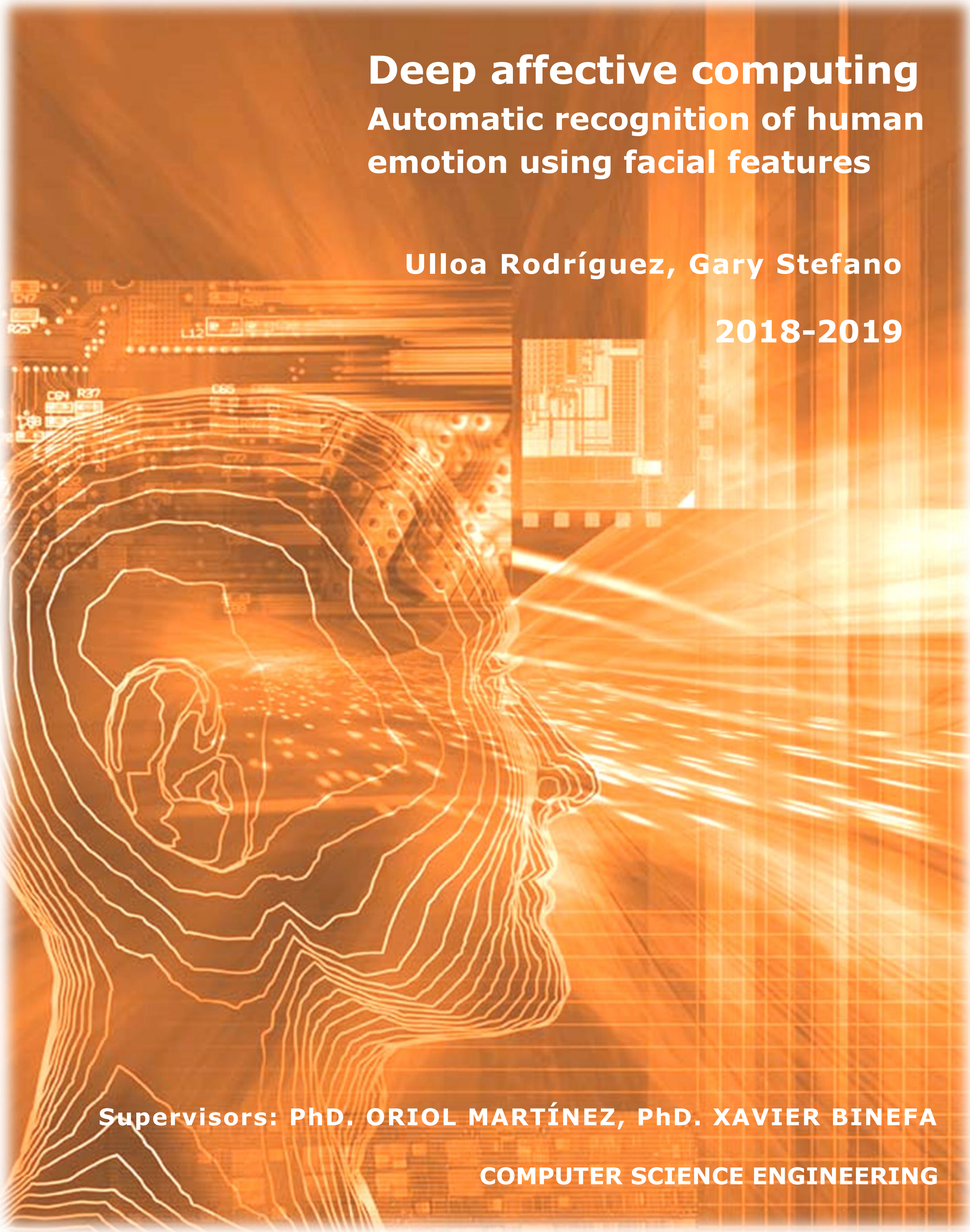

Deep affective computing

Automatic recognition of human emotion using facial features

Ulloa Rodríguez, Gary Stefano

2018-2019

Supervisors: PhD. ORIOL MARTÍNEZ, PhD. XAVIER BINEFA

COMPUTER SCIENCE ENGINEERING

upf. Universitat Pompeu Fabra Barcelona
Escola Superior Politècnica

Bachelor's Thesis

Deep affective computing

Automatic recognition of human emotion using facial features

Gary Stefano Ulloa Rodríguez

BACHELOR’S THESIS UPF / YEAR 2018-2019

SUPERVISORS

PhD. Oriol Martínez Pujol, PhD. Xavier Binefa Valls

COGNITIVE MEDIA TECHNOLOGIES RESEARCH GROUP

upf. Universitat Pompeu Fabra Barcelona

Dedications

Dedicated to my family and my university friends who worked on their thesis during the same time I did.

Acknowledgements

I want to thank my thesis supervisors: Oriol Martínez Pujol for its mentorship and for introducing me to deep learning concepts, projects and technologies; Xavier Binefa Valls for getting me into this project, talking about interesting investigations and making me excited to work in the field. Last but not least, I want to acknowledge Decky Aspandi Latif for its great support, particularly in the technical side of the project, even in the more critical parts.

Abstract

Deep learning research on computer vision has boomed during this last decade, experiencing its greatest achievements (far surpassing humans at certain tasks). Meanwhile, computers, smartphones, websites and apps are becoming integral parts of our lives, communications, jobs and leisure; affective computing is a field in which computing systems are designed so that they can adequately take into account human emotions (recognizing or simulating human emotions) for a better, more successful and useful interaction between user and machine. In this work we review the latest advancements in emotion recognition research, and construct different deep learning architectures to test and challenge the state-of-the-art. Convolutional neural networks are designed to predict the emotional levels of the users from their facial features found in images or videos, following the continuous emotion classification model of valence and arousal (as opposed to the conventional discrete models of “basic emotions”),.

Resumen

La investigación de deep learning (aprendizaje profundo) en la visión por ordenador ha experimentado un auge durante esta última década, experimentando sus mayores logros (sobrepasando enormemente a los humanos en algunas tareas). Mientras tanto, los ordenadores, los smartphones, las webs y las apps se están convirtiendo en partes integrales de nuestras vidas, comunicaciones, trabajos y ocio; la computación afectiva es un campo en el que se diseñan sistemas informáticos para que tomen en cuenta adecuadamente las emociones humanas (reconociendo o simulando emociones humanas) para una mejor interacción, más exitosa y útil, entre usuario y máquina. En este trabajo se revisan los últimos avances en la investigación del reconocimiento de emociones, y se construyen diferentes arquitecturas de deep learning para probar y desafiar al estado del arte. Redes neuronales convolucionales son diseñadas para predecir los niveles emocionales de los usuarios a partir de sus atributos faciales encontrados en imágenes o vídeos, siguiendo el modelo continuo de clasificación de emociones de valencia y excitación (frente a los modelos discretos convencionales de “emociones básicas”).

Preface

We currently live in a time where computers, smartphones, websites and apps (just to name a few examples) have become essential for some of the matters of our daily life. It is expected that human interaction with computing systems will be more present and more necessary in the future than it has ever been, accelerated by technological trends such as the arrival of more ubiquitous computing devices, the creation and improvement of social robots and wearables, the development of effective augmented and virtual reality systems, the growing adoption of blockchain, the implications of introducing 5G, the refinements on autonomous cars and intelligent virtual assistants, the embrace of the sharing economy and the already strong use of social networks.

Computing systems will be instrumental in the appearance of new solutions to face shortcomings that are present in our societies, as long as humans, when using them, can successfully interact with these systems. Otherwise, the effectiveness of such systems would suffer; there would be severe restraints in the adoption of these new technologies. Driven by this realization, I thought of the necessity of researching the affective computing field. This field comprises the development of emotion recognition systems, which is the focus of this thesis work. Emotion recognition systems could be equipped to other systems, be it for labour reasons, for leisure ones or for well-being purposes, so that the interaction between human and device and/or software is much more advantageous. Above all, this would result in a less shocking shift to whatever direction these new technologies guide our societies to than it otherwise would.

Apart from that, I have to bring up the subjects of machine learning and deep learning, both of which will be responsible for some of the technological developments of the near future previously mentioned. They are being heavily used by big high tech companies, start-ups and research groups to developed groundbreaking systems in many fields, such as health, commerce, security, entertainment, ... It is therefore not surprising that machine learning and deep learning are being currently employed in the research and the business sectors for developing emotion recognition systems.

While studying the computer science degree, I came across machine learning in many subjects, where I observed its great potential, especially that of the deep learning subfield, so I have been eager to deepen in these areas. At the UPF's Cognitive Media Technologies Research Group I discovered deep learning projects focused on visual emotion recognition. Thus, the thesis, given that it is an exhaustive and focused work, seemed ideal to tackle both deep learning and emotion recognition. Working on a deep learning system helps in providing many knowledge and practice for developing others even if they greatly differ, which inspired me because I would like to extend the current work and apply it to a specific area (health).

Contents

List of figures

List of tables

1. INTRODUCTION

In this section I will first mention some of the reasons that led me to choose this project, and thereafter the main concepts of affective computing and emotion classification will be presented, accompanied by their current positioning, because they are essential for the topic of this work. Lastly, the goals will be commented, as well as the structure of this document.

1.1 Motivation

Regarding the motivations, there are many reasons why I chose to work on emotion recognition using deep learning and computer vision, so I will group them by technical interest and personal interest.

On one side, there is a technical interest. I always intended to improve my knowledge in machine learning, which is the reason why I thought the thesis would be an ideal time to do it. Even though it is a vast topic, the project allowed me to work with some key aspects of the field, such as data manipulation, cross validation or overfitting; as well as being able to inquire in the specifics of neural networks and facial recognition. Moreover, I had the opportunity to work on advanced research on image recognition and work with leading deep learning technologies, which I always heard hold great potential for future applications, much more than they do now.

On the other side, I got a more personal interest with the ideas that I got to discuss at the UPF's Cognitive Media Technologies Research Group, a visual emotion recognition project that could serve as a basis for future applications that would be much more complex and that could be used in pioneering fields (more on that in Future work, section 7). In particular, applications to diagnose, deal with or even treat health related issues. It made me realize of the possibilities of emotion recognition, possibilities I had not think before, which motivated me to work on this topic. I am confident that what I have learned with this project will have helped me to be prepared to immerse in more advanced applications.

1.2 Affective computing

The core field of study of this research is that of affective computing, specifically the automatic recognition of emotions using computer vision and machine learning. The concept of affective computing refers to the study and development of computing systems that can recognize and process human affect (emotions), as well as systems that, in addition to those, can simulate them [1]. The modern understanding of this computing branch originates with the research paper Affective Computing (1995) [2].

The basic process of the affective computing we will work with, which can be seen in figure 1.1, starts with the collecting of (physical or physiological) data using sensors (camera, microphone, electrode, ...), so that this data can be labeled (according to the type of information that wants to be obtained by the system) and grouped to be fed to a machine learning algorithm which learns (that is, extracting patterns from the training data) how to predict the value of unseen data, resulting in a system that responds according to the user's affect.

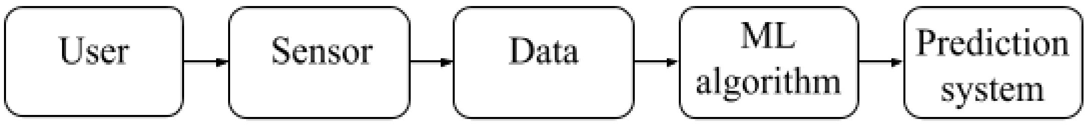

Figure 1.1: Diagram depicting each step in a machine learning system.

When discussing emotions in the context of computing technologies, we note the general trend (even after the publication of the Affective Computing 1995 paper) was to ignore the consideration of affect-related behaviours (whether as input or output). Currently, there are many efforts in the research environment to study and develop many such systems, based on the analysis of different modalities (emotional speech, body gesture, physiological monitoring, etc) or multimodal processes (which work on two or more modalities) [3]. Regarding affective computing on companies and web applications, the presence is still very limited. Many applications have been thought for being applied in varied fields (commerce, health, security, ...) but have a lack of deep investigation, are still not accurate enough or are being held back by other system components constraints.

It used to seem that emotion and informatics could not to be coupled (or at least, it was not intended), given that technology (as is science) is viewed as a product of rationality and precise and measured experiments, actions, and analysis, leaving no room for emotions [2]. The rejection of the approach to emotions was due in part to the fact that we still do not understand the intricacies of human emotions in its entirety. Nonetheless, there is evidence that emotion is crucial for learning, effective human communication and rational decision-making [4]. Thus, affective computing approaches cognitive science and psychology theories and investigations, as will be seen later when discussing emotion classification models. Nowadays, the informatics community has come to accept that artificial intelligence requires a deeper understanding of human emotions to advance in new directions and face current issues [3].

1.3 Emotion classification

Firstly, we have to take into consideration the emotion classification model that we will follow, due to the fact that the system will have to decide the kind of affect that the user is displaying (be it, for clear instances or for more subtle ones). The importance comes from the fact that the training data instances (the video frames) that the algorithm will learn from show a particular emotional state of a person, each instance having an associated label indicating an annotated value (given by a professional annotator, this originates an issue discussed in Annotations, section 3.4.2), which means the resultant system will be consistent with the chosen classification methodology because it will determine the possible label values [5]. The theory of emotion classification has two main stances: The discrete and the continuous (or dimensional) approaches. We will review a pair of theories from the affective sciences regarding the topic, since it will help establish the training dataset, as well as the type of system results.

The discrete approach states that there is a small set of distinguishable, universally shared human emotions. There have been many proposals, one of which is the one proposed by Paul Ekman in 1972 [6], which proposes the existence of six basic emotions (as shown in figure 1.2): anger, disgust, fear, happiness, sadness and surprise. Ekman stated that each basic emotion from his study was completely different to the rest of this group and also that these emotions were combined to create more complex emotional states. It has been argued that the use of a continuous model is the option to choose because the basic emotion theory does not adequately explain new findings in the scientific fields of affection [7]. The proper emotion classification theory has to be quite flexible and take better into account the complexities of human affect. Particularly if the intention is to develop a system which can deal with emotional states that are more subtle, ambiguous or that are not as easily decidable as the common options. Furthermore, even we as humans frequently have problems identifying other people's emotions, because answers are not always obvious. So this is a complex issue which requires a more flexible model, an issue that would benefit from a continuous model.

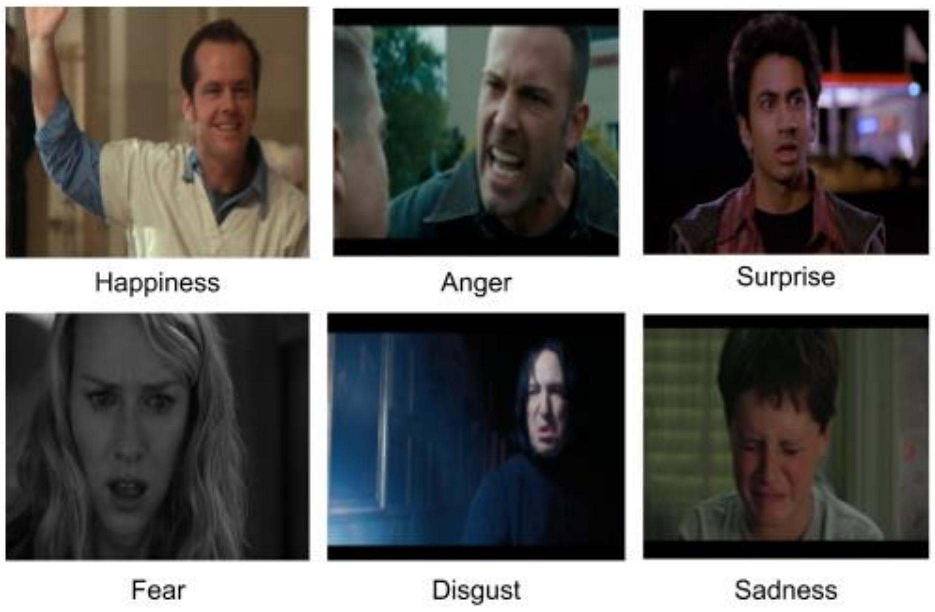

Figure 1.2: Examples of Ekman's basic emotions (images from AFEW-VA dataset [8])

The Circumplex model proposed by James Russell in 1980 [9] belongs to the continuous approach. It presents the notion that two dimensions can be used to map human emotions. One axis represents the valence, meaning how pleasant or unpleasant the state is, while the other represents the arousal, which determines the degree of excitement (activation or deactivation level). The standard values of both have a range of real values [-1, 1], although this range can be scaled (for example, AFEW-VA [8], the main dataset we work with, uses a range of [-10, 10]). These two measures can be joined as a circular two dimensional space (figure 1.3), where the center indicates neutral levels of both values and the extremes of the circle are the most intense levels. In addition to being able to answer unclear cases, this model can still work with the basic emotions, such as the ones from Ekman's research, because they can also be mapped in the classification circle: Thus, we can describe basic emotions as the result of a particular combination of valence and arousal levels.

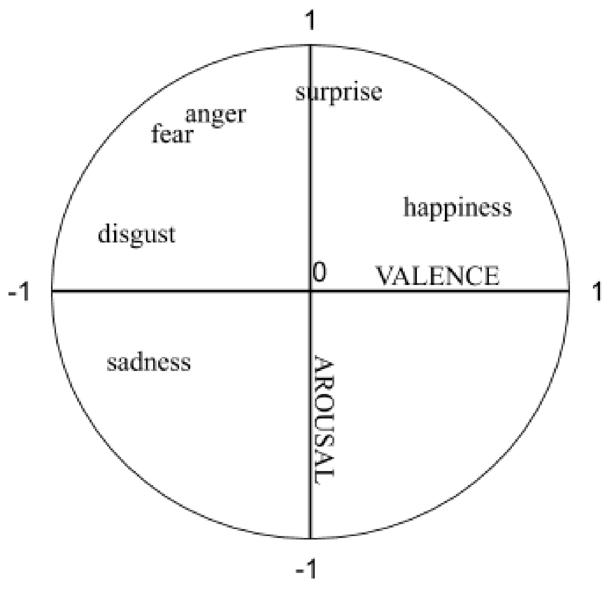

Figure 1.3: Standard two dimensional plane of valence and arousal. The six basic emotions from Ekman's research have been loosely placed in the plane according to their perceived valence and arousal levels.

1.4 Goals

A series of goals have been set in order to evaluate the outcome of this research work. When the project concepts and methodologies were first raised, some general goals began to be considered. As the study advanced, the objectives were clearly defined, and have served as targets that inspired the work since then. They were mostly influenced by the state-of-the-art. The list of objectives is composed by:

- First. Reaching the state-of-the-art in visual emotion recognition. Particularly for systems trained on in-the-wild datasets, although the same is desired with laboratory setting datasets. Following two studies that achieved state-of-the-art systems for this task, we will develop models to challenge the results of those other systems in terms of the performance measure metrics that those studies used (which, in fact, many researchers use for machine learning analysis).

- Second. Assessing the expected superiority of models trained on in-the-wild datasets in comparison with those trained on laboratory setting datasets in the sense that the former ones should generalize better when tackling laboratory setting data than the latter ones tackling in-the-wild data. The comparison will also be performed with the metrics used for the previous objective.

- Third. Obtaining a satisfactory real-time app that is powered by the best of the developed models. This app will be subjectively analyzed in terms of appreciable concordance between the answers generated by the system and the observable subject's emotional state.

- Fourth. Analyzing and comparing the ResNet [10] and VGG [11] architectures for the task of visual emotion recognition in terms of the metrics used for the other objectives. It is expected that ResNet will perform better than VGG.

There are other general goals that can be mentioned, even if they are harder to evaluate. The verification of these secondary goals will be indirectly performed when checking some of the principal objectives. One of these is the successful use of transfer learning, which can be determined when evaluating the first objective, since all the models will follow this technique. Nonetheless, it cannot be thoroughly analyzed because there will not be models trained from scratch (we follow the findings [7] of other studies that show transfer learning produces better results in the case of a short dataset such as AFEW-VA). Another goal is avoiding or preventing issues of overfitting, even though the occurrence of this issue is not confirmed. In case it appears it will have to be confronted. One last general goal is related to time constraints. Given the available hardware, one fold of a model will take between some days to some weeks to be trained (depending on the convergence of the results). We will try to parallelize the training of the maximum number of folds across the many models and also adjust the batch sizes so that the training of the models does not take months from the project. Although this is not mandatory to accomplish and the time constraints will be accepted as long as they result in benefits for the models, trying to train in the shortest possible time will be sought.

1.5 Structure of the dissertation

After having presented the motivations, the concepts of affective computing and emotion classification, as well as the goals in the Introduction, the structure of the dissertation is the following. The second large section is State-of-the-art, in which current research work on visual emotion recognition and companies using it will be commented. Then in the Theoretical framework section we will address machine learning and deep learning (what they are, how they are used, why we chose them), examine neural networks (how they work and how they are trained) and the specifics of convolutional neural networks, we will talk about some of the limitations and obstacles deep learning presents, and end this section describing the transfer learning technique. Afterwards, in the Technical framework, we will examine the CNN architectures that we used, review the datasets used, explain the system pipelines and the data transformation, and briefly mention the use of PyTorch. Next, in the Results section, the models will be analyzed and compared, verifying that the goals of the project have been accomplished. Finally, the sections of Conclusions and Future work will close the document, highlighting accomplishments, findings and future ideas.

2. STATE-OF-THE-ART

Even though nowadays the emotion recognition subject is not exclusive to research, it has not significantly expanded beyond the research environment and a small group of companies that are still investigating how to improve their own systems. Below, we will discuss the current developments of computer vision and machine learning for emotion recognition. In particular, we will review the state-of-the-art that is being accomplished by the research community. Then, we will showcase the few companies that currently work with emotion recognition. It is important to highlight that even though we will focus on the current state of emotion recognition through computer vision, there are other fields working on it analyzing speech, text, physiological information, … [12] [13]

2.1 Research works

The study and development of emotion recognition systems has been generating interest thanks to machine learning. It is important to remember that this field is not confined to visual tasks, there are systems that employ audio or physiological data too. Nonetheless, the recent rise of deep learning for computer vision (and many other fields) poses the question of how can said systems be substantially improved with the use of neural network methods, a notion which is currently gaining traction in the research community. Many advancements have come from the Audio-Visual Emotion Challenge (AVEC) [14], a competition that started in 2011 for comparing multimedia processing and machine learning techniques for emotion recognition. These techniques use video, audio, physiological data and/or metadata.

For our part, two research projects constitute the main focus of this section given that they present state-of-the-art level systems for this topic. These works we are analyzing propose machine learning techniques (deep learning models, among others) to surpass the state-of-the-art on emotion recognition systems using facial features. Also, both of them follow the continuous emotion classification model of valence and arousal (in contrast to the use of the discrete models). Another common factor between the two papers is their central focus on datasets originated in the wild (non-laboratory settings), although there are testings on other types of datasets as well.

The first paper reviewed is “AFEW-VA database for valence and arousal estimation in-the-wild” of 2017 [7]. They used various machine learning methods to test the task of emotion recognition using facial features with their own dataset. Their main drive was the almost exclusive widespread of datasets created at laboratory settings for the task of emotion recognition on the valence and arousal continuous model of human affect estimation. They anticipated that the machine learning models trained from the available laboratory setting datasets would not work satisfactorily with instances occurred in the wild, where the real world settings would not be ideal, which is much more akin with the requirements given if this technology is translated to computers, smartphones and machines of general use. This is the reason why they decided to create their own dataset, called AFEW-VA [8], which is composed of frames from movie extracts where actresses and actors display naturalistic emotions. The dataset follows the continuous model of valence and arousal because, as the researchers argue, the established discrete models of (considered) basic emotions did not capture the frequent subtleties and ambiguities of human emotion. The papers presents (then) recently published methods for continuous emotion classification (Regression, SVM, random forest, …) using video, audio, physiological and metadata modalities, or combinations of them. The machine learning models were tested on both the AFEW-VA dataset that they created and the popular laboratory setting SEMAINE dataset [15]. As they write, their findings showed that: “methods that work well in controlled environments do not necessarily perform as well in unconstrained conditions” (p.8).

The results of each method are separated depending on AFEW-VA or SEMAINE dataset, then on whether it is from arousal or valence dimensions. Each dimension is evaluated by three performance measures: Root Mean Square Error (RMSE), Pearson Correlation Coefficient (COR) and Intra-class Correlation Coefficient (ICC). In the case of the AFEW dataset, the lowest RMSE for both valence and arousal is achieved by Multiple Kernel Learning (2.639 for valence, 2.229 for arousal); Tree-based Random Forest achieves the highest PCC for both valence and arousal (0.407 for valence, 0.45 for arousal); and the best ICC is given by Support Vector Machine for Regression for both valence and arousal (0.29 for valence, 0.356 for arousal). For the SEMAINE results, we observe the following results, Tree-based Random Forest gets the lowest RMSE for valence (2.087) and a finetuned CNN obtains the lowest RMSE for arousal (1.608); Support Vector Machine for Regression gets the best PCC for both valence and arousal (0.35 for valence, 0.272 for arousal); Support Vector Machine for

Regression obtains the best ICC for valence (0.331) and Conditional Random Field gets the best ICC for arousal (0.245).

The motivation of the researchers to use deep learning came from the fact that just a few of the methods they reviewed contained neural networks, and noted that these were combined with other machine learning methods, no reviewed method published used neural networks alone. For the research work they developed two convolutional neural networks, one trained from scratch and the other finetuned (concept explained in Transfer learning, section 3.5) using AlexNet [16]. We can observe that the deep learning models did not obtain the best results in none of the cases, except the finetuned CNN with the lowest RMSE for arousal in SEMAINE. The researchers suggest that the CNN trained from scratch could have been limited by the few samples of the AFEW-VA, while the results of the finetuned one occur due to insufficient resolutions of the resized images (the images where cropped and then resized to be used as inputs) or because of the differences between the datasets and ImageNet, the dataset used to train AlexNet, which could have result in limitations on transfer learning

The second paper is “Deep Affect Prediction in-the-Wild: Aff-Wild Database and Challenge, Deep Architectures, and Beyond” published in 2019 [17]. In a similar manner to the aforementioned paper, they constructed their own dataset in response to the laboratory setting datasets that did not take into account the complexity of real world settings. It goes without saying that their focus follows the continuous model of valence and arousal since current research on emotion recognition makes use of the continuous model, putting aside the discrete models. Furthermore, their dataset is a response to the limitations in available continuous valence and arousal in the wild datasets, such as AFEW-VA (they argue it is very limited with 30000 frames; the popular laboratory setting RECOLA dataset [18] has 340000 frames and their own one has 1.2 million frames). Their dataset comes from Youtube reaction videos. They also organized the First Affect-in-the-wild Challenge, but comment that the submitted models had underwhelming results (although the benchmark was passed). Due to this reason, the researchers were motivated to develop their own models to challenge the best results. In this research work, all the models were neural networks, with no consideration for other machine learning techniques. This fact illustrates the current embrace of deep learning for emotion recognition.

The deep learning models were constructed according to a different approach: CNN (Convolutional Neural Network) or CNN and RNN (Recurrent Neural Network). In the case of the CNN architecture, a model was based on ResNet-50, another on VGG-16 and another on VGG-Face. The best results were achieved by using a configuration of a batch size of 50 and a learning rate of 0.0001. For the CNN-RNN case, the previous architectures (ResNet50 or VGG-Face) were used for the CNN part, while LSTM (Long Short-Term Memory) or GRUs (Gated Recurrent Units) were used for the RNN part. Configuration of sequence length of 80 and batch size of 4 for the best results. Their best performer was the CNN-RNN with GRUs (instead of LSTM), they name it AffWildNet. It was the better architecture in both valence and arousal prediction for the tested performance measures (the same used at the First Affect-in-the-wild Challenge). It achieved a CCC (Concordance Correlation Coefficient) for valence of 0.57 and a CCC for arousal of 0.43; an MSE for valence of 0.08 (0.2828 as RMSE) and an MSE for arousal of 0.06 (0.2449 as RMSE).

Lastly, the AffWildNet model is tested on other datasets: RECOLA and AFEW-VA. The model is finetuned and retrained for both datasets, with great results. The AFEW-VA results are compared by the PCC measure, the finetuned model obtains a PCC for valence of 0.514 and for arousal of 0.575, which surpasses the best PCC results from the 2017 paper previously seen. They note that with proper finetuning and retraining, their proposed AffWildNet can tackle state-of-the-art performances for other datasets.

2.2 Companies, APIs

Affectiva is among the companies that offer services based on affective computing. It works with facial, vocal and physiological information, through the use of deep learning, to analyze emotional states. The SDK they offer can be used for analytics on advertising and marketing campaigns. Also, it uses its technology to detect the emotional states of the drivers and passengers inside automobiles. In a similar way, the company Eyeris is also focusing on the experience inside cars, employing deep learning to construct their systems for emotion recognition. Visage Technologies uses machine learning for emotion recognition, which can be applied to analysis on marketing or user product experience. NVISO is a company that provides real-time API for emotion recognition based on deep learning. Microsoft has a facial emotion recognition API as part of its Project Oxford set of APIs. Face++ also offers API and

SDK services for emotion recognition. Other APIs for emotion recognition: ParallelDots's APIs, Nodus's Face Reader, CrowdEmotion API, Imotions API, SkyBiometry's API, … [19]

These companies and platforms use different types of "basic" emotions classification approaches, which we have seen are being put aside in current research works for being limited in the task of detecting complex emotional states. Moreover, most of these offer emotion recognition in addition to many other services (face detection, face analysis, object detection, …), so their efforts are not exclusive to the topic being commented here. However, we can notice the use of machine learning, most notably deep learning, to construct these systems.

Emotion recognition is also being incorporated to big companies. We can find important entertainment companies investigating it. Disney is employing facial expression recognition to predict how audiences react to movies [20]. They developed a neural network system that could say when a moviegoer would smile or laugh. The developers suspect that the technology can be trained for other emotions as well. Besides that, Apple bought Emotient, a company for facial emotion recognition, back in 2016 [21].

We find that deep learning is being employed by specialized companies and startups for their emotion recognition systems and services, although their products have not transcended to the general public. These products are mainly used by other companies for analyzing user experience, marketing campaigns and advertising. A few big companies have showed interest in these technologies. Certainly, emotion recognition is still not part of products of mass consumption, but recent advancements and studies on deep learning, computer vision and affective computing are reducing that distance.

3. THEORETICAL FRAMEWORK

In the following section we will delve into the theoretical framework of this research project. Here we will present the technologies and concepts that we worked with to develop the emotion recognition systems. First, the general concepts of machine learning and deep learning are introduced (why they were chosen for the project); then we will take a closer look at neural networks (particularly CNNs), discussing their components, how they work, how to improve (train) them, and the overall obstacles of deep learning and CNNs. Lastly, we will discuss transfer learning.

3.1 Deep learning

First of all, we need to introduce machine learning, the domain to which deep learning belongs. Machine learning is the study that aims to produce systems capable of "inferring" answers to situations that they have not been explicitly programmed to solve [22]. Such systems are based on "experience", which means they, through the use of algorithms, find patterns on data examples of the problem they will have to deal with [23]. It is important to highlight that these systems are able of generalizing from the examples, meaning they can answer unseen cases. This circumstance allows developers to focus on the data and the algorithms (models and parameters). Currently, machine learning presence is expanding rapidly [24], with many high tech companies embracing it, putting effort in research and using it to develop new applications; likewise, many startups established around machine learning are being founded. Machine learning can be found in e-commerce, cyber security, personal assistants, climate prediction or medical imaging, among many others.

Deep learning is one of the many subfields of machine learning. It is based on artificial neural networks methods (in contrast to other machine learning techniques, such as linear and logistic regression, support vector machine, decision trees, k-nearest neighbours, …) [25]. An artificial neural network is inspired by the brain, it is composed of connections of layers of neurons (nodes) [26]. There is an input and an output layer; if there are more (the layers in the middle are known as hidden layers), it is called deep neural network (which inspired the name of this subfield). Hidden layers add complexity to the capabilities of the system. Values pass from the input layer to the output layer: these values are modified in the process due to the often various hidden layers operations when moving to the next layer. Each connection of nodes has its own weight, a value which affects the input for the next node; as well as a bias. The output layer value is used to make the prediction.

This subfield did not originated recently. In fact, we can find successful examples of artificial neural networks systems developed for computer vision from as early as the 80s [27]. The consistent problem was the time it took to train these systems, because the deep learning training processes require high computational power. For that reason, deep learning research was overlooked for many years, until the improvement of hardware due to advancements in GPUs. Deep learning is booming, its greatest achievements have occurred in this decade [28]. Research work has experienced an exponential growth over the last five years. The current popularity of deep learning can be attributed to the enhancements of hardware, as well as the fact that it can be applied to many fields (recommendation systems, speech recognition, playing games, …). In some specific tasks, there are systems based on deep learning that are already better than humans (superhuman visual pattern recognition was achieved in 2011 [29]).

3.1.1 Why deep learning?

There are quite a few answers why we are using deep learning in the project. It is argued that the sudden growth of computer vision is largely due to deep learning, which in turn is possible because of availability of GPUs and data. As commented before, high tech companies around the globe are using it to improve their products. Moreover, the main papers analyzed in this project use deep learning (although one paper uses other algorithms as well) with favorable results, so it seems interesting to use different architectures of deep learning to test and challenge the state-of-the-art. Nonetheless, even though deep learning seems suited for computer vision because it accomplishes systems with great accuracies, computer vision is not restricted to only deep learning; many other machine learning techniques are still being used for computer vision, especially in cases where the introduction of deep learning has limitations.

One of the reasons why deep learning could be used for this investigation is the datasets. The datasets are large enough (one has tens of thousands of data, while the other reaches more than a hundred thousand of data) to be able to apply transfer learning (as we will see in Transfer learning, section 3.5). In the case of a little dataset, a system (trained from scratch) is constraint to learn and find the crucial patterns from just a small set of samples, which makes it hard for the resultant system to face unseen data [30]. When the dataset to train a deep learning system is large, the performance far exceeds those of other machine learning techniques [31]. Another issue is that of the hardware to train the deep learning models. Even though the advancements of hardware have potentiated deep learning, it still needs considerable computational power. All the models constructed for the project have been trained in a high performance computing environment (the UPF's cluster [32]), because of its GPU capacity, over the course of several days. The fact that in case that some model hyperparameters had to be changed, because then the process would have to start over, it means that it would have taken many weeks if not for the cluster. Both the sufficient (for transfer learning) datasets sizes and the hardware availability have made it possible to apply deep learning for this computer vision project.

3.2 Neural networks and training

When constructing a neural network, initially it will not work adequately for the desired task. The neural network configuration, composed of all of its parameters (weights and biases), which will transform the input data of the system to a response value, is the central mechanism behind these prediction systems. Given that these parameters are not fitting for the task when the neural network is created, they will have to be refined so that the predictions are more correct [33]. This is achieved by training the system. In the following segments we will explain what it means to train a neural network and how it "learns from experience" to improve itself. We will talk of forward propagation and backpropagation, as well as gradient descent and epochs, basic concepts that are fundamental in deep learning. Naturally, for this project we had to make use of all of them (apart from other key concepts that will be explained when needed).

3.2.1 Forward propagation

First of all, a neural network is composed of a variable number of layers, these layers also contain a variable number of nodes (or neurons) that perform operations. Nodes from a given layer are connected to some (or all, in case of fully connected neural networks) of the nodes of its nearest layers (the previous one, the next one or both if this is a hidden layer). Each connection between nodes has a weight, a value which serves as measure of importance for a particular connection. In addition, we have to add the layer bias, a value that does not come from previous nodes and is shared with all of the nodes of the same layer.

First, we are going to see how hidden layers work, these layers make up the majority of the layers in the (deep) neural networks we will work with. In a feedforward neural network (such as the convolutional neural networks seen in this research project), nodes from a given layer apply an operation to the output values generated by the nodes from the previous layer that share connections with them. When a node i receives a value from a previous node j as input, it applies equation (3.1), where M are the nodes from the previous layer that node i shares a connection with, $w_{i\,j}$ is the weight from the previous node j to node i, a_j^{l-1} is the activation result from that previous node and b^l is the bias of layer l. The resulting value is used to calculate the output using an activation function; although historically the common choice was the sigmoid function, the architectures we developed use ReLU as activation function, equation (3.2), which is widely used nowadays. In figure 3.1 we can see how these functions affect input values. This is the output of the node, and is passed to the next node or nodes, so that they can compute the operation with their own weight and bias parameters.

$$z_i^l = \sum_{j \epsilon M} (w_{i\,j}\, a_j^{l-1}) + b^l \tag{3.1}$$

$$a_i^l = max(0, z_i^l) \tag{3.2}$$

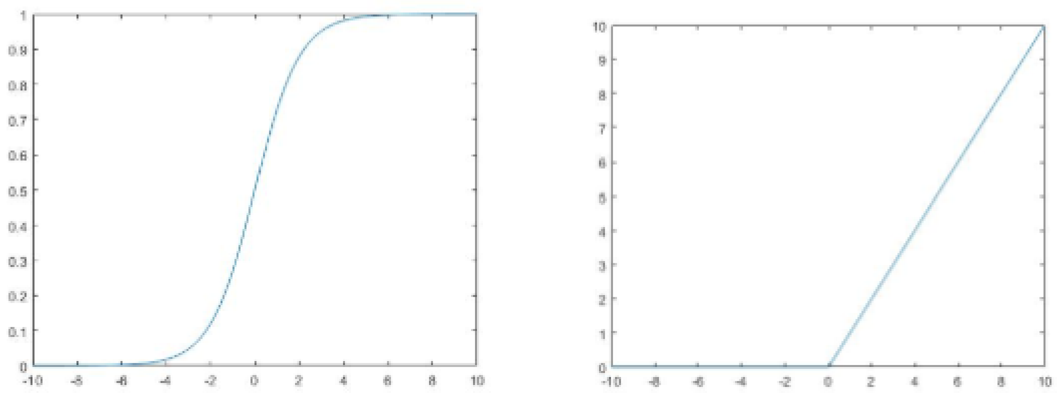

Figure 3.1: Sigmoid (on the left) and ReLU (on the right) functions

For their part, the input and output layers function somewhat differently. The input layer nodes just contain the input data of the system (for us this is the pixel information of the video frames) and pass these values to the first hidden layer so that its nodes can perform their calculations; while, in our case, the output layer nodes only compute equation (3.1), being this result the final prediction value or values (the activation function we employed for these nodes is the identity function, where the output is equal to the input, the same effect as not using any activation function, because the outputs are real numbers for the continuous, positive-negative type of answer system). The number of output layer nodes is equal to the number of classes that the system has to predict, for instance, in our case the number of classes to predict is two, one for valence and the other for arousal. If we were not in the training process of the system, then feedforward propagation alone would provide the system answers.

3.2.2 Backpropagation

Since we are training the system, another step will follow the output of the prediction value by the output layer. This step consists in computing the difference between the prediction value and the actual value of the samples. That difference is called the error or cost of the neural network. We are training on the training dataset samples, which means we have the actual labels values that we want the system to predict. We want to minimize that loss, so that the system is better at predicting any data related to the task, including other data outside the training dataset. This error will be propagated back so that the parameters can be readjusted. This method is called backpropagation.

The loss is calculated using the cost function. In our case we applied the MSE, expressed in equation (3.3), where n is the number of different outputs, Y is the label value and $\hat{Y}$ is the predicted value. This cost is propagated back to the previous layers as an error signal delta. Each node will have a corresponding delta. The deltas of the output layer nodes are computed as expressed in equation (3.4), where a_i^O is the activation value of the node *i* of the output layer, y_i^O is its corresponding label value and $f'(z_i^O)$ is the derivative of the activation function $f(z_i^O)$. Next, the deltas of the nodes of the previous layers are computed as expressed in equation (3.5), on L = {u,v, …, w} being those nodes that receive an input from node *i;* this process is repeated for all nodes (except the output layer nodes), going backwards (which inspired the name of the process) from the output layer to the input layer. As we have seen, the deltas of the output layer are computed differently from the deltas of the other layers.

$$J = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{3.3}$$

$$\delta_i^O = (a_i^O - y_i^O)\, f'(z_i^O) \tag{3.4}$$

$$\delta_i^l = \sum_{k \epsilon L} (\delta_k^{l+1} w_{k\,i}) \; f'(z_i^l) \tag{3.5}$$

Once the nodes have their deltas, it is time to update the parameters. We are using the gradient descent optimization algorithm to update them. The update expression for weights is in equation (3.6), where $w_{i\,j}$ is the weight from node *j* to node *i*, α is the learning rate, δ_i^l is the delta of node *i* and a_j^{l-1} is the activation value from node *j* (the input value in case of the input layer). To update the biases, it is applied the expression in equation (3.7), where b_i^l is the layer bias, α is the learning rate and δ_i^l is the delta of node *i*. Once the weights and biases are updated, the neural network is better adjusted to make predictions because the error will have been minimized. The system will be improved thanks to the use of the dataset samples, since they help to generate the cost, which will be reduced; so, in a way, the system is learning to make better predictions from past experience.

$$w_{i\,j} \leftarrow w_{i\,j} - \alpha\, \delta_i^l\, a_j^{l-1} \tag{3.6}$$

$$b_i^l \leftarrow b_i^l - \alpha\, \delta_i^l \tag{3.7}$$

3.2.3 Gradient descent

We are going to explain why the optimization algorithm gradient descent works [34]. The general function for updating parameters (whether weights or biases) is in equation (3.8), where the partial derivative of the cost function with respect to the parameter θ_i can be expressed as $\Delta\theta_i$. This algorithm exploits the use of derivatives to look for global minimums. We can look for global minimum because the cost function has a convex form. The cost function is what has to be optimized (to minimize system errors). Basically, the parameters (weights and biases) will be updated in a way that they are closer to values that achieve a lower cost when applying forward propagation again.

$$\theta_i \leftarrow \theta_i - \alpha\, \frac{\partial}{\partial \theta_i} J(\theta_i) \tag{3.8}$$

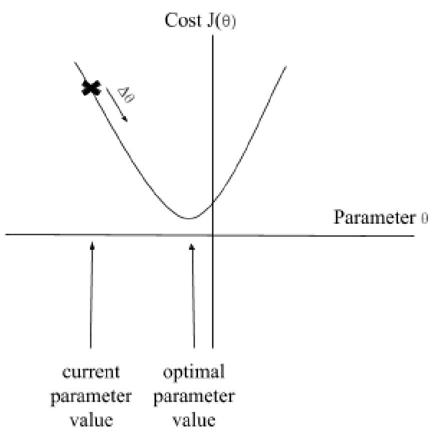

Figure 3.2: Case where the direction of the derivative of the cost function with respect to the parameter indicates the function is decreasing because the current value is at the left of the optimal one. When putting $\Delta\theta$ in the update function the parameter will be moved towards the right.

We can visualize (using figure 3.2) a plot of the cost function with respect to a given parameter variable. The plot curve would we convex, where the minimum cost value would correspond to a central value of the parameter variable. A way to achieve this central value is by using gradients. Recall that a derivative calculates the slope of a function (how much it is increasing or decreasing). If we use a simple two dimensional case to illustrate the problem, the resulting sign of this derivative tells us if the current parameter value generates a cost value that is on the left side of the minimum (if the derivative sign is negative, it means that the cost function is decreasing) or if it is on the right side of the minimum (if the derivative sign is positive, it means that the cost function is increasing). In the first case, the weight or bias will have to be increased when updating it to be closer to the minimum; in the second case, conversely, the weight or bias has to be decreased; the necessity of these opposite actions justify the minus sign in the derivative side of equation (3.8).

This will be an iterative process, in which the derivative will be able to direct the parameter to the direction of the minimum when updating. Also, the parameter will be changed minimally. We do not need great changes, because if this step is large, then the parameter value could continuously switch to the other side of the minimum in a never stopping procedure. It would never reach the optimal solution. The learning rate is the hyperparameter in the training process that determines how large will be the step of the parameters in the direction of the derivative when updating them. It can be a constant whose usual values are 0.1, 0.01 or 0.001, depending on the case. This hyperparameter forces the steps to be limited, the closer the parameter is to the solution, the more reduced the steps will be, given that the derivative value will also be lower when approaching the minimum (the derivative value is zero for the minimum cost; while the further is it from the minimum, the higher it is as an absolute value).

3.2.4 Epochs

As said before, this is an iterative process: first forward propagation, next backpropagation, then forward and backpropagation again, and so on… Each pass of both a forward propagation and the posterior backpropagation is called an epoch. The system will have to be trained on several epochs because the gradient descent algorithm will not instantaneously obtain the optimal parameter solutions, instead, it will situate the weights and biases a step closer to their solutions.

The optimization algorithm will update the parameter so that it follows the direction to the minimum. Nonetheless, each step is limited by the learning rate (needed to solve any issues of divergence of the solution). Furthermore, the closer the parameters are to the minimum, the lower each step will be. This means we will have to perform forward propagation and backpropagation several times so that the parameters are close enough to the optimal solutions. The number of epochs needed is indeterminate, it vastly varies with each training dataset, neural network architecture, hyperparameters. What is more, two systems with the same training dataset, architecture and hyperparameters will not generate the same configuration values even after the same number of epochs because the configuration values are usually randomly initialized. This fact provokes that the rate of convergence to the solutions when training will not be the same.

In our case, we trained the neural networks looking at the error values (and the evolution of the other metrics, as well), not at the number of epochs passed. If the error value converged and was low enough we could satisfactorily stop the training process. Stopping the training at a point of observable convergence is more appropriate and practical than letting it train more time for infinitesimal reductions of the error, since this process is so time-consuming (depending on the case, several days or several weeks) and so computationally demanding. What is more, letting it train more than it needs can lead to overfitting. Also, it is important to note that our architectures, because they used pretrained models, had their configurations values predetermined, except those of the added output layer (to match the desired output of valence and arousal not contemplated by the pretrained models), which had randomly initialized parameters.

Finally, for our architectures we employed mini-batch gradient descent [35], a special implementation of the gradient descent algorithm. This implementation performs the gradient descent updates using a set of samples of the training dataset, instead of using the whole training dataset, which would require unfeasible computation power. The length of this set is determined by a hyperparameter called batch size. The amount of training data divided by the batch size produces the numbers of batches, which is equal to the number of iterations needed to complete an epoch. For each model designed, after determining the architecture and training dataset, we had to adjust the batch size depending on the available GPU of the hardware. If the batch size for a given case was bigger than the GPU allowed, then the training process would crash, and a lower batch size would have to be used.

In addition to mini-batch gradient descent we also applied momentum [36], this technique consists on including past gradients with the currently computed gradient (both weighted according to the momentum value) when calculating a new step. For a given parameter θ_i, first we compute equation (3.9), where β is the momentum value and $V_{T-1}$is the previous V value for parameter θ_i. V_T is then used in equation (3.8) instead of the partial derivative of the cost function. This technique aims to cancel out oscillations because each parameter update, apart from getting closer to the optimal solution, carries noise in a certain additional direction with a variance. Two consecutive steps have opposite positions on this other direction, so in a way both would balance each other. This technique speeds up gradient descent. It has a value (another hyperparameter), whose usual value is 0.9 (using 0.9 gives more importance to past gradients than to the current one).

$$V_T = \beta\, V_{T-1} + (1-\beta)\, \frac{\partial}{\partial \theta_i} J(\theta_i) \tag{3.9}$$

3.3 Convolutional neural networks

Hereafter, we will explain about the type of deep learning neural networks that we worked with: convolutional neural networks (CNNs). CNNs are usually applied to computer vision problems (although they are also commonly used for natural language processing tasks). This type of neural network applies certain filters to the input data [37]. The necessity of this architecture comes from the fact that the data that goes through the many neural network layers come from images (even if originally they were in video format, it is still treated as frames) holding a lot of information, and consequently the training process would take so long it would not be practical to train it. Not only that but these filters also help in keeping just the most relevant parts of the images, so that the system is better trained to generalize the recognition task to unseen data. That is to say, the images size has to be significantly reduced, while still possessing the main information of the key images features. CNNs attempt to solve this issue by providing these filtering steps (in addition to other components), which makes them ideal for computer vision. In figure 3.3 we can see the components for a CNN model.

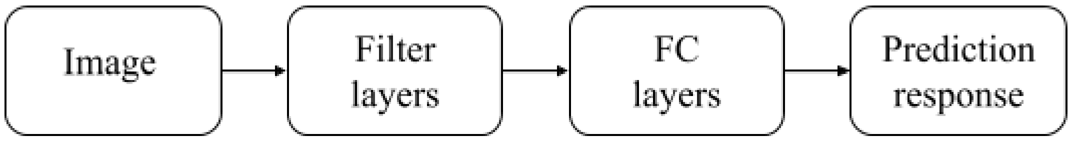

Figure 3.3: Diagram of the basic components for a CNN.

In a CNN the data goes in a forward direction from the input layer to the output layer, there are no cycles. As with any neural network, it is speculated that the connection of many layers results in the recognition of more abstract information based on the recognition of simpler information put together. The topic of the number of hidden layers will be discussed later when talking about specific CNN architectures, since this topic is common to all neural networks. The crucial components of a convolutional neural network are convolution and pooling operations (as layers) [38], in addition to a traditional feedforward neural network:

- Convolution: A kernel, a matrix of smaller size than the input (the image pixels), moves along the whole image (never placing the kernel outside of it), with each of its kernel values multiplied by the equivalently positioned input values at the moment. The size of the kernel is N by N, it has a center and will affect the N by N values surrounding the image position it is placed at (including this position too). All the multiplication results at a given moment are added, this is now the value of the corresponding position in the image (where the kernel center was placed). This process is repeated until the kernel has affected all the image.

- Pooling. Similar to convolution, a kernel of size N by N is placed over the input values. There are different types of pooling operations. The most common are max pooling and average pooling. The former consists in just keeping the highest number of the values that are surrounded by the kernel at every given time, while the latter consists in averaging the values surrounded by the kernel, replacing those values by this result. A common kernel is of size 2x2, which reduces 75 % of the image data.

- Nonlinearity. This layer adds nonlinearity to the neural network, which means adding the capacity to learn to solve more complex tasks. There are several, one of them is ReLU (Rectified Linear Unit), although there are other options. This operation usually comes after every convolution layer. Nonetheless, nonlinearity is also employed in the traditional neural network layers, as we have already seen, when using the activation functions to compute the output of each node.

- Fully connected (FC) layers. It is the traditional feedforward neural network where every node is connected to every other node of the next layer by a certain weight for each individual connection (plus a certain bias from layer to layer). The resulting image values from the previous filters are flatten as a column vector and passed to this neural network. Then the computations explained in Forward propagation (section 3.2.1) for the hidden layers are performed. The last layer produces the final prediction values. There are many functions, depending on the case, that can be chosen to produce this final value (in our case the identity function, others use ReLU, sigmoid, tanh or softmax).

In summary, the data (pixels of an image) size will be progressively reduced by the filter layers while keeping important features, and then the data will go through the fully connected layers so that in each step the system learns to identify more and more abstract information until reaching to a global recognition of the picture. The types, number and combinations of layers can vary. Because neural networks internal processes are not really understood by the researchers, so many architectures have been introduced with different results, and many more are being published. It was thought that the higher number of fully connected layers, the more complex tasks the neural network is capable of tackling; although some architectures propose reducing the number of layers. There is an ongoing debate regarding these issues. The architecture of FC layers has to be considered alongside the one of the filter layers, given that these layers reduce the data and remove irrelevant information.

3.4 Limitations and obstacles

As the research community points out and its growing use by big companies and start-ups seems to denote, deep learning holds potential for developing effective systems in many fields, not just computer vision. Many successful systems have emerged from this machine learning area. However, it is also true that this area faces some serious limitations and obstacles. One of them is the problem of overfitting. Just to briefly name other issues previously mentioned in this document, we will also talk of the time and hardware dependencies, the necessity of huge amounts of data or the existence of other system component constraints. Furthermore, we will also mention the problems that arise with the data annotations for a matter as subjective as the evaluation of emotions.

We have already commented that training such systems can take from some days to some weeks. When applying machine learning to design a system, it is important to consider the time that deep learning models take to be trained in spite of the usual effectiveness of these systems; because in some cases it might be preferable to user other machine learning techniques [31]. Another important requirement of deep learning is the hardware due to the high computational use it demands. So the use of GPUs has to be considered too for these systems.

For this project, we had to accept the time and hardware conditions and use deep learning because of two main reasons: most of the state-of-the-art in emotion recognition (that we have reviewed with the two aforementioned papers) has been accomplished using deep learning and the fact that current general computer vision development as well as recent achievements in this fields that can be attributed to deep learning. Regarding the time, one fold (from a five-fold cross validation) of the models that worked with the AFEW-VA dataset took almost a week to reach a convergence of their results, while in the case of the ones that used the SEMAINE dataset (we used a subset four times larger than AFEW-VA) it took more than a week for the results of one fold for their metrics to converge. As for the hardware, we trained the models on the UPF's cluster, which allowed us to set batch sizes of between 20 and 50 (depending on the architecture) to perform mini batch gradient descent.

Large datasets are usually needed to train complex deep learning systems. The lack of sufficiently large datasets for various complex tasks is one of the obstacles that researchers expect to become more important in the future of deep learning. As noted in the papers, AFEW-VA is not a good choice to train from scratch [7]. An amount of 30000 images is not considered large enough for our task, which is also complex. To solve this we employed transfer learning, given that the pretrained models that we used were trained on a dataset of more than a million images; so that we could retrain using AFEW-VA to refine the parameters for our own task.

Lastly, we have to comment constraints that arise not from the use of deep learning but by other system components that are fundamental for the given task. For one side, the recording equipment has to successfully capture the facial features of the user. Otherwise, the system would fail to respond correctly, be it for a low definition capture or for other camera issues. There have been cases of visual recognition systems (specifically identification of people) which failed to prove effective for various purposes [39], sometime because of issues with the recording. In our case, we used the system with a camera that properly captures the facial features of the user. For the other side, we have to consider the environment and the subject's situation when the system is working. The variations of illumination and the diversity of sceneries have been addressed by the use of AFEW-VA, which is a strong dataset in that regard. Also, the rotation of the head is an important issue, which is also taken into account with AFEW-VA, although there is a limit (too much angle of rotation and the system will not capture the user facial features).

3.4.1 Overfitting

When developing a machine learning prediction system, one of the most common issues that can occur is that of overfitting. In simple terms, a model is said to suffer from overfitting when it is great at predicting the training data, so much so that it cannot generalize well to other data. A system with overfitting is not useful, given that our objective when developing machine learning systems is to predict unseen data. There are some ways to detect and to prevent overfitting [40]:

- Cross-validation: The training dataset is split in k folds, so that k models will be trained in the end. Each model will be trained on a different grouping of the k-1 folds, and will be validated on the remaining one (no model repeating the combinations of training-validation folds). In a cross validation method called k-fold cross validation each of the k models are trained using k-1 folds, and the other one is used to validate them. Then the resulting performance measures of all the models are averaged.

- Dropout: This technique consists in removing certain nodes (and their connections) at each iteration when training. The number of nodes removed depends on a dropout rate probability. This means in each iteration the amount of neural network nodes will vary.

These are methods to face overfitting that we encountered while working on the models for the project. However, there are many other methods to confront this issue, such as data augmentation, regularization L1, regularization L2, early stopping, ...

3.4.2 Annotations

The data annotations pose another issue. The data instances, each showing the emotional state from the image or video individual, has to be associated with a label indicating an annotated value. In our case, each video frame has two corresponding labels, one for the valence value and the other for the arousal one. These labels are given by a professional annotator, when they analyze the data. The issue comes from the fact that in our case the data comes from emotions, which is a subjective matter, meaning that the valence and arousal annotations for the same data instance may highly differ among annotators, even professional ones.

As we have already mentioned, frequently it is not clear to us how someone else is feeling at a given moment, human emotions are a complex subject. Because the criterion with which emotions are evaluated is not the same from person to person, different proposals have been thought to tackle this issue, such as averaging the annotations of many annotators for the same instances. One study [41] proposes the consideration of discrete ranges in the space of valence and arousal, the ranges lengths are specific to every annotator. The annotators share the same number of different discrete ranges, so that then their annotations can be fused with the other annotators' corresponding discrete ranges.

The AFEW-VA dataset only comes with an annotation for valence and another for arousal for every frame, so we could not perform any modification. In the case of the SEMAINE dataset we had up to five different valence and arousal annotations for every frame, our only choice was to average them. In future work, it will be important to consider the importance on the annotation strategies that follow the datasets that we chose because this issue can undermine the system performance.

3.5 Transfer learning

Transfer learning is a method commonly used nowadays in machine learning to highly accelerate the training process by taking advantage of other systems. The idea of this method comes from the fact that humans do not learn from scratch, there is always previous knowledge which is treated as the basis for the understanding of new concepts and tasks. For their part, machine learning algorithms used to exclusively be isolated mechanisms, in the sense that these systems, such as neural networks, were entirely built and trained for a concrete task, with no consideration of similar already existing systems which may have possessed a considerable amount of useful understanding for the new purpose. Transfer learning tries to imitate the human process of using prior knowledge to learn [42]. It is a technique that redefines pre-existent systems that were designed for a given task (and domain) so that it can be used for a different objective. Its conveniences, which we will later see, have made it a popular choice in deep learning.

Primarily, it makes use of pretrained models. These models are usually designed for complex datasets, datasets with lots of data (for example, a popular one is the ImageNet dataset [43], which contains millions of images for a 1000-class classification task). They are developed by researchers and are published for others to use if accuracy results are satisfactory. Clearly, training neural networks on such datasets takes a long time (a few weeks). Nonetheless, said neural networks have been tested with great results for their tasks [44], which compels us to think how they could be beneficial for our own systems, taking into account that deep learning processes, as well as machine learning ones in general, have a great limitation by being task specific. The problem of task specificity prevents machine learning from being a more powerful tool, because any of these systems is only capable of tackling a very specific task [45]. In a way, transfer earning can be seen as a starting point to face this issue.

It is a fairly simple process to conduct [46]. First, a suitable pretrained model is chosen, considering which of the published ones can be better adjusted for the new task and domain in question (how similar is the dataset they were trained for). This pretrained model is initialized, which means that the starting weights and biases values will be predetermined. Then, both the input layer and the output layer have to be replaced: The new input layer must be able to receive the new data dimensions (if different) of the input data (the pixels in case of images); the number of outputs from the output layer has to coincide with the new number (if different) of classes for the new classification task. The parameters of the output layer will be randomly initialized (or initialized at zero) if the number of classes is different (which is our case). Next, a decision has to be made: using finetuning or feature extraction. Deciding between them depends on the size of the new dataset and its similarity to the one used to train the pretrained model [42]. Finetuning consists in updating all the neural network weights when retraining the pretrained model. It is suited for cases in which we have datasets of considerable length (such as AFEW-VA), since the whole system could be retrained to obtain better results because there would be less risk of overfitting. Feature extraction, instead, just updates the weights that belong to the output layer. It is used when the new dataset is quite short, because in that case, the whole system would be retrained from a few samples: such a system would most likely suffer issues of overfitting.

It works because of how neural networks function. It is argued that the first layers take care of the recognition of low level features (in a computer vision problem, these would be edges, colors, shades, ...). As said in Convolutional neural network (section 3.3), the connection of layers adds higher levels of abstraction until finally reaching a global understanding of the given data. This means that the initial layers could be easily shared for other cases within the same field (computer vision, NLP, …) because of the recognition of basic features common to any image, while the reusability of latter layers depends progressively on the similarity between the domain of the original task and the one of the new task. To name a case, a model which classifies dogs and cats will be more adequate and will suffer less internal changes when used to employ transfer learning for a task of classification of other animals than for the task of classifying vehicles. The domain is more similar in the first case, the progressive recognition of general features and then the distinct particularities of the new animals can be better extrapolated to the animal case because they share many more similarities with other animals than with vehicles. However, we have to note that these are speculations because we do not really know how neural networks work internally. So it is important to choose pretrained models that have been used for similar domains as the ones of the new models. It has to be emphasized that using finetuning for a case of similar datasets will (slightly, if they are very similar) redefine the weights making the most of the similarities for both tasks, it does not completely alters or substitutes the values (if that was the case, which is not, then it would be as good as training it from scratch).

This technique presents many advantages. One of them if speeding up the process of training models, because if the domain of the new task is in a way similar to the new one the capabilities of recognizing many features can be transferred. If compared with a process which does not use it, it can occur that adding transfer learning produces [47]: a better initial performance, a faster convergence to best possible accuracy while being trained and/or higher best possible accuracy. Also, the researchers of [7] developed two deep neural networks, one using transfer learning, the other without it; the results were better for the one with transfer learning. They commented that training a neural network from scratch with such a dataset does not provide good results. Furthermore, this method is useful in case of not having large datasets or in case of, although having a large dataset, not having adequate hardware or enough time to train. Many research groups present pretrained models with great accuracies for complex tasks. Thanks to transfer learning we can readapt their purposes for new tasks.

4. TECHNICAL FRAMEWORK

In this section we will describe the more technical aspects of the project, starting with the CNN architectures that we employed, then we will describe the datasets used to train and test the models and also explain the system pipelines (for training and for the real-time app). A brief description of PyTorch is also included.

4.1 Architectures

For the project we decided to perform transfer learning, which is why we employed pretrained models, each with its own configurations. Various CNN architectures have been published during this decade for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [48], a challenge in which participants (research groups) develop a system to recognize entities (persons, animals, tools, vehicles, …, up to 1000 classes) from a 1.2 million images dataset. Many successful models have originated for that challenge over the course of the decade [44]: AlexNet, Inception (Google) [50], VGG (Oxford), ResNet (Microsoft), … These architectures (and posterior versions of them) became state-of-the-art when they were released and have been fundamental in computer vision deep learning research, VGG and ResNet are the ones we chose as pretrained models. These two are some of the last game changers in CNN architectures, and are still being employed by the deep learning community. Although new architectures and modifications on existent ones have been developed, no truly groundbreaking improvement have been achieved in the last four years.

We will first point out some aspects that are common to both architectures. First, the input image dimension has to be 224 x 224 (which is why we resize to this dimension in the preprocessing pipeline) with 3 channels (because of RGB). Also, these architectures were designed to work on a prediction task of 1000 classes, which requires us to change this output layer of 1000 nodes by an output layer of two nodes, to predict valence and arousal. This means that the configuration parameters of both will be predetermined when initializing them; except the parameters corresponding to the added output layer, whose parameters will be randomly initialized. Also, no activation function is added to this output layer (as if we were using the identity function as activation), because the output values have to be real numbers. Finetuning is performed, so all the model parameters will be refined when training.

4.1.1 VGG

VGG [11] is a CNN architecture developed by the Visual Geometry Group of the University of Oxford. It was presented at the ILSVRC of 2014, where it won the 2nd place. VGG has a straightforward design that is usually used as base for more complex architectures (such as ResNet). A problem with this architecture is that it works with many parameters (more than 130 million on any of its variations) and it trains slower compared to other designs. Some variations on the base VGG architecture where designed, with different number of layers: VGG13, VGG16, VGG19, ...

We chose VGG11. The following is its architecture with the modification on the output layer: It receives the 224 x 224 RGB values. Filter layers (where each 3x3 convolution is followed by a batch normalization [49] and a ReLU operation before going to the next filter): 3x3 convolution, 2x2 max pooling, 3x3 convolution, 2x2 max pooling, two consecutive 3x3 convolution, 2x2 max pooling, two consecutive 3x3 convolution, 2x2 max pooling, two consecutive 3x3 convolution, 2x2 max pooling. FC layers (their activation are ReLU, except the output layer whose activation can be thought as an identity function): FC (in 25088, out 4096), ReLU, dropout (p = 0.5), FC (in 4096, out 4096), ReLU, dropout (p = 0.5), FC (in 4096, out 2). The output are two float values.

Both AFEW-VA and SEMAINE where used to train different VGG11 models. Each dataset was separated in five folds, so that we could train a distinct VGG11 model on four folds (each model with a different combination of folds) and validate them in the remaining fold. Configurations: A batch size of 20, a learning rate of 0.001 and a momentum of 0.9. A trained VGG11 model is over 510 MB.

4.1.2 ResNet

The Residual Network (also known as ResNet) [10] was released by the Microsoft Research. This architecture won the 1st place on the ILSVRC of 2015. It makes use of skip connections. It has far fewer parameters (ResNet152 has 60 millions) than VGG and it also trains faster than the other architecture. Many variations on the base architecture were released: ResNet18, ResNet34, ResNet50, ...

We chose ResNet152. The following is its architecture with the modification on the output layer: It receives the 224 x 224 RGB values. Filter layers (all the convolutions are followed by batch normalization; the last convolution of each series is followed by ReLU after its batch normalization): 7x7 convolution, 3x3 max pooling, 50 consecutive series of a sequence [1x1 convolution, 3x3 convolution, 1x1 convolution] where the input of each of these sequences is also used as input (in addition with the output values from that whole sequence) to the following sequence (these are the skip connections), 7x7 average pool, FC (in 2048, out 2). The output are two float values.

Both AFEW-VA and SEMAINE where used to train different ResNet152 models. Each dataset was separated in five folds, so that we could train a distinct ResNet152 model on four folds (each model with a different combination of folds) and validate them in the remaining fold. Configurations: A batch size of 50, a learning rate of 0.001 and a momentum of 0.9. A trained ResNet152 model is over 230 MB.

4.2 Datasets

The continuous classification model leads us to the dataset, since there are many valence-arousal datasets to choose from. We need a dataset to feed to the machine learning algorithm so that the system is trained to learn to easily identify the affect of any person. Also, the more data it has the better, particularly if the dataset subjects are diverse, not just a small group of participants that are present in the majority of the recording sessions.

First, we have to decide another aspect: Whether the dataset emotions have to be posed or spontaneous. In the first case, participants are asked to exhibit some emotions; in the other case, emotions occur naturally. The problem with the posed ones is that they are not accurate because they are exaggerated and end up differing in appearance with the intended affect if compared with its spontaneous occurrence. Nonetheless, as the researchers of the KRISTINA Valence-Arousal Database pointed out [51], even the spontaneous ones can present inconveniences if they are originated at a laboratory setting, because of convenient (not naturalistic) illumination and pose; they also add that the laboratory recordings usually work with a low number of participants, which limits the predicting capacities of the system. So the dataset instances have to originate from non-laboratory settings.

One of the databases that integrates both the valence and arousal classification model and also contains instances that are spontaneous is the AFEW-VA [8]. It is not exactly the same as other spontaneous databases, it actually is part of the “afew” group. *Afew* is “acted facial expressions in-the-wild”, which means they are taken from multiple recordings where actresses and actors display several emotional states. In this way, the dataset contains naturalistic instances (spontaneous affect with a non-ideal illumination or pose). In particular, the dataset contains video frames from 600 movie extracts where different actresses and actors display various acted emotional states. Having different performers from different movies benefits the resulting system because it will learn from a variety faces in a variety of settings.

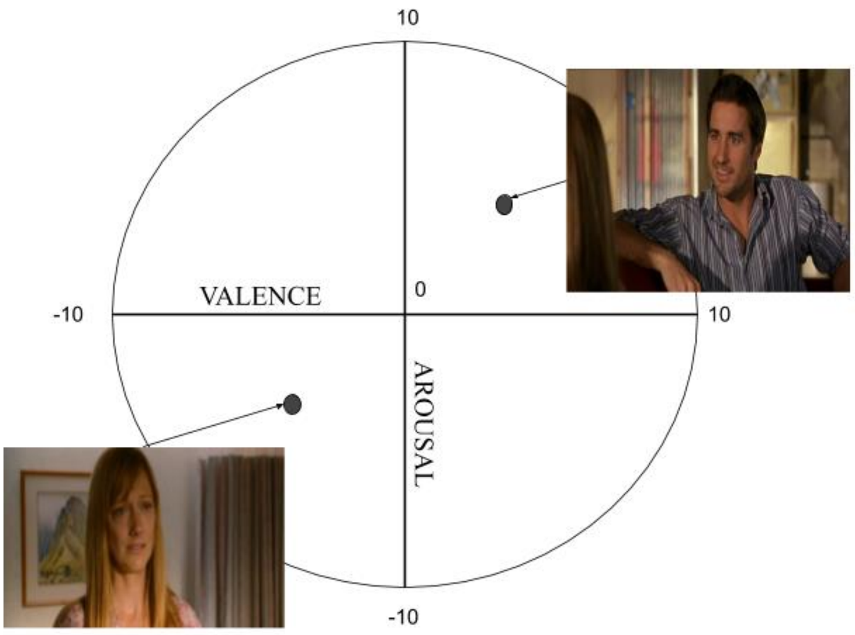

Figure 4.1: AFEW-VA sample frames placed in the dataset plane for valence and arousal. The image in the first quadrant has a valence of 3 and an arousal of 4; the image in the third quadrant has a valence of -4 and an arousal of -3.

The video clips come as frames (images). Each one has corresponding valence and arousal values (as we can see with some samples in figure 4.1), both in the range of [-10, 10]. Apart from the valence and arousal annotations, each frame has a 68 landmark point annotations (68 image position x-y pairs), which represent points in the image that indicate the contour of the face, as well as other facial points of interest (mouth, nose, eyes, eyebrows). These landmark points will be used to discriminate the image segment of the face, ignoring the rest of the image (because it is important that the system only learns from the face part), since the frame images also contain all the unnecessary scenery from the original movie scene.

This database comes from the “AFEW-VA database for valence and arousal estimation in-the-wild” paper [7] that we analysed before. The main interest behind this paper was to contribute with a database that could confront with the ones that originated in laboratory settings with acted instances, because, as it is argued in the paper, “spontaneous facial expressions are characterized by subtle, minimal facial deformations which are difficult to track, and frequent out-of-plane head movements whose effects are difficult to remove” (p.1). Their solution was to use extracts from movie clips, since professional actresses and actors naturally perform in a realistic way. For the time being, this seems to be the best proposal for making spontaneous databases, even though technically these clips are still acted by movie performers.

One of the major weaknesses of AFEW-VA is its size. It is comprised of around 30000 frames. In figure 4.2 we can see how it compares with popular datasets (RECOLA, ImageNet), the dataset is clearly not large enough for training models from scratch. For this reason, we performed transfer learning (using models pretrained on the ImageNet dataset), so that we could benefit from complex models that could be redefined for our own task with the AFEW-VA dataset.

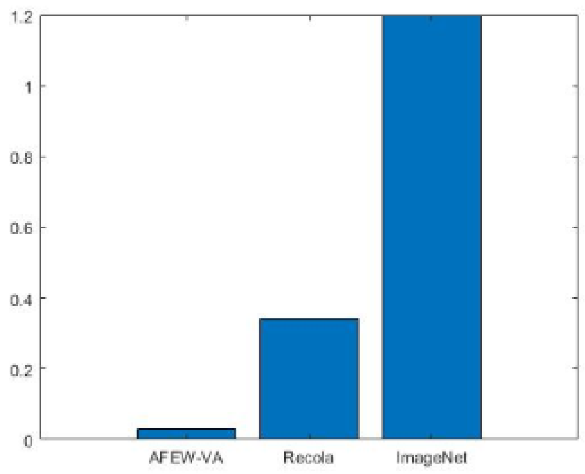

Figure 4.2: AFEW-VA size put into perspective with RECOLA and ImageNet (number of frames in millions).

Among other current research work that also aims to develop naturalistic and spontaneous datasets, some focus on YouTube clips from normal people (namely, reaction videos), as the other paper we also analyzed [17]. Its main problem is that some of YouTube reaction videos are thought to be faked or exaggerated for popularity-gaining purposes. Also, among YouTube reaction videos there tends to not be as much variety of emotions (most of them are joy, surprise, disgust and disappointment) as with movies. Another important point to make concerning this approach is that reacting to something (which can involve a certain expectancy level) does not really equals to other routine moments, such as having a conversation with someone, being by yourself performing a certain task or being mostly inactive. There has to be, of course, a deeper discussion and analysis on this kind of approach because of the questions it raises; which should remind us of the importance of working with investigation by psychologists and neuroscientists to advance in the study of emotion recognition systems. The aforementioned reasons give more confidence to the movie clips approach even if it is technically an acted-oriented approach.

Nonetheless, the reaction videos approach is interesting because this type of data is originated at a particular context, (mostly) watching a computer screen. This type of context can be well suited for some emotion recognition applications that could share such a context. Apart from that, this approach also inspires another strategy: constructing datasets with data from smartphones cameras (instead of computers), given that these data would have some particularities regarding head pose, head rotation, illumination, scenery. Systems trained on other type of datasets could suffer when trying to refine them for a smartphone application. If emotion recognition is to take off one of the most appealing implementations would be on smartphones, although this is a topic for future discussion.

We also worked with another dataset, the SEMAINE dataset [15]. This is a known dataset in the community of visual affective systems, it was published in 2007. Its data was originated at a laboratory setting, from recorded conversations between two individuals. It follows the valence and arousal classification system (both measures with a range of [-1, 1]). We worked with those recordings that provided valence and arousal annotations, generating frames from their videos. From those we had to use only 10 % of the data (keeping one frame for every ten) because the size was so large that it would have drastically affected the time to train the models. Since no facial landmark points were provided, we used a facial landmark generator to produce this information in the same manner of the AFEW-VA dataset (68 image position x-y pairs).

SEMAINE suffers from the previously stated issues of the datasets originated from a laboratory setting (invariant scenery, convenient illumination, head position). The researchers of the

AFEW-VA dataset also commented in their paper on SEMAINE, stating that it is not a versatile dataset because of the few different participants displaying emotions. They worked with SEMAINE to compare their models trained on AFEW-VA, we are also taking this strategy. In particular, we will test how a model trained on one dataset generalizes to the other one, expecting the ones trained on the afew dataset to be better because the complexity of emotions and settings that characterize this dataset can more easily adapt to data with ideal conditions.

4.3 Pipeline and data transformation

The emotion recognition system will receive input data and generate a corresponding response. This data will go through a pipeline, a sequence of steps that will affect the data, both external and internal to the neural network model that the system will be working on. Among these steps there are some that belong to the data preprocessing for better training the models, which in turn have an effect on the steps applied when testing the resultant systems. A crucial component of the designed pipeline is facial landmarking, we will start by explaining this concept.

4.3.1 Facial landmarking

Facial landmarking is a technique for tracking the facial points of interest (facial landmarks) of a given user. These points can indicate face contour as well as eyes, eyebrows, nose and mouth (as the software we use does). This is a computer vision technique that has been heavily studied through the use of machine learning algorithms to develop systems that can accurately predict these facial landmarks. Many facial landmarking softwares are available, these softwares work really well [52].

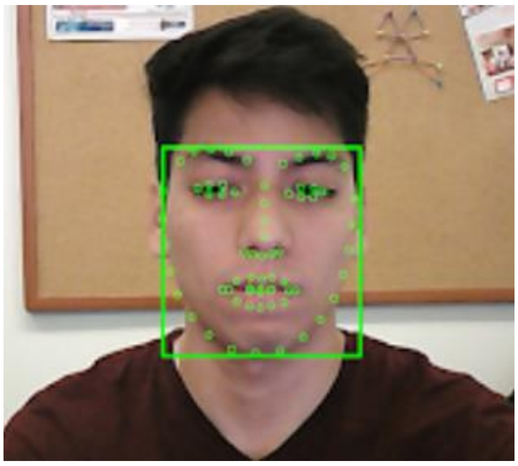

Figure 4.3: Software for tracking facial landmarks. A bounding box is also defined.

We employed a facial landmarking software to generate these points. The image or series of images (for videos) captured by a common camera can be passed to these systems so that they can produce these position points in the image (we could then paint these points to visualize its accuracy, as shown in figure 4.3). This process was added to the system pipeline, as we will see later, so that whenever a user uses the emotion recognition system, some of this facial information can be used to improve the system responses (facial landmarking was also employed when training the systems for the same reasons).

The AFEW-VA dataset already includes the positions of the facial landmarks in the frames, so we only had to save them, using them later in the pipeline when training with this dataset. For the SEMAINE dataset, we had to generate these points with a facial landmarking software for each of the frames we worked with. Similarly, when testing any of the emotion recognition systems we also employ it. We use these points to remove most of the image information external to the face (although minimum information of the scenery is kept). These allows the systems to be more comfortable to use since the facial landmarking software will focus on providing only the face information to the emotion recognition model; models that have been trained almost exclusively with face information and are therefore better at predicting the emotional values than models trained with pointless information (scenery information).

4.3.2 Pipeline for training

The data used to train the models has to go through a preprocessing pipeline before actually being fed to the designed neural networks. Preprocessing the data offers benefits for the model. The data transformation techniques that we will see down below are standard in machine learning. Particularly, these are of widespread use for computer vision deep learning systems. The purpose of this data transformation is to treat the data in a way that the model learns in a more sharpened manner, in the sense that the data we are working with could possess expendable information and getting rid of it can enhance the resulting configuration values (which are responsible for making the predictions); some of these transformations also aim to produce more similar data distributions. Such transformations can produce a faster training, for example. Not only that, this preprocessing also encompasses the data conversions into the needed formats, sizes and the data structure that this data is going to be manipulated in.

After having accessed the AFEW-VA dataset and stored all the images directory paths and their corresponding two types of annotations, the 68 x 2 facial landmark values and the 2 x 1 valence and arousal values, in three separate lists, the dataloader will access a determined set (depending on the batch size) of images and their corresponding annotations after each iteration in the training process.

Before the neural networks access the data generated by the dataloader, this data will have been transformed. Now we will comment each step of this preprocessing by looking at the general case of the transformations of one set of images from a given iteration, to illustrate this process:

- A group of images (determined by the batch size) is accessed. Each image is of size 720 x 576 in PNG format, and is accessed by looking for it with its stored data path. Each image contains an individual displaying an emotional state in a certain location (with a certain illumination, head pose and rotation angle of the head).

- A conversion to RGB mode is defined because the pretrained models that we use receive RGB values as input (although each image of this dataset is already in RGB mode). Each image pixel is represented by three numbers, each number ranging from 0 to 255.

- The facial landmark points of each image are accessed too. Using the most extreme landmark points (highest, lowest, leftmost and rightmost), a rectangular bounding box is defined around the area of the face (taking a bit of the scenery as well). This bounding box is used to crop the face, so most of the image information that is not the face is removed. In this way, the CNN will focus on the faces.

- After cropping the face area, each image size will have been reduced (the new size will vary with each case). It will then be resized to 224 x 224. In a similar way, the landmark will also be readjusted to the new 224 x 224 dimensions (just in case they were needed).

- The values of each image are passed to a tensor. The tensor of these images will be of batch size x 3 x 224 x 224 dimensions, where the second one is the number of channels (in RGB there are 3 channels). Likewise, the valence and arousal values for these images are stored in a tensor of batch size x 2 x 1 dimensions.

- Normalization of the RGB values of each image. Each image channel is normalized with a mean of 0.5 and a standard deviation of 0.5, which results in scaling each channel value to a [-1, 1] range. The purpose behind normalization is producing a faster convergence on gradient descent.

At this point the images values are ready to be passed to the convolutional neural networks. When the dataloader goes through each iteration while training the model, a tensor of batch size x 3 x 224 x 224 dimensions with its values completely transformed after the preprocessing will be used as input for the CNN. Although the set of images is passed at the same time, internally, the CNN will process each image separately. Once the neural network generates the output of batch size x 2 x 1 dimensions for the predicted valence and arousal values for the input images, the cost can be computed with the actual valence and arousal values stored in a tensor of labels that is also passed by the dataloader.

The pipeline for training on SEMAINE is almost identical to the one with AFEW-VA, the only difference is that SEMAINE does not provide facial landmark information for their recordings, so we had to employ a software to generate these points (68 x 2 points too), as we commented before. This step would be at the start of the pipeline, after having access to the dataset.

4.3.3 Pipeline for real-time app

For the resulting emotion recognition system the pipeline is pretty similar. The real-time app pipeline, in figure 4.4, is as follows. A camera (a computer webcam) captures the face of the user. The facial landmarking software tracks the facial points of interest and generates 68 x 2 landmark points. The landmark is used to crop the face area, then the image will be resized to 224 x 224. It is then passed to a tensor of 1 x 3 x 224 x 224 dimensions. Finally, the RGB values are normalised to a [-1, 1] range. The tensor is passed to the trained model so that it performs forward propagation. The system outputs a pair of values, the predicted valence and arousal for the image of the user captured by the camera at a given moment. The system will continuously be generating prediction values as long as a user is using the system and the facial landmarking software detects there is a face (so that the model can work with this information).

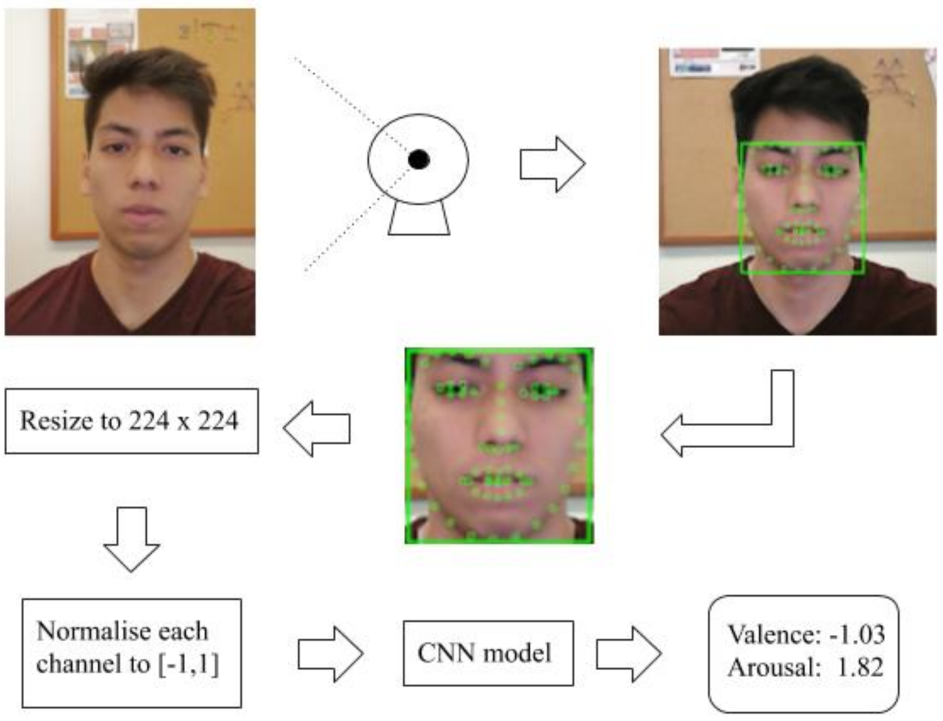

Figure 4.4: Real-time app pipeline.

The preprocessing steps of cropping, resizing and normalization are required when testing because they were used when training the models, which means the models parameters are specialized in working with data that has gone through the same transformations as the training data did. Not preprocessing the input data will cause the system to respond inaccurately.

4.4 PyTorch

PyTorch is a Python-based package for deep learning purposes which exploits the use of GPUs and is dynamic and fast [53]. It was developed by Facebook and was released in 2016. The package has two main high-level features that differentiates it from other deep learning frameworks (such as Keras, Theano or its strongest competitor, TensorFlow), as mentioned in its documentation: “Tensor computation with strong GPU acceleration and deep neural networks built on a tape-based autograd system”. A tensor is a type of data structure which can be thought as a multidimensional array, PyTorch uses tensors to contain data so that it can perform mathematical operations to it. The tape-based autograd system is a feature which consists in recording every occurring operation so that when going backwards (as in backpropagation), the gradients can be automatically calculated. PyTorch has the support of a growing community and it is currently one the preferred choices for working in deep learning research, although it lacks a strong product deployment support.

PyTorch was used in the project alongside Python 3 (it was designed to be effortlessly integrated with Python) to construct the neural networks, train and test them and evaluate the results. A great advantage of PyTorch is the fact that its code can be debugged easily, which was useful when trying different hyperparameters (such as the various tests of the maximum possible batch size for each architecture and for each dataset considering the utilized hardware) or for rearranging them when observing really poor initial training results; with no necessity of having to wait for the whole code to execute. It also has a simple mechanism to move from CPU to GPU and vice versa. Even though it has a large catalogue of tensor operations, the fact that Python is so easily integrated means that other libraries (such as NumPy, Matplotlib, ...) can be imported if necessary. Lastly, we can mention that it has many downloadable pretrained models (VGG, ResNet, Inception, AlexNet, …) to perform transfer learning, as we did.

5. RESULTS

5.1 Quantitative analysis

First of all, we are going to provide the mathematical definition of the performance measures that we employed to analyze and compare the models with the state-of-the-art. These measures are RMSE, PCC and ICC. It is important to remark, to prevent misunderstandings, that MSE was used as cost function and also to observe convergences, both while training with the training sets of the dataset. The performance measures that we used where employed for the validation step of the training to study the resulting models and compare them with the state-of-the-art models that we presented before, which are also evaluated in terms of some (if not all) of these measures (given that they are commonly used in machine learning).

RMSE is the Root-Mean Square Error where Y_i are the prediction values and $\hat{Y}_i$ are the label values. Lower is better.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{5.1}$$

PCC is the Pearson Correlation Coefficient, where COV is covariance and σ is standard deviation. Higher is better.

$$PCC = \frac{COV(\hat{Y}\ ,Y\ )}{\sigma_{\hat{Y}}\ \ \sigma_{Y}} \tag{5.2}$$

ICC is the Intra-class Correlation Coefficient. Higher is better.

$$ICC = \frac{2\,COV(\hat{Y}\ ,Y\ )}{\sigma_{\hat{Y}}^2 + \sigma_{Y}^2} \tag{5.3}$$

We can fully compare with the AFEW-VA paper [7] models because the researchers used these three performance measures, noting that both RMSE and PCC are widely used for estimating systems that predict valence and arousal levels. The Aff-Wild paper [17] uses MSE, PCC and CCC, so we can still compare with the first two measures.

5.1.1 Challenging the state-of-the-art

At this point we are going to comment the results regarding trying to reach the visual emotion recognition state-of-the-art. Since our focus is in in-the-wild datasets, first we will present results on the AFEW-VA dataset.

Table 1 depicts the PCC results of the 5-fold cross validation from our best performing architecture, which are the ResNet152 trained models, and of the AffWildNet architecture (presented by the "Deep Affect Prediction in-the-Wild: Aff-Wild Database and Challenge, Deep Architectures, and Beyond" study [17]), which holds the record on PCC on the AFEW-VA dataset.

Table 1: PCC results on arousal and valence for AffWildNet and for the 5-fold cross validation of the finetuned ResNet152 models.

Model	Arousal PCC	Valence PCC
AffWildNet [17]	**0.575**	**0.514**
Finetuned ResNet152	0.432	0.49

Although the PCC for arousal of AffWildNet is much higher than the one achieved by our model, the latter is pretty close to the other one in terms of PCC for valence. When checking the measures of the individual folds, we observed that a pair of them clearly reached the results from the state-of-the-art, while the other three where responsible for moving away from that mark when performing the 5-fold cross validation. Nonetheless, we consider that we are still really close to the desired results (particularly for valence estimation).

Because AffWildNet was not quantified in terms of RSME and ICC, we are going to compare our results on these measures with the best results presented by the "AFEW-VA database for valence and arousal estimation in-the-wild" paper [7] (we are going to group them as if they belonged to one model, although in reality a distinct model is responsible for one or at most two of the values). We compare with the 5-fold cross validation ResNet152 model and the 5-fold cross validation VGG11 model.

Table 2: Comparison in terms of RMSE, PCC and ICC for arousal and valence estimation in AFEW-VA between the best results from [7], the 5-fold cross validation of the finetuned ResNet152 and the 5-fold cross validation of the finetuned VGG11.

Model	Arousal			Valence		
	RMSE	PCC	ICC	RMSE	PCC	ICC
Best of [7]	2.229	**0.45**	**0.356**	2.639	0.407	0.29
Finetuned ResNet152	**1.513**	0.432	0.303	**1.556**	**0.49**	**0.392**
Finetuned VGG11	2.392	0.377	0.264	2.509	0.434	0.325

We can see in table 2 that the ResNet152 model is better at most measures for both arousal and valence, which was somewhat expected because we just saw our Resnet152 is close to the AffWildNet results and this architecture far exceeded the best results of [7] on PCC. However, our model is lower, although close enough, to the results of [7] in the case of PCC and ICC for arousal. In the case of VGG11, even though its results are not as good as the ResNet152 ones, it is better than the results of [7] for measures on valence and for RMSE on arousal. We can therefore conclude that these CNN architectures, such as ResNet and VGG, can be satisfactorily used for visual emotion recognition using transfer learning.

When we introduced this objective we also mentioned trying to reach the best results on the SEMAINE dataset in order to have an idea of how well these architectures work on laboratory settings datasets when applying transfer learning. Because the researchers of [7] also trained and tested on this dataset, we can compare with them. It is important to note that our SEMAINE results belong to just one model trained on four folds (validated on the remaining). In particular, the one that has the best results from the models that we could train for each architecture. We could not train the five models for each case to be able to apply 5-fold cross validation due to lack of time. So these have to be seen as preliminary results.

Table 3: Comparison in terms of RMSE, PCC and ICC for arousal and valence estimation in SEMAINE between the best results from [7], our best fold model from the finetuned ResNet152 and the best one from the finetuned VGG11.

Model	Arousal			Valence		
	RMSE	PCC	ICC	RMSE	PCC	ICC
Best of [7]	1.608	0.272	0.245	2.087	**0.35**	**0.331**
Finetuned VGG11	**0.511**	**0.336**	**0.318**	**0.586**	0.240	0.208
Finetuned ResNet152	0.519	0.282	0.269	0.608	0.187	0.161

In this case, is we observe table 3 we can see that VGG11 works better than Resnet152 (we are going to comment this fact in section 5.1.3). For the measures on arousal and RMSE for valence, our models are better than the ones from the paper. Nonetheless, our models are worse in terms of PCC and ICC for valence. We can say that our models could be used particularly for arousal estimation on laboratory settings datasets, not for valence estimation. As said before, this was not a crucial objective to accomplish.

5.1.2 Generalization strength

We will verify for this second objective that using in-the-wild datasets to train produces models that generalize better than models trained on laboratory settings datasets. We will use AFEW-VA to represent the former type of dataset and SEMAINE for the latter one.

We evaluate our best performer, the ResNet152 model trained on AFEW-VA, by testing it on SEMAINE. For the other case, models trained on SEMAINE that were tested on AFEW-VA, we evaluate both a ResNet152 and a VGG11. We use RMSE, PCC and ICC for the comparisons.

Table 4: Comparison in terms of RMSE, PCC and ICC for arousal and valence estimation between a ResNet152 model (trained on AFEW-VA) when tested on SEMAINE, a VGG11 model (trained on SEMAINE) when tested on AFEW-VA and a ResNet152 model (trained on SEMAINE) when tested on AFEW-VA.

Model	Arousal			Valence		
	RMSE	PCC	ICC	RMSE	PCC	ICC
AFEW-VA-trained ResNet152 on SEMAINE	**0.657**	**0.3049**	**0.1945**	**0.545**	**0.315**	**0.253**
SEMAINE-trained VGG11 on AFEW-VA	2.111	0.139	0.108	2.016	0.1155	0.0925
SEMAINE-trained ResNet152 on AFEW-VA	2.028	0.121	0.112	1.898	0.030	0.026

The results in table 4 clearly show that training on an in-the-wild dataset, as expected, accomplishes far better generalization capacities than training on laboratory settings datasets. What is more, if we compare these results with the ones from table 3 of the previous objective, we can see that the ResNet152 trained on AFEW-VA also has a lower RMSE for valence than the models trained and tested on SEMAINE. Our model is also close to the best results of those models on RMSE for arousal and PCC for both arousal and valence, which is pretty significant because it means that training on in-the-wild datasets can be almost as good as training on laboratory settings datasets for tackling data from laboratory settings (that would be our ultimate goal, generalizing fairly well to any situation).

5.1.3 ResNet VS VGG

This is another important objective of the project: Comparing ResNet and VGG for visual emotion recognition. We were expecting ResNet to perform better because this architecture was introduced after VGG, improving its results, and have since been one of the preferred options by the research community to use on deep learning for computer vision.

We have to refer to table 2, only looking at the results obtained by the ResNet152 and the VGG11 models. For all the measures, ResNet152 is better. At first glance, it may seem that ResNet152 works better, at least for in-the-wild datasets. In addition to that, the ResNet152 was trained during 100 epochs, while the VGG11 was trained during 190 epochs. We could use a bigger batch size, of 50, for the former; 20 was used for VGG11 because we could not use a bigger one given our GPU capacity.

Surprisingly, in the case of the SEMAINE dataset, if we look at table 3, we can see that VGG11 is better than ResNet152 for all the measures. We were not expecting this, we expected ResNet to also perform better with SEMAINE. A possible initial explanation we can provide to justify this situation is the fact that SEMAINE is much larger than AFEW-VA (four times larger with the SEMAINE subset that we used). Because VGG is much more complex and has many more parameters (more than double the amount of ResNet parameters), it might be better suited for tackling a much bigger dataset. A more thorough investigation (using other datasets to represent in-the-wild and laboratory settings ones) would be needed before drawing any conclusion (stating that ResNet is better for in-the-wild datasets and VGG is better for laboratory settings datasets, for example).

5.2 Qualitative analysis

The resulting system (as we have said in Pipeline for real-time app, section 4.3.3), a real-time app, takes an image of the user, applies facial landmarking to crop the face area, performs the data preprocessing and feeds the preprocessed image to a trained model, which outputs valence and arousal prediction values for that user's emotional state. One of the objectives of this project was to obtain a model that would provide the system with adequate responses to the emotional affect of the user that is using it. We will subjectively evaluate the app, looking for a concordance between the observable emotional state of the user and the valence and arousal system responses.

Our best performer, the best ResNet152 fold model, was used to power the app. We will evaluate it by performing some of the "basic emotions" and a neutral state because we can clearly map them in the Circumplex model plane in terms of valence and arousal. No ambiguous or complex emotional state will be performed because its corresponding response values would be harder to check.

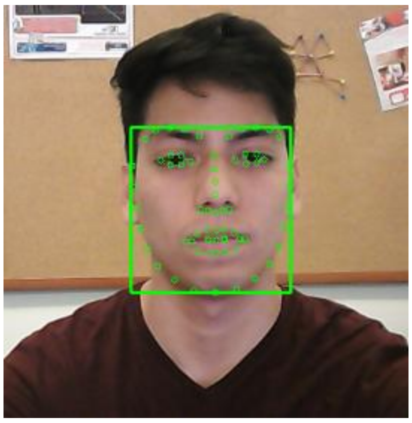
Figure 5.1: User performing a neutral emotional state.

We will begin with the neutral state in figure 5.1. We observe an odd behaviour with the estimations. We expect both valence and arousal to be practically 0, instead, we can see that valence is -1.03 and arousal is 1.82. This is undesirable and seems to indicate that the system does not work properly. However, when inspecting the AFEW-VA data distribution (how often each valence value and each arousal value appears in the dataset) that the researchers provided on their paper [7], we can observe revealing information: The mean of the valence values could be close to -1.5 or -1 (the most occurred values is 0, then -1; the mean is shift to -1.5 or -1 because the positive values are underrepresented in comparison with the rest of the negative values) and the mean of the arousal values clearly seems to be around 2. The data distribution could justify the problem of the system responses for the neutral cases, it would be necessary to address this issue by further analyzing the AFEW-VA dataset.

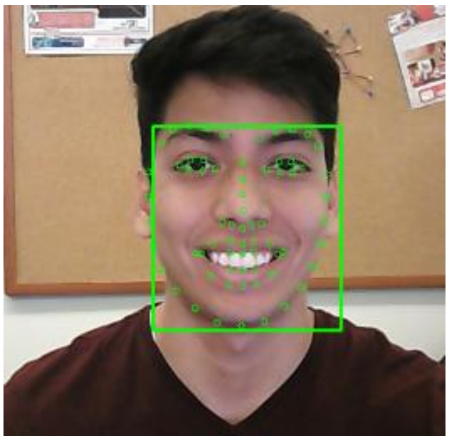
Figure 5.2: User displaying an affect of happiness

For the case in figure 5.2 we displayed a state of happiness. The system responses are a valence of 1.56 and an arousal of 3.12. These values seem coherent with the shown emotional state. 1.56 seems close because happiness has a positive valence value while at the same time not being too extreme, between ranges from 4 to 7. In the case of arousal, 3.12 seems reasonable, this affect has a certain excitement, although, again, not too extreme, from 3 to 5. We could place these values in the plane of the Circumplex model (figure 1.3) near the area of the happiness "basic emotion", although not precisely because of some differences with the happiness ranges (perhaps due to the aforementioned issue with the valence and arousal dataset distribution).

In the test of figure 5.3 we performed a state of anger. The responses are a valence of -2.47 and an arousal of 3.74. Also, these responses seem relatively accurate. As expected, valence is negative and arousal is positive while being more intense than the values for the neutral case. Nonetheless, we expected more extreme values, for valence we can define a range from -5 to -3 (although the system response is close enough), for arousal a range from 6 to 8.

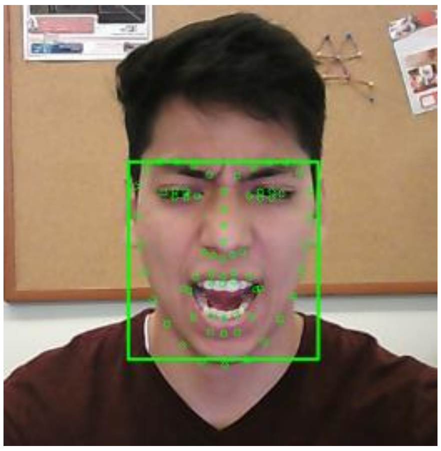
Figure 5.3: User displaying an affect of anger

For the "basic emotions", the system seems to give responses that are somewhat close to the expected ranges. For some cases valence is more accurately responded, for other cases it is arousal. The system responses approach the expected ranges, although not always reaching them. Moreover, we noted, with the neutral case, that the dataset distribution of valence and arousal values has a certain bias towards some values, which could be responsible for issues concerning the system responses not being optimal. The model needs to be far better to power a satisfactory real-time app.

6. CONCLUSIONS

We will recapitulate the findings of the research work, emphasizing the most important objectives accomplished. First we found that we can successfully adapt (employing transfer learning) complex, available CNN architectures (ResNet, VGG) for visual emotion recognition (in particular valence and arousal estimation) using in-the-wild datasets. What is more, our models were fairly close to reaching the state-of-the-art results. Our studies indicate that the ResNet152 is particularly strong for valence estimation, with arousal estimation being more limited when compared with the state-of-the-art.

Secondly, we verified that training on in-the-wild datasets generates models that generalize better than training on laboratory settings datasets. In fact, so much better that the in-the-wild datasets trained ones can be almost as good (in some aspects even better) as the laboratory settings datasets trained ones when testing both on laboratory settings datasets (meanwhile, the ones trained on laboratory settings datasets had really poor results when testing on in-the-wild datasets). We were expecting this to occur, it was the reason why we were investigating the use of in-the-wild datasets, because we suspected that their consideration for the complexities and ambiguities of human affect, in addition to a wide variety of conditions (diverse scenery and non-ideal illumination, head poses and head angles) could be better for generalization capacities.

Another observation we can mention is that while ResNet worked better than VGG for an in-the-wild dataset, the opposite happened for a laboratory settings dataset. The expected outcome was for ResNet to be better at both. Since this did not occur we thought of possible explanations. We could initially say that VGG works better for SEMAINE because this dataset is much larger than AFEW-VA and the VGG architecture is much more complex than the ResNet one, although we consider that a more focused investigation on this topic could be done, using other datasets to represent in-the-wild datasets and laboratory settings datasets.

Lastly, when testing the real-time app powered by the best ResNet152 model, our best performer, we noted that the valence and arousal distribution of AFEW-VA could be responsible for odd responses (observed in the case of a neutral state). “Basic emotions” seem to be estimated relatively close to the expected value ranges, in some cases being accurate to the expected ranges. Ultimately, the system should be much more accurate in its estimations to be considered a successful emotion recognition real-time app.

7. FUTURE WORK

Now that we have showcased the work done and the more significant observations, we can discuss how the emotion recognition systems of the project could be further expanded in the future. In broad terms, these extensions will be oriented in the direction of developing better versions of the systems so that they can be deployed in specific fields.

One of the major motivations for this project was to examine the affective computing field in order to design computing systems that could recognize the users' emotional states as input so that they could more adequately and effectively respond to them. There exists an intention by various researchers and companies to equip devices and softwares with emotion recognition processes in determining areas, such as health, social services, education, entertainment or communications, for addressing the social shifts and necessities and the technological trends of the near future. However, for such implementations to happen, the models should have much better results than the ones we saw earlier.

If the system does not usually respond adequately to the user's affect, then the affective computing objective is missed, because the system answers would not always be coherent with the input. This would result in an unsatisfactory user experience and frustrating interactions. The models previously seen are still not quite apt for a deployment in specific areas, and yet these are the state-of-the-art that the research community has reached for now. The next step, thus, is surpassing the current best models. There could be several approaches for the desired improvement of these systems.

We would like to employ other pretrained systems that we did not use for this project (such as Inception). Apart from that, we would also consider the creation of our own type of architecture, specifically designed for this task (which again would serve to analyze the actual effectiveness of transfer learning), since some research studies on the topic have created their own architectures.

Furthermore, we would look for other in-the-wild datasets. As we have seen, AFEW-VA has certain irregularities in the data distribution of valence and arousal, which could affect the responses of the trained models. This dataset also had the downside of being short. We suspect its dataset size could be the reason why models trained from scratch on AFEW-VA clearly give worse results than when using transfer learning. It would be interesting to train from scratch on a large in-the-wild dataset to see if the results would be better. Additionally, we would like to stick to in-the-wild datasets originated from acted movie instances, since there are many reasons (previously stated) why such datasets should be a better option rather than, say, YouTube reaction videos, or any other supported in-the-wild strategy. Although we would continue working on the acted approach, we acknowledge a formal study comparing both strategies (and any other current strong in-the-wild approach contender) is necessary. We expect new movie originated datasets, larger and more challenging, will be created to train new models so that these could produce better results.

Of course, any new finding occurred in the fields of psychology, neuroscience or affective science would also mean that the research on emotion recognition could be nuanced, boosted or driven in new directions. We still ignore a great deal of the hidden mechanisms behind human emotions, any new discovery could be useful for the improvement of these systems. We would still employ, unless new proposals arise, the valence and arousal continuous model for emotion classification because it describes and captures emotional states in a more meaningful and expressive manner. We have also thought of combining other modalities (such as speech) to the visual processing of the facial features, since for some applications the use of many modalities could be more appropriate.

Finally, two future targets. We would like to study the employment of visual emotion recognition in smartphones, given that they are equipped with cameras and we constantly experience many emotional states through them. For that we consider it would be necessary to build a dataset where its data come from smartphone cameras, given that the images captured by these devices has certain properties of head position, head angle, illumination, image resolution, etc. Apart from that, if we were to apply emotion recognition to a specific field that would be health, working on systems that could help in diagnosis or treatments.

8. BIBLIOGRAPHY

[1] Ahmad, K. (2013). *Affective Computing and Sentiment Analysis*. Springer.
[2] Picard, R. W. (1995). *Affective Computing* (M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 321). Retrieved from MIT Media Lab: Affective Computing Group, https://affect.media.mit.edu/pdfs/95.picard.pdf
[3] Tao, J., Tan T. (2005). *Affective Computing: A Review*. Retrieved from ResearchGate, https://www.researchgate.net/publication/220270285_Affective_Computing_A_Review/download
[4] Cambria, E. (2016). *Affective Computing and Sentiment Analysis*. Retrieved from IEEE Xplore, https://ieeexplore.ieee.org/document/7435182
[5] Ismail, N. (2018). *The success of artificial intelligence depends on data*. Retrieved from https://www.information-age.com/success-artificial-intelligence-data-123471607/
[6] Ekman, P. (1972). *Universals and Cultural Difference in Facial Expressions of Emotion.* Retrieved from PaulEkmanGroup, https://www.paulekman.com/resources/journal-articles/
[7] Kossaifi, J., Tzimiropoulos, G., Todorovicc, S., Pantic, M. (2017). *AFEW-VA database for valence and arousal estimation in-the-wild*. Image and Vision Computing.
[8] *AFEW-VA Database for Valence and Arousal Estimation In-The-Wild*. (n.d.). Retrieved from https://ibug.doc.ic.ac.uk/resources/afew-va-database/
[9] Russell, J. (1980). *A Circumplex Model of Affect*. Retrieved from ResearchGate, https://www.researchgate.net/publication/235361517_A_Circumplex_Model_of_Affect
[10] He, K., Zhang, X., Ren, S., Sun, J. (2015). *Deep Residual Learning for Image Recognition*. Retrieved from arXiv, https://arxiv.org/abs/1512.03385
[11] Simonyan, K., Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Retrieved from arXiv, https://arxiv.org/abs/1409.1556
[12] Guo, L., Wang, L., Dang, J., Zhang, L., Guan, H., Li, X. (2018). *Speech Emotion Recognition by Combining Amplitude and Phase Information Using Convolutional Neural Network.*
[13] Cuervo, M., Matínez L., Alarcón, A. (2017). *Emotion recognition techniques using physiological signals and video games - Systematic review -*
[14] Schuller, B., Valstar, M., Eyben, F., McKeown, G., Cowie, R., Pantic, M. (2011). *The First International Audio/Visual Emotion Challenge.*
[15] McKeown, G., Valstar, M., Cowie, R., Pantic, M., Schroder, M. (2007). *The SEMAINE database: annotated multimodal records of emotionally coloured conversations between a person and a limited agent.*
[16] Krizhevsky, A., Sutskever, I., Hinton, G., (2010). *ImageNet Classification with Deep Convolutional Neural Networks*. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
[17] Kollias, D., Tzirakis, P., Nicolaou, M., Papaioannou, A., Zhao, G., Schuller, B., Kotsia, I., Zafeiriou, S. (2019). *Deep Afect Prediction in-the-Wild: Af-Wild Database and Challenge, Deep Architectures, and Beyond*. Retrieved from ResearchGate, https://www.researchgate.net/publication/331073627_Deep_Affect_Prediction_in-the-Wild_Aff-Wild_Database_and_Challenge_Deep_Architectures_and_Beyond/download
[18] *Ringeval1, F., Sonderegger, A., Sauer, J., Lalanne, D. (2013). Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions*. Retrieved from IEEE Xplore, https://ieeexplore.ieee.org/document/6553805

[19] *20+ Emotion Recognition APIs That Will Leave You Impressed, and Concerned.* (2015). Retrieved 2019 from, https://nordicapis.com/20-emotion-recognition-apis-that-will-leave-you-impressed-and-concerned/
[20] *Disney Is Building Facial Recognition to Figure Out When You'll Laugh During Toy Story 5*. (2017). Retrieved march 2019, https://gizmodo.com/disney-is-building-facial-recognition-to-figure-out-whe-1797267294
[21] *The 7 most likely ways Apple will use Emotient, the artificial intelligence startup it just bought*. (2016). Retrieved march 2019, https://www.businessinsider.com/how-apple-could-use-emotient-2016-1?IR=T
[22] Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer.
[23] Slavio, J. (2017). *Deep learning and Artificial Intelligence.* CreateSpace Independent.
[24] *Machine Learning Will Drive Economic Growth In Every Industry*. Retrieved may 2019, https://www.guidepoint.com/machine-learning-will-drive-economic-growth-in-every-industry/
[25] Aggarwal, C. (2018). *Neural Networks and Deep learning*. Springer.
[26] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
[27] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L. (2019). *Backpropagation Applied to Handwritten Zip Code Recognition*. Retrieved from IEEE Xplore, https://ieeexplore-ieee-org.sare.upf.edu/document/6795724
[28] Major Milestones of Artificial Intelligence from 1949 to 2018. (2018). Retrieved may 2019, https://medium.com/@angelapowell/major-milestones-of-artificial-intelligence-97d42bb5714c
[29] SUPSI. (2013). *2011: First Superhuman Visual Pattern Recognition*. Retrieved march 2019, http://people.idsia.ch/~juergen/superhumanpatternrecognition.html
[30] *Why Deep Learning Has Not Superseded Traditional Computer Vision*. (2018). Retrieved march 2019, https://zbigatron.com/has-deep-learning-superseded-traditional-computer-vision-techniques/
[31] *Why Deep Learning over Traditional Machine Learning?* (2018). Retrieved march 2019, https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063
[32] *HPC High Performance Computing: Home.* (2019). Retrieved november 2018, https://guiesbibtic.upf.edu/recerca/hpc
[33] Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.
[34] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep learning*. MIT Press, http://www.deeplearningbook.org
[35] A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size. (2017). Retrieved january 2019, https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/
[36] Stochastic Gradient Descent with momentum. (2017). Retrieved may 2019, https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d
[37] Convolutional Neural Network. Retrieved may 2019, https://developer.nvidia.com/discover/convolutional-neural-network
[38] A Comprehensive Guide to Convolutional Neural Networks. (2018). Retrieved april 2019, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[39] China's 'Big Brother' surveillance technology isn't nearly as all-seeing as the government wants you to think. (2018). Retrieved may 2019, https://www.businessinsider.es/china-facial-recognition-limitations-2018-7?r=US&IR=T

[40] Preventing Deep Neural Network from Overfitting. (2018). Retrieved may 2019, https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a
[41] Ruiz, A., Martinez, O., Binefa, X. and Sukno, F.M. (2017). *Fusion of Valence and Arousal Annotations through Dynamic Subjective Ordinal Modelling*. 12th IEEE International Conference on Face and Gesture Recognition, Washington, DC, USA, 2017.
[42] Transfer Learning Introduction. Retrieved march 2019, https://www.hackerearth.com/practice/machine-learning/transfer-learning/transfer-learning-intro/tutorial/
[43] ImageNet. (2016). Retrieved march 2019, http://www.image-net.org/
[44] Evolution of CNN Architectures: LeNet, AlexNet, ZFNet, GoogleNet, VGG and ResNet. Retrieved april 2019, https://iq.opengenus.org/evolution-of-cnn-architectures/
[45] The limits and challenges of deep learning. (2018). Retrieved february 2019, https://bdtechtalks.com/2018/02/27/limits-challenges-deep-learning-gary-marcus/
[46] Finetuning Torchvision Models. Retrieved november 2019, https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html
[47] A Gentle Introduction to Transfer Learning for Deep Learning. (2017). Retrieved march 2019, https://machinelearningmastery.com/transfer-learning-for-deep-learning/
[48] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, Li. (2015). *ImageNet Large Scale Visual Recognition Challenge*. Retrieved from arXiv, https://arxiv.org/abs/1409.0575
[49] Ioffe, S., Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Retrieved from arXiv, https://arxiv.org/abs/1502.03167
[50] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich A. (2014). *Going Deeper with Convolutions*. Retrieved from arXiv, https://arxiv.org/abs/1409.4842
[51] Lingenfelser, F., Schiller, D., Martinez, O., Stellingwerff, L. (2017). *Advanced version of the mimics and gesture analysis techniques and multimodal fusion*. Retrieved from Kristina project, http://kristina-project.eu/media/cms_page_media/32/D4.3.pdf
[52] Wu, Y., Ji, Q., (2018). *Facial Landmark Detection: a Literature Survey*. Retrieved from arXiv, https://arxiv.org/abs/1805.05563
[53] PyTorch. Retrieved november 2019, https://github.com/pytorch/pytorch