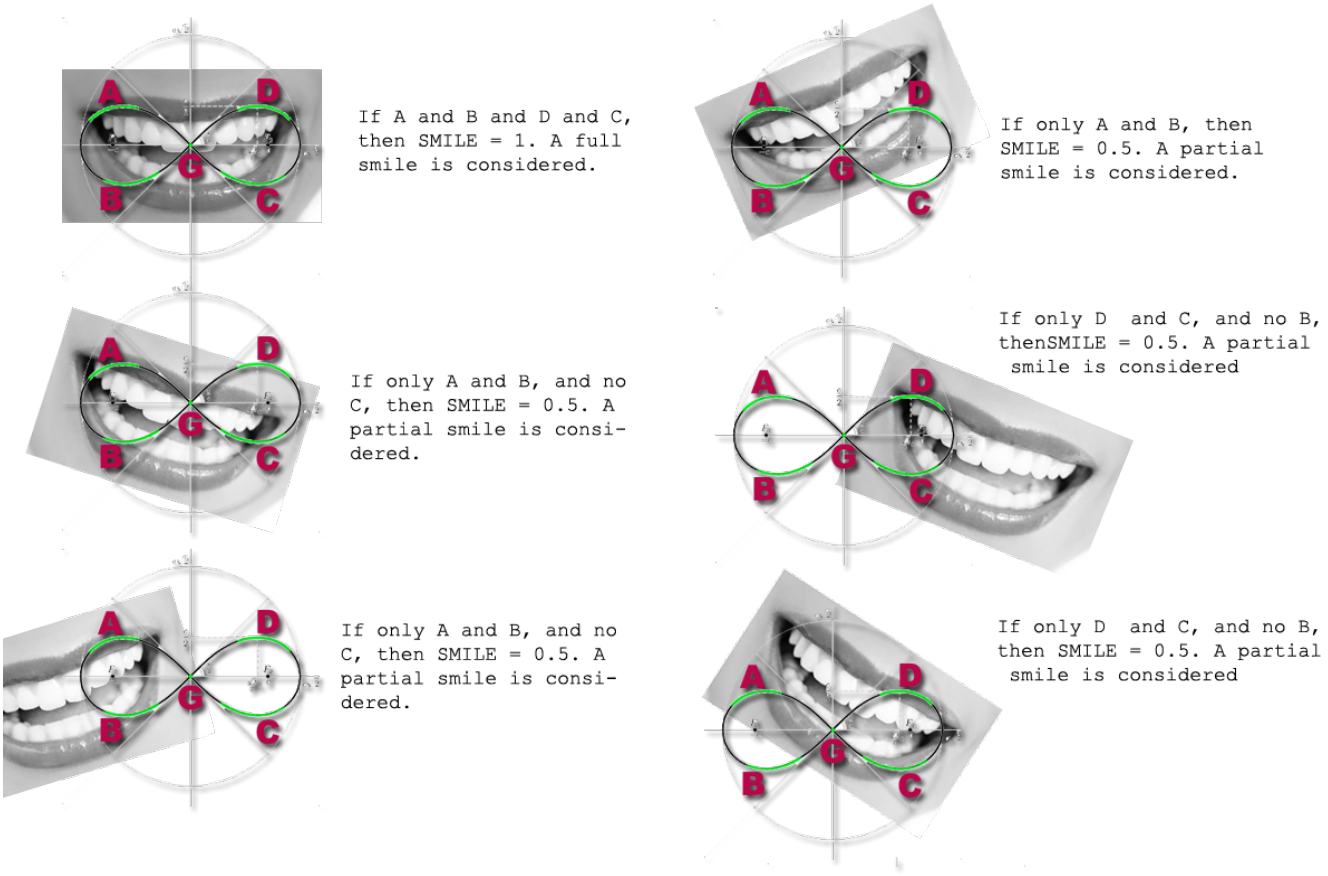


Figure 2.11: Some combinations that corresponds to smiling states.

Finally, after all the samples are analyzed, the system extracts a definitive prognosis about the expression detected. Figure 2.11 depicts a big part of possible combinations, and the weighting values given in order to determine a possible smile. In Figure 2.12 some examples are represented of what is considered as a Not-Smile - or a disgust expression, although this U-inverted shape is not always related with the sadness or disgust feeling. It is also important to note that the rest of the results are considered as neutral states - some examples are shown on Figure 2.13.
In order to minimize the error of measurements - partly caused by Android's constraints -

[6] In hexadecimal: #000000,#05050,#08080 ... ,#B3B3B3

it is needed to establish a *quality threshold*. As it has been explained before, the measures are weighted. So that, it is also possible to set an initial quality limit to overcome, in order to make a better assertion about a particular expression and exclude erroneous measures. On a five-samples process, this threshold level has been set to 1.

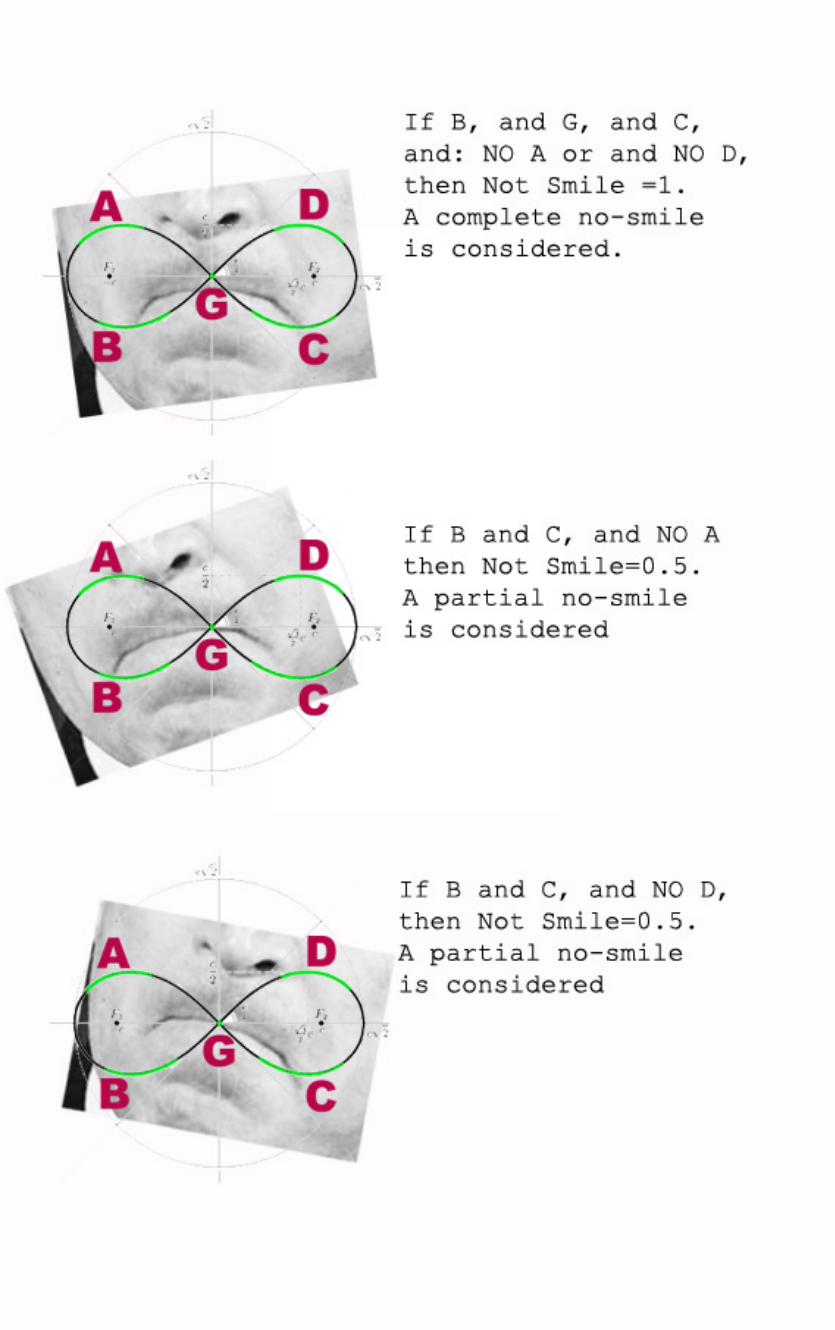


Figure 2.12: Some combinations that corresponds to disgust states.

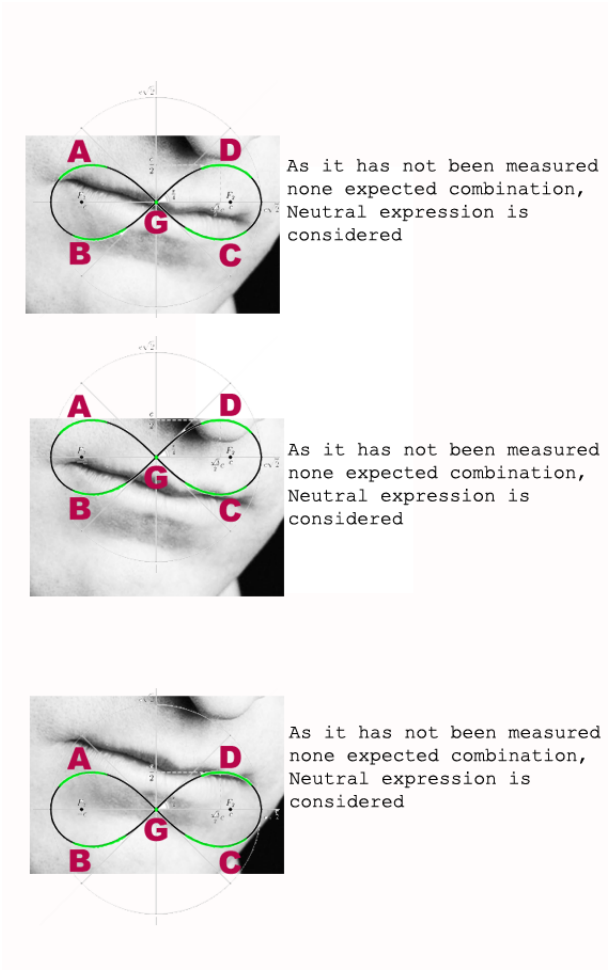


Figure 2.13: Some combinations that corresponds to neutral states.

Besides, some other strategies are necessary to improve the processing time. A considerable amount of time is spent during the pixel's analysis - Lemniscate's Trip - and next solutions make this *travel* shorter:

- There is no need to explore the entire Lemniscate curve, but just some parts of its period: region A $\{^{3\pi}/_{4} to ^{15\pi}/_{16}\}$, region B $\{^{-5\pi}/_{8} to ^{-7\pi}/_{8}\}$, region C $\{^{\pi}/_{8} to ^{3\pi}/_{8}\}$ and region D $\{^{-\pi}/_{16} to ^{-\pi}/_{4}\}$. Region G corresponds to a four-pixels cross on the center point of the Lemniscate curve, as it is shown on Figure 2.9.

- Even the periods above don't need to be fully explored. Once the system finds an active pixel - is gray enough -, the finding process is immediately stopped. There is no reason to complete the remaining exploration.

Finally, the application makes user know the assertion about the expression. Next chapter includes the obtained results during a particular number of test experiments, and also, the constraints of the technique. There are analyzed expressions of joy, disgust and neutral.

Chapter 3

Results

Here are presented the results obtained during the system's tests, and also, some drawbacks detected and related with the technique. During the next points both the method, statistics and results, are going to be developed. What is studied here is the dialog among a human *making* and expression and the device that have to interpret it.
Firstly, it is necessary to consider the fact that Human's expressions have a huge range of visual particularities, due to:

- Individual's physical traits - e.g. beard, moustache, wrinkles, lack of hair,
- individual's behaviour - is moving, talking or totally static,
- external factors, mainly light conditions - in a dark place or extra-sunny,

All these features shape the final expression, the sample which is necessary to test. On the other side, smartphone's camera devices offer some particular constraints, such as:

- Light conditions poor response - e.g. on high contrast situations, details can be lost.
- Bad time response - it is necessary to wait for taking and processing the picture, this is what *Acquisition Time (At)* mesures. And also, some time is needed in order to process the information and get the final assertion: the *Processing Time (Pt)*.

According to the previous scenario, an initial group of more than 130 images - videos and still pictures - has been selected in order to start testing the technique. As no particular conditions has been preestablished, several surprises immediately appear:

- The Face detection system and the associated trigger, is usually working with difficulties when the individual is wearing glasses, a hat, a mask or similar, brackets, is having a moustache or beard. It doesn't mean that the face is not recognized, in fact, the system detects the face. The problem is the rectangle which is associated to the bounds of the face: it is partially uncorrect. As the technique is using the rectangle to determine the mouth's region, it drives to a wrong sampling and test. In order to minimize this error, a five-samples system is necessary.

- The lemniscate technique has some problem dealing with high contrasted faces, per instance, with individuals which have big wrinkles. After processing the image captured - and contrasting it - the technique can interpret a wrinkle as it was a lip. However, these physical traits can be significantly reduced if the lighting conditions are regular, soft and no directly exposed to the light source -e.g. with the sunlight against the face. This can avoid having annoying shades on individual's faces. Although this is going to be excluded from the experiments, some related solutions are proposed in the Chapter 4.

- The technique doesn't work efficiently with low resolution images - a minimum amount of visual information is needed -, and non frontal pictures - profile images make the face detection and processing go wrong. Because the *Processing Time* is high, there is important to avoid talking people - because the lip's shape is changing fast -, camera movements - such as travellings, zooms in/out -, or individual's tosses.

Finally, 48 images have been selected and analysed, both motion picture and still images. The common visual conditions are: closeup pictures, soft enlightened, frontal depict, no movement and no talking. However, and as it is explained below, some images with different particularities have been also tested in order to evidence some of the technique limitations. At this moment it is important to notice that:

- The aim is to demonstrate that the system is working properly. The accuracy rates are interesting for it, but it is also important detecting the limitations in order to go further in future works.

- Some experiments with real individuals - not recorded images - have been made. However, the system - as it has been explained before - still needs five samples to complete a test. It means too much time in order to capture natural human expressions and it implies individuals acting during the experiment time. Actually, it is the same result that using recorded images, because the natural response is lost in both cases. So that, to experiment with real individuals has been finally discarded.

Figure 3.1 is a visual summary of all the work implemented until this moment. All the obtained results, and also the found limitations, are exposed on the next section.

3.1 Experiments and drawbacks

As it is explained before, the experiments have been made by using a group of 48 samples. The statistic method applied is partially related with the *holdout method*. This relationship is because an initial group of data - more than 130 images- has been divided in one part to train the system, and another to test it. However, it is important to note that the solution developed does not automatically learn, as those implemented with SVM (Support Vectors Machine). It means that statistic methods such as the real *Holdout* or *Cross-validation*, e.g. *K-fold Cross validation*, are not suitable to be applied here. The statistical study with the 48 samples is developed according to the next stages:

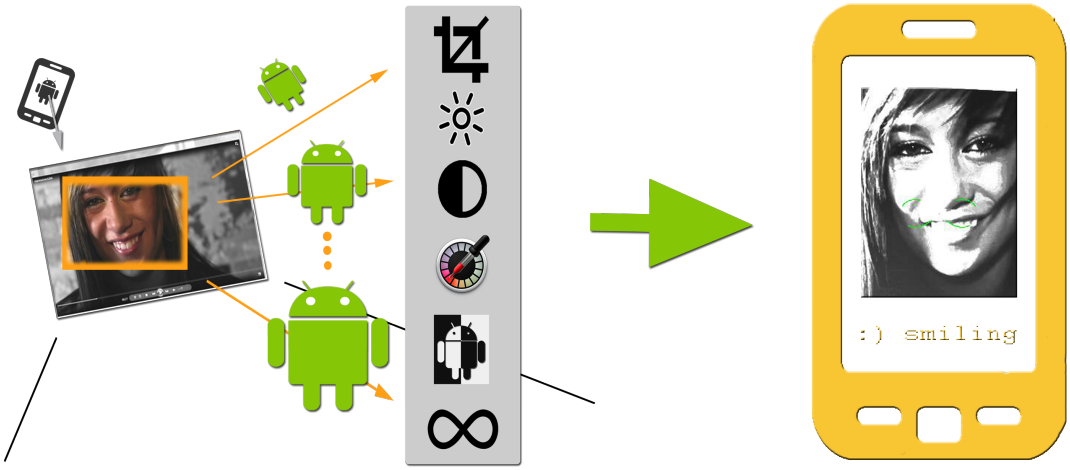


Figure 3.1: Main steps of the method

- Each sample - image - is tested at maximum three times,
- In order to accept a valid result - the assertion about if the expression is related with joy, disgust or neutral - two equal samples are needed,
- If each of the three results are different, the image is rejected. It has happened only on 3 images out of 48,
- Some times the face detection process does not work properly - because of the reasons previously pointed. If the process breaks, stops or the face detection is wrong - the rectangle area is out of the face region - it is necessary to start again with the test of this particular image. Fortunately, it has occurred only in some singular cases.

The **real-time** performance; in order to determine this issue, it is important to take into account the next information:

- The *Acquisition Time (At)* for each picture is approximately of 50-70 milliseconds.
- The image visual processing - contrasting, cutting, and other steps detailed in Chapter 2 - is about 50 milliseconds. The analysis of each capture to recognize the expression lasts 200 milliseconds
- So that, the resulting *Processing Time (Pt)* is 250 milliseconds.
- The total time *At+Pt*, for each capturing, sampling and saving of the image to the device's memory is *300 milliseconds*. Each image processed is saved in order to have a better control of the experiment process, and allowing a further exam. However,

saving the file would not be necessary and a little while could be saved. But since this time is much more smaller than the time wasted waiting for the five face-detections, the saving-file option has been left.

- The average time to complete the five-samples round is about *30 seconds*. A lot if considering the time needed to process each sample, and consequence of *face detection's* slow response.

In sum, as the 5 samples are mainly necessary due to the *FaceDetection* and the *device's native* constraints, it is possible to state that the system and Lemniscate technique could perform on real time in case that just two samples were necessary. In this scenario, it could be possible interpreting an expression in about one second. This time is considered suitable to work with natural human responses, and this should be a milestone to reach in further implementations of Lemniscate technique on future faster and accurate devices.

Finally, it is time to study the technique's accuracy. For that, 45 images have been analyzed. All the experiments have been video recorded [1] and the results are shown on the following sections.

3.1.1 Disgust - non smiling - expressions

As Table 3.1 shows, the number of correct recognitions of images depicting disgust expressions, is 7 out of 15. A binomial distribution calculation, according to Equation 3.1, *where n= number of samples (15); x= number of total successful results; p= probability of success per sample (0.33); q= probability of failure per sample (1 - p)*, offers the values of the probability according to 7 successful results: $P(>=7) = 0.195$ and the $P(<=7) = 0.916$. The accuracy rate is 47%.

$$P(x) = \frac{n!}{(n-x)!x!} p^x q^{n-x} \tag{3.1}$$

As the success rate is higher than the probability rate - 19.5% - it is possible to conclude that the system is correctly detecting the disgust expressions[2].

[1] A video sequence with all the images is available on `https://www.dropbox.com/s/b37jbti0hrxzyvy/0riginalvideos_lowr.mp4`

[2] There is available a video sequence with all the recorded experiments about disgust expressions , as well as the final results: `https://www.dropbox.com/s/f0rgpier15jv9mb/notsmiletestsALL_lr.wmv`

Experiments	Test 1	Test 2	Test 3	Average	Results
Sample 1	Disgust	Disgust	X	Disgust	OK
Sample 2	Disgust	Disgust	X	Disgust	OK
Sample 3	Disgust	Neutral	Joy	?	REJECTED
Sample 4	Joy	Joy	X	Joy	WRONG
Sample 5	Neutral	Neutral	X	Neutral	WRONG
Sample 6	Joy	Joy	X	Joy	WRONG
Sample 7	Neutral	Disgust	Disgust	Disgust	OK
Sample 8	Neutral	Neutral	X	Neutral	WRONG
Sample 9	Joy	Neutral	Joy	Joy	WRONG
Sample 10	Neutral	Neutral	X	Neutral	WRONG
Sample 11	Neutral	Disgust	Disgust	Disgust	OK
Sample 12	Neutral	Neutral	X	Neutral	WRONG
Sample 13	Neutral	Joy	Disgust	?	REJECTED
Sample 14	Disgust	Joy	Disgust	Disgust	OK
Sample 15	Disgust	Disgust	X	Disgust	OK
Sample 16	Neutral	Neutral	X	Neutral	WRONG
Sample 17	Disgust	Disgust	X	Disgust	OK

Table 3.1: Experiments with disgust - not smiling - expressions

Figure 3.2 and Figure 3.3 depict an entire successful recognition of a disgust expression. It is possible to see the samples which the system processes and saves into the device, also in grayscale, and that are placed next to the original image - in colour. As it is possible to note, the Lemniscate technique is applied just on particular regions of the image. When some grayscale area is detected the system stops, by saving time and as it has been explained in Chapter 2.

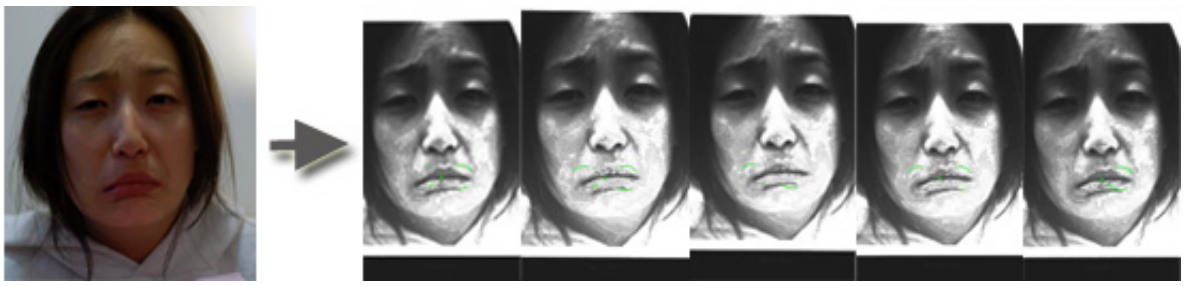


Figure 3.2: Image sequence from the Sample 1, Test 2, of the experiments with disgust expressions

Figure 3.3: Image sequence from the Sample 14, Test 2, of the experiments with disgust expressions

In Figure 3.4 one drawback appears. The system asserts that this image corresponds to an smiling face. The reason is because of the skin wrinkles which the individual has next to the mouth. These grooves are transformed in black lines after the image processing - in fact, originally they already are dark lines. The application interprets this trait as part of an smiling mouth, and unfortunately, it is not possible to state that this is just a singular feature of few individuals. Usually, when human individuals act out the *U-inverted* expression, they tend to contract specific muscles of the face, resulting these symmetric black lines on each side of the mouth. This constraint would need to be solved on future works, per instance, by applying a particular control on these areas: as they are two symmetrical almost-straight lines and associated only with disgust expressions, it could be useful to combine these features - e.g, comparing the same symmetric and complementary regions of the Lemniscate - above and below, left and right - and if they are cutting on the same positions and in both sides of the mouth, it could be possible to state that two vertical black lines are crossing the individual's face. In case this trait happens also when other disgust manifestations are recognized, the *smiling* option could be rejected.

Figure 3.4: Image sequence from the Sample 4, Test 2, of the experiments with disgust expressions

In Figure 3.5 it is possible to note other constraint of the technique. In fact, it is related with other drawback stated above. Again, some annoying black lines are going to confuse the application, which recognize a smiling expression when it is not at all. In this particular case, these black lines are not only wrinkles but also part of the moustache. These *hairy* traits are another drawback to overcome in future works. Per instance, by implementing a strategy similar to the one applied to the wrinkles' problem.p
However, asymmetric traits or objects, such as beards, masks, tatoos, brackets or singular hairstyles, are potentially harmful for the system and they need a focused study on that.

Experiments	Test 1	Test 2	Test 3	Average	Results
Sample 1	Neutral	Neutral	X	Neutral	OK
Sample 2	Neutral	Disgust	Disgust	Disgust	WRONG
Sample 3	Joy	Joy	X	Joy	WRONG
Sample 4	Neutral	Neutral	X	Neutral	OK
Sample 5	Joy	Neutral	Joy	Joy	WRONG
Sample 6	Neutral	Neutral	X	Neutral	OK
Sample 7	Joy	Joy	X	Joy	WRONG
Sample 8	Joy	Joy	X	Joy	WRONG
Sample 9	Neutral	Neutral	X	Neutral	OK
Sample 10	Neutral	Neutral	X	Neutral	OK
Sample 11	Joy	Neutral	Neutral	Neutral	OK
Sample 12	Disgust	Neutral	Joy	?	REJECTED
Sample 13	Neutral	Neutral	X	Neutral	OK
Sample 14	Neutral	Neutral	X	Neutral	OK
Sample 15	Neutral	Neutral	X	Neutral	OK
Sample 16	Disgust	Disgust	X	Disgust	WRONG

Table 3.2: Experiments with neutral expressions

Figure 3.5: Image sequence from the Sample 9, Test 3, of the experiments with disgust expressions

3.1.2 Neutral expressions

The number of correct recognitions is 9 out of 15. According to the results of the binomial distribution 3.1, the probability of this result is $P(>= 9) = 0.029$, and the $P(<= 9) = 0.992$. The accuracy rate is 60%. As the success rate is higher than the probability rate - 2.9% - it is possible to conclude that the system is correctly detecting the neutral expressions[3].

Figures 3.6 and 3.7 show two successful recognitions. Rightly interpreting this state is crucial, because it is a difficult one: some mid-smiling or cynical expressions are going to push the application's limits and it is important to be aware of that on future works.

[3] A video sequence with all the recorded experiments about neutral expressions is available, as also all the results: `https://www.dropbox.com/s/1rll4nhak4c22j6/neutraltestsALL_lr.wmv`

Figure 3.6: Image sequence from the Sample 4, Test 2, of the experiments with neutral expressions

Figure 3.7: Image sequence from the Sample 10, Test 1, of the experiments with neutral expressions

As it happens with other type of expressions, some visual features can confuse the system's prognosis. Per instance, in Figure 3.8 it is possible to perceive an important challenge to overcome: individuals with dark skins and large lips. Since the technique works by contrasting the image, if the original is dark - and few light-contrasted - the result is not going to be conclusive. In order to solve this limitation in future implementations, making a previous analysis of the particular image visual traits could be useful. This can allow further applying of the suitable image processing for each image. In the case of individuals with dark skins, maybe other processing techniques could be used, such as controlling not only the black but also the white regions - e.g. on the mouth, by studying the teeth zone, in white, and its shape.
There is also the difficulty with the large lips which can trouble the technique. Again, it could be useful studying the black pixels' density among the teeth and skin zones, and resizing the Lemniscate a value, as it is shown on Figure 2.7, according to the lips' real size.

Figure 3.8: Image sequence from the Sample 3, Test 1, of the experiments with neutral expressions

In Figure 3.9 it is possible to perceive another visual difficulty, which is not related with individuals' physical traits but with the performance of camera devices. The system is interpreting a partly-shaded face in a wrong way. The reason is related with the camera's diafragma and the automatic setting of that. The device looks for the best way to correct the highlighted area - and closes the diafragma - but since the visual response of these

Experiments	Test 1	Test 2	Test 3	Average	Results
Sample 1	Joy	Joy	X	Joy	OK
Sample 2	Disgust	Disgust	X	Disgust	WRONG
Sample 3	Joy	Joy	X	Joy	OK
Sample 4	Joy	Joy	X	Joy	OK
Sample 5	Disgust	Disgust	X	Disgust	WRONG
Sample 6	Joy	Joy	X	Joy	OK
Sample 7	Joy	Joy	X	Joy	OK
Sample 8	Neutral	Neutral	X	Neutral	WRONG
Sample 9	Disgust	Joy	Joy	Joy	OK
Sample 10	Disgust	Joy	Joy	Joy	OK
Sample 11	Joy	Joy	X	Joy	OK
Sample 12	Disgust	Joy	Disgust	Disgust	WRONG
Sample 13	Disgust	Joy	Joy	Joy	OK
Sample 14	Joy	Joy	X	Joy	OK
Sample 15	Joy	Joy	X	Joy	OK

Table 3.3: Experiments with joy - smiling - expressions

devices is clearly worse than human's eyes, and its latitud range is poor, the result is that the other zones - which were dark - ends up almost black. This bad-performance on contrasting transforms the capture in a *two-face* image that is going to make the system fail. A future solution could be a previous correction of the dark zones without altering the more enlightened ones, and in order to have an homogeneous image.

Figure 3.9: Image sequence from the Sample 8, Test 1, of the experiments with neutral expressions

3.1.3 Joy - smiling - expressions

The number of correct recognitions is 11 out of 15. By calculating the binomial distribution [3.1], it is possible to know that the probability of this result is $P(>= 11) = 0.0016$, and $P(<= 11) = 0.9997$. Finally, the accuracy rate is 73%. As the success rate is quite higher than the probability rate - 0.16% - it is possible to conclude that the system is correctly detecting the joy expressions[4].

Figures 3.10 and 3.11 show two successful sequences.

[4]A video sequence with all the recorded experiments about joy are available, as also the final results: `https://www.dropbox.com/s/xlqmtw5d0yyq2za/smiletestsALL_lr.wmv`

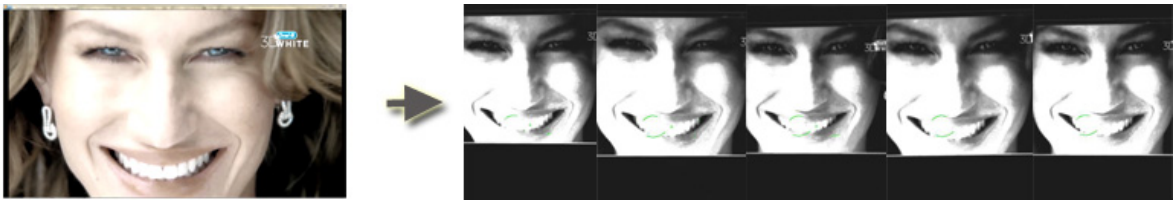


Figure 3.10: Image sequence from the Sample 10, Test 2, of the experiments with joy expressions

Figure 3.11: Image sequence from the Sample 4, Test 1, of the experiments with joy expressions

Figure 3.12 illustrates a constraint which has also been studied above, on the neutral expressions recognition section. Again, problems due to the lighting and too much contrasted regions, which could be solve with a specific correction of each area. It is also important to note another drawback that is stated on the Figure 3.13: what does it happen when the individual is moving and speaking? Today, the application is probably going wrong, because the individual opens and closes the mouth and the system makes a right or wrong prognosis, depending on the type of each sample. This trouble is clearly related with the *Response Time*, and it could be fixed if the system was faster. This is an interesting fact of improving in future works.

Figure 3.12: Image sequence from the Sample 5, Test 1, of the experiments with joy expressions

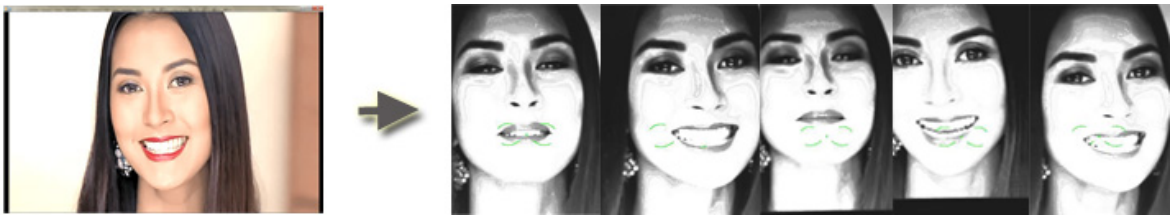


Figure 3.13: Image sequence from the Sample 12, Test 1, of the experiments with joy expressions

The main conclusions of this chapter are:

- It is demonstrated that the system is able to rightly recognize joy - 73% in a set of 15 images - , disgust - 47% in a set of 15 images - and neutral - 60% in a set of 15 images- expressions.
- It is needed a further improvement of the *Response time* and also, of the system's performance with highly contrasted images, particular physical traits - such as dark skins, hairstyles, wrinkles or large lips - and singular objects wearing - such as glasses, dental braces, hats or masks.

In the next subchapter a particular case is studied: what does the system think about the Mona Lisa? Is she smiling?

3.2 The Mona Lisa Smile

This part is an intriguing recommendation of Dr. Enric Martí, the advisor of this project. He is wondering about how the technique would perform with a picture of Mona Lisa. Historically, it has been a mystery if Leonardo's had painted one smiling, cynical, haughty or shy lady. While - as it has been said in chapters above -, the system here implemented is not able to recognize affections, the Mona Lisa's expression interpretation is a challenge by itself.

Figure 3.14: Image sequence from Test 1, of the experiment with the Mona Lisa, resulting NEUTRAL

As it is depicted on Figures 3.14, 3.15, 3.17, the system prognosis are: Neutral, Smiling and Smiling. So that, the final conclusion is that the Mona Lisa is smiling... Is the mystery finally solved?

Figure 3.15: Image sequence from Test 2, of the experiment with the Mona Lisa, resulting SMILING

Figure 3.16: Image sequence from Test 3, of the experiment with the Mona Lisa, resulting SMILING

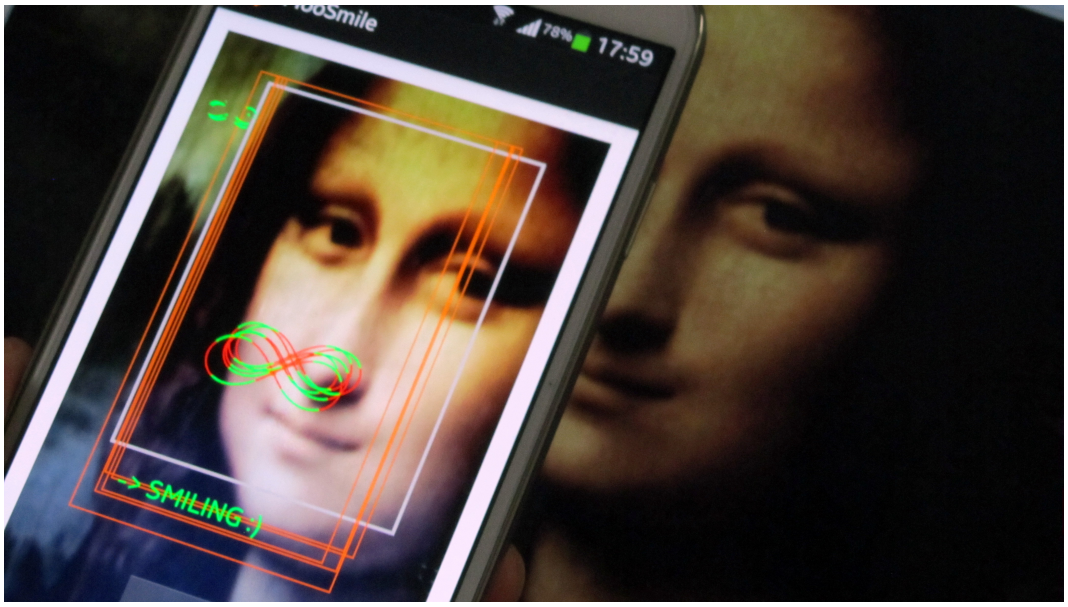


Figure 3.17: Result of Mona Lisa testing

Chapter 4

Conclusions and Future Work

It is presented a new expression-detection solution - an Android Application - which implements a recognition technique that makes possible to recognize expressions of joy, disgust and neutral, on real-time, in a simply way and on mobile environments. A large number of experiments has been made in order to determine a starting level of accuracy of this method's prototype. Still images, video recordings and real life images have been used to set up the working algorithm and to detect some limitations, as also possible ways of future improving.
The actual state of the application makes possible an immediate application in order to develop simple interfaces or applications - per instance, it could be a fast way to obtain images from millions of users around the globe, enhancing images databases used by other existing solutions. Further research could be related with making this technique more complex in order to detect a wider range of expressions. Maybe, through future combinations of different expressions, it could be possible detecting emotions as well. At this moment, the development of related social systems - such as those with medical, veterinarian or educational purposes - will be a little bit closer.

The main findings of the project are:

- It has exposed and tested a new technique of recognition: *The Lemniscate's Trip*, whose fast performance and easy implementation make it suitable for mobile devices.
- It has been demonstrated that the system performance is accurate: 47% on neutral expressions out of 15 test images, 60% on disgust expressions out of 15 test images, and 73% on joy expressions out of 15 test images.
- It has been indicated that a system based just on image processing is viable - without needing mathematical transformations . Thus is far important in order to develop mobile - or even simpler - systems, which have calculus power limitations and difficulties to deal with a lot of graphical data.
- The developed technique makes the expression recognition possible without using visual references - as images databases in order to define classifiers. In fact, the system is using the real face's captured images - on real-time - to acquire some reference

points and to analyze them during the process. This is also an important fact in order to develop *lightweight* systems.

- It is has been detected that today's mobile technologies have still some constraints. Android's top-of-the-range devices - like Samsung S4 - present poor performances on Face Detection results, and also, they are not providing qualitative data related with its own native specifications - e.g. not offering eye's or mouth's coordinates, although it is specified on Android Api 14 definition.

- The results indicate that it is possible a system based on a one-sample method for recognizing expressions on-real time. For the moment, the application needs to correct today's mobile devices limitations, by acquiring extra samples which are necessary because the initial error level - added by the measurement factory drawbacks.

- Some expressions related with particular emotions are more easily to detect than others. The smile - and its physical counterparts - offers a reference pattern much more feasible to detect than disgust expressions.

The main limitations of the current work are listed below, complemented with related future work:

- The system could be faster in case it was not necessary having extra samples to correct native errors related with Face Detection performance. At the future, once these factory bugs are addressed, the application will immediately gain more speed of work. Probably, in that moment, it will be automatically able to just working with 2 samples in order to detect an expression variation, and with 1 sample for recognizing a particular type of expression.

- The image processing step - brightness, contrast and saturation process - is based on constant parameters and could be enhanced. The captured image's conditions are really important, so that, it is recommended to develop a system of image capturing which is automatically correcting some original drawbacks, such underexposed or overexposed zones of the image.

- By the moment, the mouth center is geometrically approached. Once Android Api 14 features are really working properly, the center of the mouth will be provided by the device and the previous approach won't be necessary. In consequence, the obtained samples quality will immediately improve.

- The process is designed to reject those pictures which do not provide clear information. However, in order to get the conclusion of rejecting a sample, also an amount of processing time has been wasted. So that, it would be very useful any future improvement in order to extract information from those *useless* - at the moment - image captures. Per instance, some particular features could be explored onto this captures, such as white colour density on mouth's area which could be a trace of a *possible* smile.

Bibliography

[1] Pamela K. Adelmann and R.B. Zajonc. Facial efference and the experience of emotion. *Annual Review of Psychology*, 40(2):249–280, 1989.

[2] Jeremy N. Bailenson, Emmanuel D. Pontikakis, Iris B. Mauss, James J. Gross, Maria E. Jabon, Cendri A. C. Hutcherson, Clifford Nass, and Oliver John. Real-time classification of evoked emotions using facial feature tracking and physiological responses. *Int. J. Hum.-Comput. Stud.*, 66(5):303–317, 2008.

[3] Charles Bell. *The Anatomy and Philosophy of Expression as Connected With the Fine Arts*. Henry G.Bohn, 6 edition, 1872.

[4] Charles Darwin. *Expression of the Emotion in Man and Animals*. D. Appleton and Company, 1 edition, 1873.

[5] Paul Ekman. Facial expression. *American Psychologist*, pages 384–392, 1993.

[6] Paul Ekman and Wallace V. Friesen. Constants across cultures in face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[7] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, 2010.

[8] Ursula Hess and Pascal Thibault. Darwing and emotion expression. *American Psychologist*, 64(2):120–128, 2009.

[9] Mohammed E. Hoque, Daniel McDuff, and Rosalind W. Picard. Exploring temporal patterns in classifying frustrated and delighted smiles. *T. Affective Computing*, 3(3):323–334, 2012.

[10] Carroll E. Izard. Emotion theory and research: Highlights, unanswered questions and emerging issues. *Annual Review of Psychology*, 60:1–25, 2009.

[11] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry A. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] Daniel Lélis Baggio, David Millán Escrivá, Naureen Mahmood, Roy Shilkrot, Shervin Emami, Khvedchenia levgen, and Jason Saragih. *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing, 1 edition, 2012.

[13] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

[14] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009.