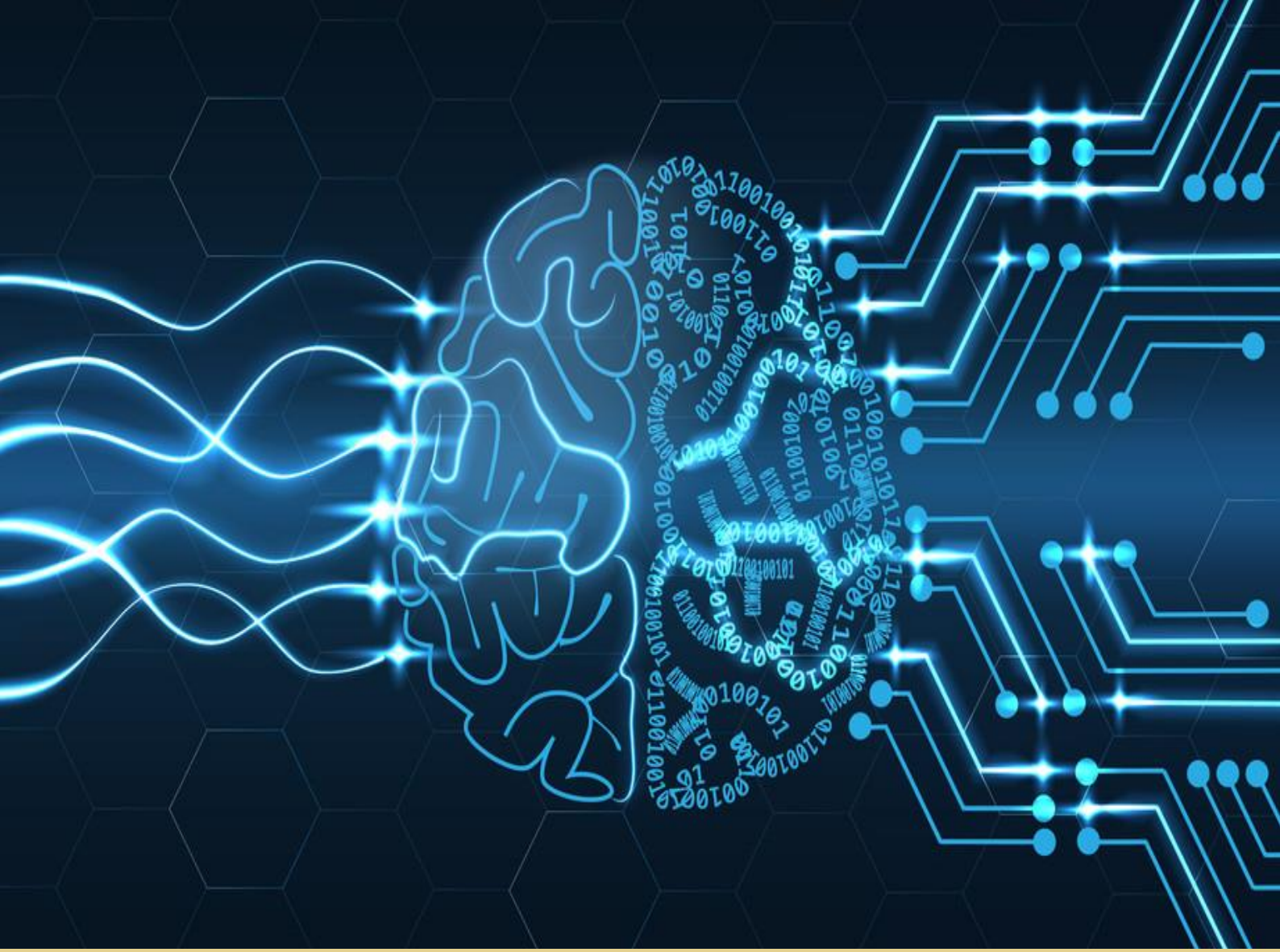


Diogo Manuel de Castro Rodrigues

Integrating Vision and Language for Automatic Face Descriptions

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Electrical and Computer Engineering

September, 2018

· U C ·

UNIVERSIDADE DE COIMBRA

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Integrating Vision and Language for Automatic Face Descriptions

Diogo Rodrigues

Coimbra, September 2018

Integrating Vision and Language for Automatic Face Descriptions

Diogo Rodrigues

Supervisor:

Prof. Doctor Simon Dobrišek

Co-Supervisor:

Prof. Doctor Hélder Araújo

Jury:

Prof. Doctor Jorge Manuel Moreira de Campos Pereira Batista

Prof. Doctor Fernando Manuel dos Santos Perdigão

Prof. Doctor Hélder de Jesus Araújo

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Coimbra.

Coimbra, September 2018

Acknowledgements

First of all, I would like to express my deep gratitude to Ph.D. Simon Dobrišek for his patient guidance, enthusiastic encouragement, and useful critiques of this research work. My grateful thanks are also extended to Ph.D. Vitomir Štruc and Mr. Klemen Grm for sharing knowledge and ideas during the process of implementation. I would also like to thank Ph.D. Hélder Araújo for his advice and assistance during my exchange and for keeping all the details on time.

I would also like to extend my thanks to the Faculty of Electrical Engineering of the University of Ljubljana for making it possible for me to do my master's thesis abroad.

Special thanks should be given to all my colleagues at the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Coimbra for their friendship through this path both in work and at leisure.

Finally, I wish to thank my parents, grandparents, brother, and sister for their endless support and encouragement throughout my years of study.

Abstract

In this dissertation, computer vision and Natural Language Processing (NLP) are integrated to create a unique example of a face-to-text and text-to-face system. Its intention is to provide a solution that can help humans to perform their jobs with better quality and with a quick response. The aim is to create a system that can be used, for example, to describe faces for visually impaired people or to generate faces from descriptions for criminal investigations. However, this is a preliminary version as it is an ambitious goal to be achieved during the time available for its realization.

To accomplish this motivation, a system was created with the capability of describing, textually, facial images, along with the ability to automatically generate face images from text descriptions. The system is divided into two sub-systems. The first part predicts attributes from the face images through a Convolutional Neural Network (CNN) method that are used, further, as a base to the Natural Language Generation (NLG) model. The descriptions are generated on a rule-based methodology. The second part of the system uses a simple keyword extraction technique to analyze the text and identify the attributes on that description. After that, it uses a conditional Generative Adversarial Network (GAN) to generate a facial image with a specific set of desired attributes. The reason why attributes are used as a base on the method is because they are a dominant identifier that can efficiently transmit characteristic about a face.

The results demonstrate, once again, that either CNN and GAN methods are presently the best options for recognition and generation tasks, respectively. This conclusion is due to their convincing results. On the other hand, the NLP methods worked well for their purposes. However, its results are less remarkable, especially the NLG model.

This work proposes a reliable and functional solution for solving this complex system. Nevertheless, this area needs an extensive investigation and development.

Keywords: Artificial Intelligence, Deep Learning, Convolutional Neural Network, Generative Adversarial Network, Natural Language Processing.

Resumo

Nesta dissertação, para criar um exemplo único de um sistema de face para texto e texto para face foi integrado visão por computador e processamento de linguagem natural. O propósito é fornecer uma solução que permita ajudar os seres humanos a realizar funções com maior qualidade e de forma mais rápida. Assim sendo pretende-se criar um sistema que possa ser usado, por exemplo, para descrever rostos para pessoas com deficiência visual ou para gerar rostos a partir de descrições para investigações criminais. No entanto trata-se apenas de uma versão preliminar, na medida em que o curto tempo disponível para a sua realização não permitiu alcançar a ambiciosa proposta.

De forma a atingir este objectivo, foi criado um sistema com a capacidade de descrever textualmente imagens faciais e por outro lado, gerar automaticamente imagens faciais a partir de descrições textuais. O sistema é dividido em duas partes, a primeira tem como função prever atributos das imagens faciais através de uma rede neuronal convolucional. Estes são utilizados como base para o modelo de geração de linguagem natural, gerando descrições textuais numa metodologia baseada em regras. A segunda parte, usa uma técnica simples de extração de palavras-chave para analisar o texto e identificar os atributos nessa descrição. Seguidamente, o sistema usa uma rede generativa adversarial para gerar uma imagem facial com o conjunto das características desejadas. Os atributos são usados como base no nosso método, uma vez que representam um identificador dominante que transmite características sobre um rosto com eficácia .

Os resultados demonstraram, mais uma vez, que os métodos CNN e GAN são atualmente as melhores opções para, tarefas de reconhecimento e geração de imagens, respectivamente. Esta conclusão destá assente nos resultados convincentes. Por outro lado, os métodos de processamento de linguagem natural apesar de terem funcionado bem, de acordo com os

objectivos, os seus resultados são menos notáveis, especialmente o modelo de geração de linguagem natural.

Este trabalho propõe uma solução fiável e funcional para resolver este sistema complexo, no entanto é uma área que merece uma extensa investigação e desenvolvimento.

Palavras-chave: Inteligência Artificial, Aprendizagem Profunda, Rede Neuronal Convolucional, Rede Adversarial Generativa, Processamento de Linguagem Natural

Our intelligence is what makes us human, and AI is an extension of that quality.

— Yann LeCun

Contents

List of Acronyms

AdaGrad	Adaptive Gradient Algorithm
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
D	Discriminator
DL	Deep Learning
G	Generator
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit
HSV	Hue Saturation Value
LSTM	Long Short-Term Memory
ML	Machine Learning
NN	Neural Network
NLG	Natural Language Generating
NLU	Natural Language Understanding
NLP	Natural Language Processing
OpenCV	Open Source Computer Vision
RAM	Random Access Memory
ReLu	Rectified Linear Unit
RGB	Red-Blue-Green

RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
Tanh	Hyperbolic Tangent
TF-IDF	Term Frequency - Inverse Document Frequency

List of Figures

List of Tables

1 Introduction

We are experiencing a significant change in our lifestyle with the fourth industrial revolution or the so-called digital revolution. Society is increasingly dealing with concepts such as advanced robotics, supercomputing, big data, neurotechnology, Internet-of-Things, and last but not least, Artificial Intelligence (AI). As a result, the research conducted in these areas are numerous, and the solutions presented by them are improving every day.

Humans have a recognizable capacity to identify objects or persons by their features that overcome any known machine. Thereby, it is one of the best skills that humans possess. Thus, AI and, consequently, Machine Learning (ML) have developed algorithms to give machines the capacity to at least equalize the human ability and develop applications for the real-world.

On the other side, language is another skill that distinguishes humans from almost all the other animals in the world. However, due to the numerous languages and their complexity, it has been tough to develop algorithms with similar language capacity as humans.

In this thesis, we present a possible solution for a real-world application that connects two big areas, which are computer vision and NLP. The purpose is to provide a solution that can help humans to perform their jobs with better quality and with a quick response. Although our motivation is ambitious, every study and every involvement in any area needs to start somewhere. Therefore, we want to make our investigations publicly available to provide to further researches and researchers some answers to relevant issues such as what can be done, what it performs accordingly to the goal, or what can be enhanced. Our system presents a functional implementation of a possible solution.

Throughout this thesis we develop a unique system. We integrate vision and language in a specific way to deal with the problem of transposing what is seen into text and vice-versa. Our

work applies this concept to facial images. To sum up, our purpose was to create a system able to describe, textually, face images, along with, the capability of generating automatically face images from text.

The main idea was to build an interface software or even a web page in which every person could request a text description from an inputted image of a face or an automatic generated face related with an inputted text description. However, due to lack of time, this step was not accomplished.

Despite similar work to generate text descriptions from images such as [Yao et al., 2010], [Kulkarni et al., 2011], [Mitchell et al., 2012] and to generate images from given text descriptions like [Reed et al., 2016], our database does not have identical data. Theses papers present results on databases with several examples of text descriptions, written by humans, for each image. However, there is no similar dataset for face images. Therefore, instead of developing an end-to-end system as those presented before, our work was divided into two parts to compose the unified system.

The first part of the system is the image-to-text model. Its purpose is to generate text description from given face images. There are similar investigations that also translate images into text such as [Kulkarni et al., 2011], [Yao et al., 2010], [Mitchell et al., 2012]. These three papers work with similar approaches. First, the algorithm detects objects, actions, and relation between them in the image. [Kulkarni et al., 2011] and [Yao et al., 2010] also relate the detected features to the text descriptions for the same image. Second, they use NLP techniques like syntactic and semantic constraints, n-gram, or template-based methods to generate the text descriptions for the images.

Furthermore, in the second part of the system the inverse occurs. At this stage the model generates face images from given text descriptions. [Reed et al., 2016] proposes a similar approach where it uses a conditional GAN. However, instead of using labels as conditional information as our model, it uses descriptions of text written by humans. This method is applied to bird and flower images instead of facial pictures.

In short, our approach follows an attribute-based policy. First, our pattern recognition model predicts the attributes of a specific face image, which are used to generate text descriptions. Second, from our Natural Language Understanding (NLU) model, the text description is decomposed into attributes, which are used, at another stage, as input to the face generator model. In Figure 1.1 we show an illustration of the first stage of the process and in Figure 1.2 an illustration of the second stage of the system.

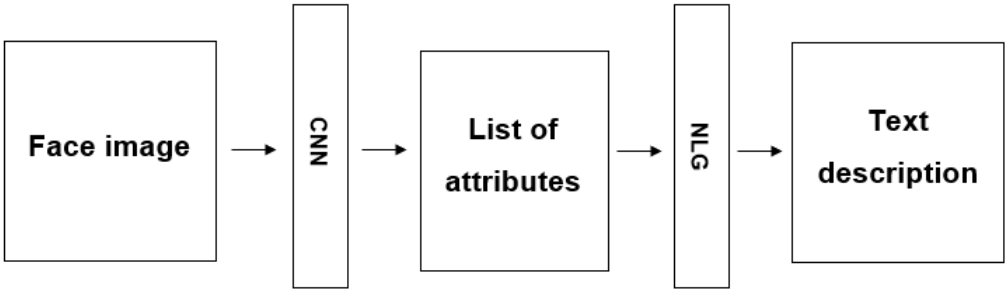


Figure 1.1: Illustration of the Face to Text Part of the System

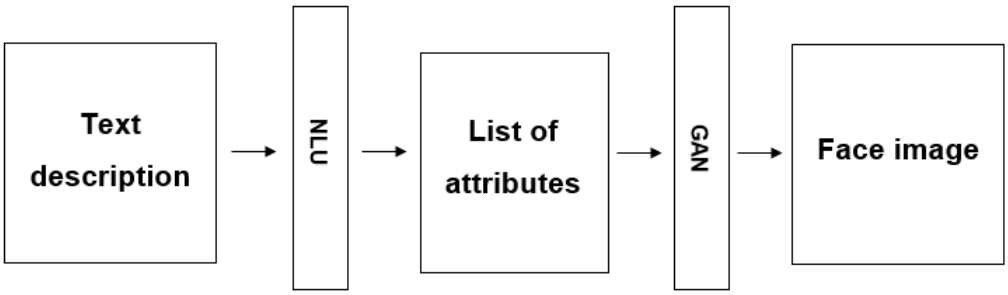


Figure 1.2: Illustration of the Text to Face Part of the System

Moreover, we use attributes to overcome the absence of a dataset with examples of face descriptions. The reason why we did is due to the fact that attributes are a powerful identifier when the goal is recognition [Manyam et al., 2011] or verification [Kumar et al., 2009], [Song et al., 2014], [Berg and Belhumeur, 2013] of a face on an image. Attributes can be beneficial regarding the need to restrict searches or provide information about a face. Depending on the type and amount of attributes that the dataset provides, they can express demographic information, such as gender, age or ethnicity, as well as, physical information about a face like nose size, eyebrow thickness, or hair color, and also certain accessories related to the face, for example, wearing glasses, hats, or earrings.

Initially, the first part of our system is divided into two models. First, the face attribute recognition model is responsible for predicting the attributes that are present in the face image and second the natural language descriptions in which the model uses the predicted attributes as the basis for generating the text descriptions. Like almost all the recognition problems in computer vision, our work takes advantage of a CNN model for the attribute recognition. CNN achieves great success when the task is verification and recognition of objects in images [Huang et al., 2008], [Kemelmacher-Shlizerman et al., 2016]. Both these papers were evaluated in large-scale face images with the purpose of identifying faces, and they present impressive results. On the other hand, the natural language description models have the responsibility of generating text

descriptions according to the number of predicted attributes. The structure of the descriptions generated depends on the number of attributes, since for each number of attributes the model will work differently and will follow a different set of rules to generate the text.

The second stage of our system is also divided into two models. One that applies a basic NLU technique which is keyword extraction. This technique extracts possible attribute candidates from the text description. The other uses those same attributes as input to generate face images. Since our model only works with text that has the purpose of describing face images we do not have to deal with context understanding and, as a consequence, the ambiguity in words. Therefore, our work focuses on extracting keywords rather than understanding the meaning of the descriptions. In this way, we can work with the attributes and pass them into the face generator model. This last part of the system uses a derivation of the GAN [Goodfellow et al., 2014] model. Our approach follows a similar attribute to the face image method as [Yan et al., 2016]. However, both approaches distinguishes themselves on the type of algorithm used. Since our method uses a conditional GAN where the conditional information that will influence the output of the model directly are the attributes and the method presented by [Yan et al., 2016] uses a Conditional Variational Auto-Encoder.

Note that each model was implemented with the intention of being incorporated on the next one, which means that, for instance, the output of the face attribute recognition model is prepared to be the input of the natural language descriptions model.

Each model was evaluated individually since our system is built from four independent models that can work together to form the whole system. Therefore, the models directly related with images, which means the face attribute recognition and the face image generator models were trained and evaluated on the CelebA database [Liu et al., 2015]. Moreover, the NLU model was evaluated with the Face2Text dataset [Gatt et al., 2018].

The unified system that we develop and demonstrate throughout this thesis can have multiple and desirable applications in the real-world. That being said, we will enumerate a few application examples that were in our mind during the procedure of this thesis. However, keep in mind that the research done during this thesis is a preliminary investigation of all the possible applications presented below.

1. The main purpose of this research was to develop a system able to generate face images from queries. The system could receive text descriptions, or even, a combination of selected attributes to generate faces with the desired features.

2. A future outcome from this work could be for interaction purposes. This means that a person could deliver the text description by speaking to a machine and from that the generated faces would appear, for instance, on a screen.

3. Another possible application of the work developed in this thesis would be for crime investigation, in which the graphical interface would sketch a possible example for a generated face image accordingly to the description. This description could be delivered in written or in speech form. The final version would apply this concept to the system, and in the end, the generated face could be improved as the person wanted. This means that the face could be changed according to desire. For instance, specify if a specific attribute is characterized too much or too little.

4. The last possible application would adopt the integrated system to be used for visually impaired people. This way a camera would detect a face and through the features of the face, generate a sentence that could be translated to speech for blind people to hear.

1.1 Main Contributions

The contributions of this thesis are as follows:

- We proposed a solution for a dual problem. Firstly, the generation of text descriptions for face images. Secondly, generating face images from a given text description.
- We tackle the first part of the system by learning CNN and propose a NLG model based on rules to structure the output of the text descriptions.
- Then, we learn an NLU technique by applying a keyword extraction method to the text descriptions. After that, we also study a conditional GAN to generate the face images accordingly to the desired set of attributes.

1.2 Thesis Overview

The thesis is structured as follows. In Chapter 2 we expound the background theory that motif this paper. In Chapter 3 we review the related work. In Chapter 4 we describe the datasets used. In Chapter 5 we clarify the methodology for each of the methods. In Chapter 6 we analyze the performance and present the results for every model that composes the system. We discuss the results and conclude the thesis in Chapter 7.

2 Background Theory

In this section, we discuss some concepts to help the reader to attain the basic knowledge that relates to the topic of the thesis.

2.1 Artificial Intelligence

Regarding AI or machine intelligence, there are entirely different perspectives. On the one hand, there are those who are in favor of its development and of all the changes that it will bring out in our society. On the other, there are those who are against everything that AI can bring to us in the near future. It is difficult to predict which are the benefits or detriments that this technology would bring. We can make a little analogy with a technology that has revolutionized our way of dealing with the problems of our daily life, the Internet. One of the main reasons for this comparison is the capability of the Internet in simplifying our life and making it easier to solve problems. For example, when we want to do a road trip or go to some place that we never been before, we do not rely anymore on paper maps, we just put the desired location into Google Maps and drive through the indications that the application gives us. Perhaps in the future, we will be able to tell our car the desired place, and not even bother with driving ourselves.

The term AI already has existed since 1956, having been described by John McCarthy, and in 1965 Herbert Simon stated that: “Machines will be capable, within twenty years, of doing any work a man can do.” [Hurlbut et al., 2011]. Considering the previous quote, we only have been in contact with the subject matter in the past ten years due to the substantial increase of data and technologies that surround us, which has simplified and significantly improved the performance of AI.

In AI, for most cases, there is a direct association with robots powered with human intelligence. Although, according to [Hurlbut et al., 2011], narrow AI is the most successful demonstration of intelligence. Besides that, narrow AI, which specializes in developing algorithms for specific applications in several areas such as science and industry, has a considerable difference between what is called general AI, which means intelligence compared to the human intellect. As general AI is not goal-oriented research, it makes everything more difficult to accomplish.

In general, AI research aims to give to the machines the same intelligence that is demonstrated by humans and other animals, which is called natural intelligence. In other words, AI will enable a device global decision-making power, learning and acquiring knowledge about several subjects. For the machine to acquire this knowledge, it is necessary to implement algorithms that use appropriately annotated data, and evidently, this data depends on the type of question that is addressed [Deng, 2012] [Liu et al., 2015] [Parkhi et al., 2012].

Current AI capabilities are still not extremely advanced compared to the cognitive abilities of humans. For instance, if we consider image comprehension, which is one of the most well-specified tasks of AI, we realize that there are already learning algorithms capable of identifying several visual and semantic concepts, yet humans are quite adept at recognizing or identifying features through images. Even so, if we consider the performance according to [He et al., 2015] we see that the machine was able to surpass human-level performance on this visual recognition challenge. Note that this does not indicate that machine vision outperforms human vision on object recognition in general, it only states that machine algorithms start to have tremendous potential to match human-level performance on this matter.

Individuals have begun to use existing technologies that proved capable of facilitating some task of our daily lives, such as asking your *smartphone* simple questions and get reasonable answers, or get useful recommendations on Netflix and Spotify, or even to search for keywords on Google Photos and instantly finding pictures wholly related with the subject requested. Although these applications are easy to use and have acceptable results, they are still just simple accessories in our daily lives. However, AI technology is quite far from what it can achieve and to be completely integrated in our way to live. Nevertheless, we already have some applications with much more impact and implication such as music composers, self-driving cars [Bojarski et al., 2016], systems able to compete at the highest level in strategic games like chess [Campbell et al., 2002] or Go [Silver et al., 2016] and even detecting brain tumor [Havaei et al., 2017] or skin cancer, among others.

2.2 Machine Learning

ML is the main branch of AI, and its purpose is to allow machines to have decision-making power. In other words, to enable the machine to go beyond the rules that the programmers have specified them to perform and thus learn by itself how to complete a given task. Succinctly, a ML system is trained rather than explicitly programmed. As such, it is presented with many examples relevant to a task and it finds statistical structure in these examples that eventually allows the system to come up with rules for automating the task.

The enormous growth and exploitation of this area are due to the fact that we now have faster hardware such as Graphical Processing Units (GPUs) and larger datasets. We can simplify the ML operation to three crucial factors, and these are the input data, examples of expected outputs, and the mechanism to measure the accuracy of the algorithm. To verify if the algorithm is indeed learning the method determines the difference between the current output and the expected output values. This distance between the outputs is used as feedback to adjust the algorithm. This procedure is commonly used in several applications, such as image or video analysis, speech recognition, and NLP, among others.

In short, an ML model intends to transform the input data into something that the machine can understand and be aware of when presented with similar data that it has never seen before. Consequently, how the model treats the input data can lead to the final results being closer or further from the target, which means that the data entry procedure is significant. Since there are two possibilities to represent images. Firstly, and probably the most common, in RGB format, and secondly, in HSV format. This is important, because if the purpose is to execute some color classification is more straightforward in RGB format, whereas if the task is to filter the image for a less saturated output, HSV is much more applicable. Therefore, we can conclude that the way data is treated has a direct influence on how accessible it can be for a specific type of task.

ML systems are quite identical with the concept of statistical mathematics. However, these two themes differ on a crucial matter. The ability to deal with large and complex datasets. On the one hand, we have already seen that with ML that can be easily done, on the other hand, with a classical statistical approach such as Bayesian analysis would be impractical. For this reason, we can state that ML and, consequently Deep Learning (DL), are the best tools to use when we want to approach a problem from an engineering view.

When choosing the algorithm and learning style to be applied to the ML problem, it is necessary to consider some important aspects. It depends, among others, on several factors such as the

nature of the task, the quantity and type of data available, the desired output, the time and computational resources available. Despite the differences in the domains of ML, the characteristics of each algorithm will indicate its way of operating. Moreover, some algorithms require memory, whereas others require considerable processing power. Some need a significant amount of annotated data, while others are more efficient with reduced amounts of data.

Therefore, we will give a brief introduction to some concepts to provide the reader the obvious sense of our approach to the problem at hand. To begin with, we have the supervised learning methods that consist of using examples based on input-output pairs of properly labeled data and this method analyses the training data with the aim of mapping new examples. This method is used mainly for pattern recognition task, more specifically for classification.

Aside from that, ML can also use unsupervised learning methods that consist of algorithms that use unlabeled data. So, the outputs cannot have a similar accuracy due to the fact that the task is structured from unannotated data and the purpose is to find groups of similar objects on the dataset. This is the main difference between unsupervised and supervised learning. The methods that usually use unsupervised learning are clustering, anomaly detection and some neural networks, such as, among others, auto-encoders, deep belief nets, GANs.

Third, ML systems can also work with a mixture of these two concepts, a method called by semi-supervised learning. We can describe this technique by literally falling between unsupervised learning and supervised learning. Essentially the algorithm will use a small amount of labeled data as well a significant amount of unlabeled data. This results from a hard time of acquiring properly annotated data from skilled humans due to the significant amount of required time and money to do so.

Lastly, one branch that comes to the surface of the water after getting notice with Google DeepMind is reinforcement learning. This method tries to replicate human behavior in a way that only the good actions will influence the decision-making power. In other words, reinforcement learning uses a software agent to detect information about its environment and choose actions that will maximize some reward. For instance, a human knows when playing football, the main purpose is to score goals, so in reinforcement learning, this action will maximize the purpose of the training. In ML, this method is commonly used for games purposes [Chollet, 2017].

To conclude with a practical solution, one of the most useful applications of ML is the email spam filter. [Tretyakov, 2004] proposed that through the concatenation of the results of several algorithms like the Bayesian classifier, Artificial Neural Networks (ANNs) and support machine vector, can identify which emails are unwanted by the users.

2.2.1 Deep Learning

DL is a specific approach within the large field of ML. This learning method treats data in the same way as ML. Basically, in an approach of a DL model, there is the same input-output mapping which is done by observing several examples of inputs and expected outputs. The only difference between these methods is that the mapping of input to output is done via a deep sequence of simple data transformations, this means, by layers.

Although the number of successive layers describes the depth of the DL model, it does not directly mean that an increase in this depth will positively influence the final results [Sun et al., 2016]. These layers have a parameter called weights, and this property saves in a mathematical way the transformation implemented by the layer to the data. On top of that, model performance will only be at its peak if one set of values for the weights of all layers is found and correctly maps the input data to the desired output. The trickiest part of this process is that by modifying a value of one parameter will automatically influence the behavior of all others. Also, a model can contain hundreds of thousands of parameters and find the balance between all the parameters of a deep neural network could be like trying to find a needle in the haystack.

A DL model is the implementation of any Neural Network (NN). These networks use feedback control to get the knowledge of how the model is behaving before a given example, as well as, to adjust the layer's weight values. Because the initial weight values are randomly assigned, the network will only produce random transformations. However, the training process intends to rearrange these values to get the expected data to be as close as possible to the expected output. At a certain point, and in best-case scenario, the network will achieve its best performance. Additionally, it allowed all layers of the model to learn at the same time. In this way, it automatically adjusts and adapts the internal changes without human intervention. A single feedback signal supervises any change in the model in order to achieve the desired goal. To summarize, regarding the learning process, DL has two crucial factors in its favor. The first one is the way that from layer to layer the method can analyze more detailed features and complex representations. The second is that all layers are connected and learning on the same page. This means that whenever a layer is being updated it will follow the needs of the layer above and the layer below. These two properties made DL a successful technique around the previous ML approaches.

Therefore, the main reason why the DL surpasses the other techniques is that its performance is better and is able to come up with solutions to much more complex problems that would not be obtained with other ML methods. To conclude, it made solving problems easier.

2.3 Neural Networks

In order to implement a solution to an AI problem using DL, we first need to understand the concept of NN. In this section, we will briefly discuss the general concept of an ANN as well as the networks used in this thesis that derive from the main one, which are CNN and GAN networks. These two networks are used in several problems associated with AI. It should be noted that each of these NNs has its specific application, yet it is also possible to implement a solution to the same type of problem using two different NNs. However, each NN typically has an expertise area within AI research and assumes its state-of-the-art status within that same research area. These issues will be addressed individually later.

The great achievement of AI with these NNs was the ability to learn how to solve complex problems in a reasonable amount of time. According to [Yegnanarayana, 2009] some of the desired features that the biological NNs own that sophisticated AI computer system is trying to achieve are the following:

- Robustness and tolerance to failures;
- Flexibility regarding the capacity of the network to automatically adjust to a new environment without using any programmed instructions;
- Ability to deal with a variety of different types of data, such as fuzzy, probabilistic, noisy and inconsistent information;
- Aptness to perform many operations in parallel and also a given task in a distributed manner.

2.3.1 Artificial Neural Network

ANN can be easily explained as being a computing system that tried and accomplished to simulate an actual human brain and consists merely of an interconnected network of processing units, designated by artificial neurons. These artificial neurons are authentic replicas of the neurons that we possess in our brain. As well as our neurons, the artificial neurons are connected via synapses which are represented by the weights. Notice Figure 2.1 for an example of an illustration of an artificial neuron.

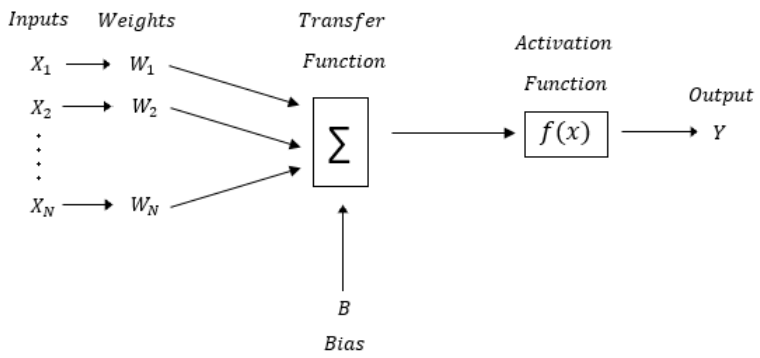


Figure 2.1: Artificial Neuron

The artificial neuron receives a set of inputs (X_1, X_2 ... X_n) and each input is then multiplied with its corresponding weight (W_1, W_2 ... W_n). These weighted inputs are added together along with the bias and passed through an activation layer which gives the output of this particular neuron. That output signal is then used as an input in the next layer in the stack. Relatively to the activation functions, we will address its content further ahead.

Beyond that, an ANN, also known as a multilayer perceptron, is formed by three kinds of fully connected layers.

- Input layer: An input data vector before applying any alteration to the data on it. For instance, if we have the vector X, each value of that vector (X_0, X_1 ... X_{n-1}) will represent an input to a neuron. When the input is an image, like in our case of study, the number of inputs will be the same as the number of pixels in the image.

- Hidden layer: Contains all the hidden neurons as well as all the hidden intermediate operations of the network. Each hidden neuron holds a weight and a bias, which are the elements that will modify the input data. The number and size of the hidden layers may vary.

- Output layer: Here where the classification process takes place, and the output layers have as many neurons as possible classes to predict in the problem. For example, if the task is to predict if an image is of a cat or a dog the number of output classes will be two.

Once the theory behind the structure of an ANN is understood it is important to know how exactly the layers work internally. In the Figure 2.2 we can see a visual example of a ANN with three input layers, four hidden layers and two output layers.

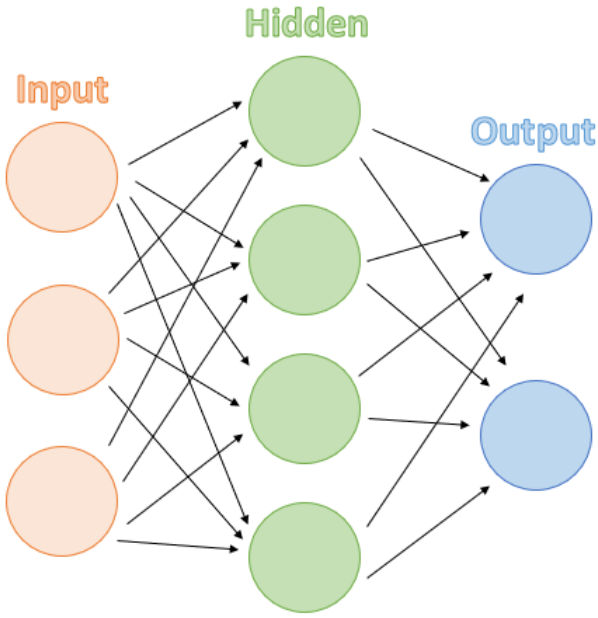


Figure 2.2: Example of an ANN Architecture

The mathematical modifications on the input data follow the following equation:

$$h(x) = s(Wx + b) \tag{2.1}$$

where b is the bias vector, W is the weight matrix and $s(.)$ is the activation function.

Adam Optimizer

Despite classical stochastic gradient descent like SGD, the Adam optimizer does not keep a single learning rate for all weights updates, and, for that reason, escapes from problems due to the fixed learning rate during training. The fact that this optimizer can change their learning rate according to how the parameters are distributed benefits its robustness, and according to [Kingma and Ba, 2014] is a method suitable for use in a wide range of optimization problems in the field of machine learning.

Besides that, this optimizer is characterized by taking the advantages of two other extension of stochastic gradient descent. The methods are Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The advantage acquired from these two methods will provide the Adam optimizer the ability to control the learning rate. For instance, one of the features is the update frequency of the parameter, which means that the lower it is, the bigger the updates will be.

Activation Functions

There are different activation functions, and each of them is used depending on the desired goal. In spite of the variations for each of the activation functions, all of them give non-linearity to the problem and, as a consequence, enable the network to obtain more complex solutions.

The most used activation functions are the Sigmoid or Logistic function, the Tanh (Hyperbolic Tangent) and the ReLu (Rectified Linear Units).

The Sigmoid activation function follows the following equation:

$$f(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{2.2}$$

and its represented by an S shape. In Figure 2.3a we can observe that its range is between 0 and 1. The use of this function brings with it some disadvantages as the output is not zero-centered, has a vanishing gradient problem[1], and has slow convergence. This makes the optimization process harder. However, it is easy to understand and apply.

The Tanh function follows the equation:

$$f(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \tag{2.3}$$

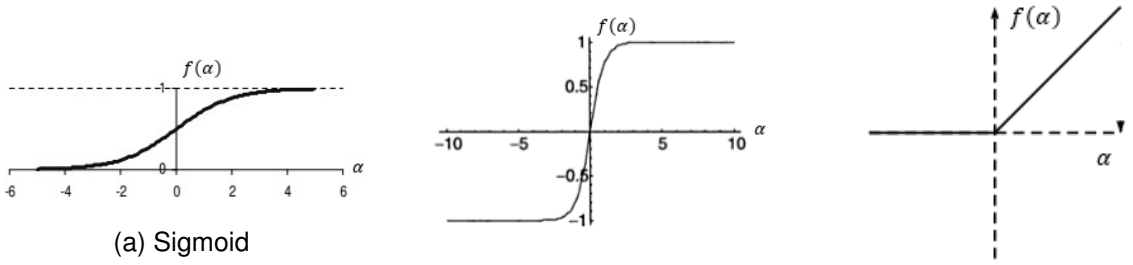
and its graphical aspect is as seen in Figure 2.3b. Its range is between -1 and 1 as observed. The optimization of this method is easier and, for that reason, in practice is always preferable to Sigmoid. However, it still suffers from the vanishing gradient problem.

The ReLu function follows the following mathematical form and an aspect as Figure 2.3c:

$$f(\alpha) = max(0, \alpha) \tag{2.4}$$

This method already proved to be efficient and additionally pays off thanks to its simplicity. One of its benefits is that it avoids and rectifies the vanishing gradient problem. Almost all DL models use the ReLu activation function nowadays.

[1] When the gradient is vanishingly small and the ANN face difficulties to effectively change the weight value. This can lead the NN to stop the training process.



(a) Sigmoid

(b) Tanh

(c) ReLu

Figure 2.3: Behavior Illustration for each of the Activation Functions Discussed. Images adapted from [Karlik and Olgac, 2011] and [Xu et al., 2015].

2.3.2 Convolutional Neural Network

Once we already have the base knowledge about what is a NN and also an ANN, in this section we briefly explain what is a CNN and how it works. CNN is a category of NN that has proven very effective in areas such as image recognition and classification [Huang et al., 2008], [Kemelmacher-Shlizerman et al., 2016]. The fact that these networks can extract high-level features from input images in order to classify them or to extract features vectors has made it possible to successful identify faces, objects, traffic signs, and, consequently, power robots with vision skills. Self-propelled cars are one of the most enthusiastic applications achieved through the use of CNNs [Bojarski et al., 2016] . These networks have been used since the 1990s in works as [Lecun et al., 1998], and since then they have remained mainly similar with no significant variations.

A CNN can easily be interpreted as an ANN, however, there is one factor that sets these two NNs apart. For this type of NN the weights W behave as filters that calculate the convolution of an input image. All layers that define CNN process the input image and, as a result, the output is obtained through the last layer of the network.

Type of Layers

A CNN is formed by the following layers:

- Convolutional Layer: It is the fundamental layer of a CNN. Its main purpose is to extract features from the input image through a set of learnable filters. It preserves the spatial relationship between the pixels, with every filter extending through the full depth of the input volume.

A practical example follows in order to cement the main idea behind convolutional layers. A usual filter size equals to $5 \times 5 \times 3$. The first two values coincide with the width and height pixels and the last value corresponds to the depth, which represents to the color channels[2]. Considering the filter as a window, for each input data the window will slide across the width and height of the image and calculate the dot product between the entries of the filter and the input at every position. As we slide the filter over the image, more precisely, over the matrix, we will produce a two-dimensional activation map or feature map at every spatial position. Each layer will have multiple filters, and each of them will generate an output map like the mentioned above. Intuitively, each filter will be responsible for learning something different and look at different visual features on the image. In the end, all the activation maps are stacked along the depth dimension and produce the output volume.

To be more clear about this matter, see Figure 2.4 for an example of a single convolutional layer. For the input data, the images are considered as a matrix of pixel values, and each pixel will have a value within the range of 0 to 255.

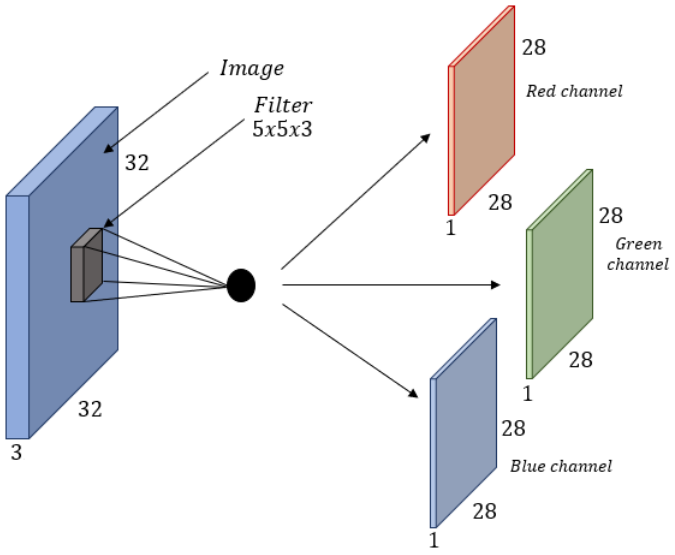


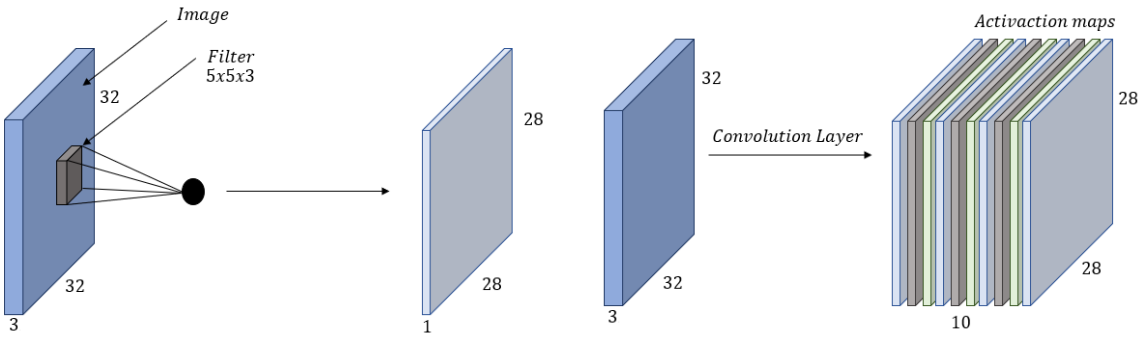
Figure 2.4: Example of a Single Convolutional Layer

The output dimensions will be influenced by the stride[3], the zero-padding[4] and the number of filters for each layer. For example, if we have an image with the dimension $32 \times 32 \times 3$, a stride of one pixel, zero-padding of zero and ten filters, the output value will

[2]Three channels for RGB images and one channel for grayscale images.
[3]The stride is the size of each slide. If it is higher than one it will reduce the output dimensions.
[4]Number of columns and rows with zero value for each position, that will surround the input matrix. Will allow us to control the feature map.

be ten stacked matrix of $28 \times 28 \times 3$. Therefore, connecting the neurons to each other all along the depth of the input will carry an excessively high number of parameters. Nevertheless, this problem has a solution, and that is parameter sharing. This method reduces the number of parameters dramatically by making one reasonable assumption: if the information calculated on a spatial location is useful, then, it should be useful to compute at a different position. Consequently, all the neurons of the same depth will share the same parameters. Figure 2.5 presents two examples of a convolutional layer after applying the parameter sharing technique. In Figure 2.5a the illustration shows a single filter example and in Figure 2.5b, for ten different filters.

Notice that sometimes the method of parameter sharing is not useful for all cases. This method gives the network the ability to learn features everywhere in the image, rather than in just one specific area. This is extremely useful when the object of interest could be anywhere in the image or even have a different object of interest in different parts of the image. For instance, if the image is a face that has been centered in the image, it is common to relax the parameter sharing. Once relaxing the parameter sharing allows the network to look for given features only in a specific area.



(a) One Filter

(b) Ten Filters

Figure 2.5: Example of a Convolutional Layer

- Pooling Layer: This layer has the ability to reduce the dimensionality of each feature map, yet ensures that it retains the most important information. The most used pooling mechanisms are the average and max-pooling.

  The method is based on pure mathematical calculation. For the average pooling, the method will calculate the average from the featured map within a specific window dimension. The same thing will happen for the max pooling, with the difference being that the output value will be the maximum value within that region. The most frequently used dimensions for the window are 2×2, 3×3 and 4×4. Figure 2.6 shows how both pooling

mechanism work with a 2×2 window.

These two methods will guarantee that the network does not over-fit and also will reduce the training parameters. As a result, the model will be lighter, faster, and less computationally expensive.

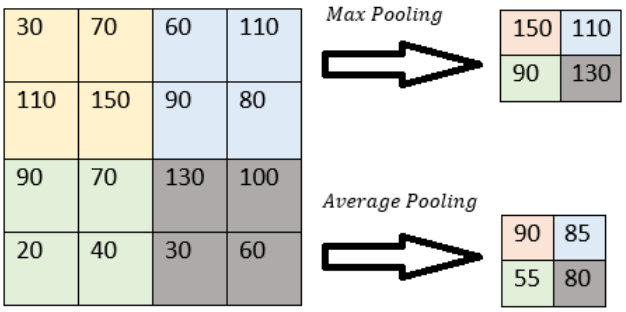


Figure 2.6: Example of the Two Most Used Pooling Mechanisms

- Fully Connected Layer: The purpose of this layer is to use the high-level features obtained from the convolutional and pooling layers to classify the input image into different classes based on the training dataset and the task at hand. Since its purpose is to store the data into categories, this layer is at the end of the CNN.

2.3.3 Generative Adversarial Network

This network was first proposed by [Goodfellow et al., 2014]. The main idea is to have two networks that are trained simultaneously, and each of them tries to fool the other. The method is composed by a generator, G, and a discriminator, D. G is trained to deceive D creating realistic images, and D is trained not to be tricked by G when deciding if either the image is real or not.

At first, G generates images by sampling a noise vector from a simple distribution, usually normal distribution. It is common for the first generated images to be very noisy. Nevertheless, D will identify them as fake images and will force G to improve and generate images more alike to the inserted images. D will be able to decide if an image is real or not because the network will receive as input the generated images by G and also the real images from the dataset. This way D can distinguish real images from fake images. We can simplify the work of D as a classifier that has the option of identifying an image as false or real. Since the generated

images are more similar to real images, D will improve its ability to decide if the samples are as real as the actual images in the database.

The training method is based on backpropagation, explained in the next section, which means that G will receive the feedback from D and if the images are classified as fake, G will improve its network to generate better images. See Figure 2.7 for more information.

The goal of this network is to obtain generated images that are as similar as possible to the actual images in the database.

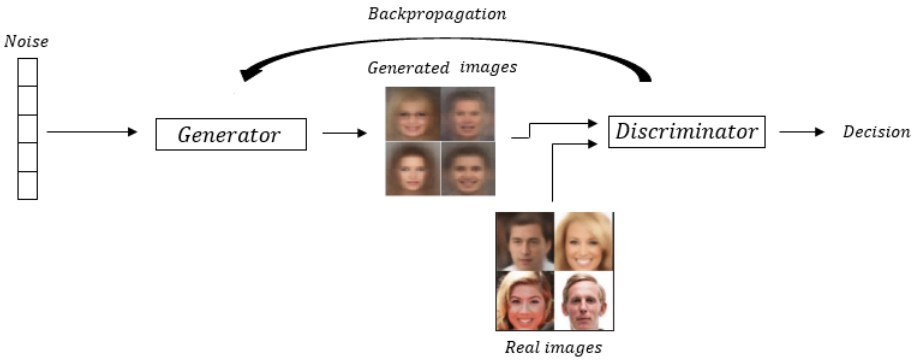


Figure 2.7: Example of a Simple GAN

2.3.4 Training Technicalities

In this section, we will approach, first of all, the most used network training technique and, after that, two methods that are commonly used when building ANNs. All of these methods have the same goal, which is improving the global performance of the networks. All these methods can be applied to any of the several different types of NN.

Backpropagation Algorithm

The backpropagation algorithm, presented by [Rumelhart et al., 1986], was a revolutionary method that come to solve the problems that were unresolvable. Nowadays, this method is one of the pillars of NN.

In general, the network training will only modify the weights W and the bias b, since parameters such as the number and size of filters, architecture of the network have all been fixed before starting the training process. This algorithm will consist of two main concepts. Firstly, the forward propagation step is responsible for predicting the class or group of data that the input belongs. Secondly, the backward propagation step where the network learns by feedback

according to the previous result.

Taking into account that the weights W are randomly initialized the first classifications of the training will also be random. To improve this, the algorithm will modify its parameters. To begin with, the algorithm calculates for each output the total error as follows:

$$TE = \sum \frac{1}{2}(TP - OP)^2 \tag{2.5}$$

Where,

TE - Total Error

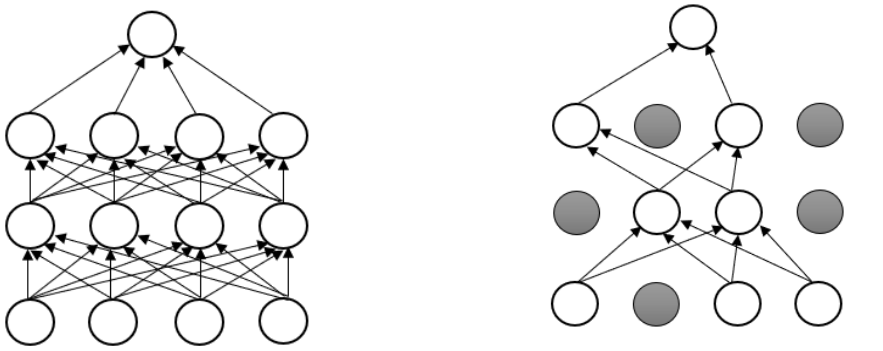
TP - Target Probability

OP - Output Probability

Moreover, using backpropagation, the network can calculate the gradients of the error corresponding to each and all of the weights and use gradient descent to update all parameters values and, as a consequence, minimize the output error. The weights are adjusted in proportion to their contribution to the total error.

Dropout

One of the most contentious problems when developing and training neural networks is overfitting. Therefore, we should use the dropout technique [Srivastava et al., 2014] to try to overcome this issue. Once in training, this method will deactivate part of the neurons, which will improve the capability of learning general information and will force the network to learn the same concept from different neurons. See Figure 2.8 - on the left a fully connected neural network, with two hidden layers and on the right, the same neural network after applying the dropout technique.



(a) Fully Connected Network

(b) Network After Applying Dropout

Figure 2.8: Neural Network

This method is commonly used in DL models, since this models have many layers and, consequentially, many neurons which increases the probability of over-fitting the model. For this reason, the dropout technique is advantageous when the purpose is to learn on a more generalized way.

Batch Normalization

According to [Ioffe and Szegedy, 2015], the use of this method decreases the training time of a network significantly and, as a consequence, it optimizes the NN. The contributions made by this technique are:

- Allows higher learning rates
- Reduces dependency on initialization
- Provides regularization

Owing to the fact that the networks have many parameters that require a careful parameter initialization it forces the use of slower learning rates. As a result, incorporating this method into the NN will help overcome some problems and provide the network with at least the same precision that it would achieve without it.

2.4 Transfer Learning

When trying to solve complex real-life problem in areas like image or speech recognition we need to be smart regarding the possible resources that we need and can use. As seen in

Section 2.2 one of the resources needed to develop such systems are the GPUs and a lot of Random Access Memory (RAM). In the world of today, RAM on a machine is cheap and can be easily available. However, access to GPUs is not that cheap and would involve a significant investment. Besides, to implement a successful DL model, in terms of resources, a significant amount of data is also needed (if dealing with image recognition it is recommended to use several thousand of different images) and consequently a considerable time to train the model. As such, one of the most used solutions is transfer learning. In short, this concept means that it is possible to use pre-trained[5] models by making small changes and adaptations to the problem at hand. In other words, transfer learning can boost the knowledge of our model without the need to implement a NN from scratch.

Another crucial detail when using transfer learning is that if a model is trained on a given type of data, the model that will use its information after training has to be used on similar data, otherwise the results will not be reasonable. For instance, if we are trying to implement a model to recognize faces in images, we should not use a pre-trained model on voice recognition but instead, use a pre-trained model on global image recognition. The modifications that we make on a pre-trained model is called fine-tuning[6] the model.

As referenced in Section 2.2.1, the weights are responsible for saving the acquired knowledge by the layers of the network, and that will influence the performance of the model directly. Therefore, we can directly use the weights and architecture obtained and apply the learning to the desired problem.

For image recognition purposes the most used pre-trained models are trained under the ImageNet dataset [Deng et al., 2009]. This dataset contains 1.2 million images correctly organized into 1000 different categories. These categories represent object classes well known to humans and incorporate many everyday objects such as vehicles types, airplanes, and different species of animals, among other classes.

2.5 Natural Language Processing

Language is the bridge of knowledge. It has the capability of filling the gap between not knowing something and to be experts in it. Furthermore, language, written or spoken, gave to humans the ability to communicate with each other and share thoughts, knowledge, and experiences,

[5] A model created and trained by someone else for identical problems as a starting point.
[6] Process that takes advantage of the front layer features extractor on a pre-trained model network.

among many other things. Language is one of the skills, if not the most important one, that distinct human from other animals.

Accordingly to [Liddy, 2001] NLP is the practical application of theoretical knowledge to analyze and represent texts at various levels of linguistic analysis, in which language processing is intended to be as similar as possible to human perception. Besides, it is intended to apply this concept to several types of task and applications.

Subsequently, NLP is the field that helps the machine to understand, interpret and manipulate human language. Its purpose, as mentioned in the definition above, is to accomplish human-like language processing and, as a result, comprehend the way humans communicate, either by speech or text. We can say that this is not an easy task. One of the reasons is that it takes a few years for humans to begin to develop the ability to speak and begin to have dialogues with other humans. Not to mention all the rules and different languages that comes with it.

Furthermore, the issue of context comprehension goes beyond a machine's ability to understand all the structural, lexical and semantic elements. The same word can have multiple meanings and for humans to understand what is the context in a text or a speech is easy. However, for the machine it is not. Since most NLP methods use probabilistic algorithms and eventually combined with classic DL approaches the results can only be reasonable.

The NLP field has several branches of study such as NLU, speech-to-text and vice-versa, question answering or even sentiment analysis. These are few examples of what it can be achieved inside of this research area. All these specific NLP approaches have multiple ways to process data, and each of them does it with a different technique according to the desired goal.

To be more specific, nowadays, we have natural language interaction applications to perform specific tasks only by voice command. Intelligence assistants such as Siri, powered by Apple, or Alexa, powered by Amazon, can answer a question with considerable speed and correct answers, saving time to do other important things. This means that these assistants do the job of searching and even replies as if it were a real person. The new Android powered by Google is able to go further and make calls in which the call model is always identical regardless of whether the receiver is different. These calls can save you time, and they are a functional example of a natural language spiking device. These are some of the possible applications within a monopoly of options.

3 Related Work

In this section, we discuss the development work already done by researchers in each of the study areas.

3.1 Face Attribute Recognition

Some researches has taken advantage of attributes to develop algorithms that are capable of performing verification or identification tasks. For instance, [Kumar et al., 2011] presented an algorithm that is able to searching through several images to find individuals who look similar to a specific person. This article identifies ways in which attributes can be used, such as in face verification. Face verification detects if two images belong to the same person. This paper significantly contributes to the face recognition research area because it demonstrates that with attributes, it was possible to identify pattern recognition, as well as, semantically search. Both applications had notable successes and thus have led to the enhanced functions which allow the machine to perform closer to human's ability.

Besides that, the work presented by [Vaquero et al., 2009] follows a similar use of attributes as before. The purpose of the application is to detect people with specific characteristics from video surveillance cameras.

Nevertheless, attributes can be presented in two different ways. The traditional way treats attributes with binary values [Tretyakov, 2004], which represents its presence or lack thereof in an image. Besides that, [Parikh and Grauman, 2011] proposed a more natural perspective on visual attributes. According to this perspective, attributes are represented as a continuous value and, for that reason, can be compared as, for example, "more...than" or "less...than". Algorithms that use this last type of attribute representation are commonly used to compare

a pair of images. It has been found that these algorithms identify more similar results when compared to the human point of view.

Furthermore, there are two methods that are used to identify attributes from an image of a face. Firstly, the local methods, like [Luo et al., 2013], [Kumar et al., 2009], [Chung et al., 2012] extract low-level features using, for instance, the help of landmarks[1] to train the classifier to look for a specific number of desirable attributes. Secondly, the global methods such as [Zhang et al., 2014], [Sun et al., 2013], [Liu et al., 2015] focus on extracting high-level features from the images without relying on landmarks[1].

Our model is similar to the global methods, as its focus is on high-level features which recognize attributes on a face image. Additionally, it identifies with [Zhong et al., 2016], since both models take advantage of the VGG-16 pre-trained model to improve the attribute prediction by using its CNN structure for localization. This paper found approximately the same results regarding the attribute prediction values as the state-of-the-art model [Liu et al., 2015] for this task. The state-of-the-art paper presents two combined CNN architectures, one for face localization and the second for attribute prediction, which were both trained differently.

3.2 Natural Language Descriptions

With the intent to create machines that more closely represented humans, there arose a need to create solutions and applications with linguistic abilities, either expressed through speech or text. Previous works focus their efforts on describing textually static images, videos or actions. Several different techniques were studied and implemented to complete such tasks.

Initially, the first NLG methods were based on templates [Mirkovic and Cavedon, 2011]. These methods provide a model with individual words that are used to fill the gaps in the templates. [Krishnamoorthy et al., 2013] generates textual descriptions to identify action in videos with a subject-verb-object template. [Barbu et al., 2012] also generates textual descriptions for videos. However, this approach generates text with the Hidden Markov Model algorithm. Template-based methods are also used for summarization [Zhou and Hovy, 2004], dialogue systems [Channarukul et al., 2003] or describing actions, such as flight affairs [Fedder, 1991]. A similar work, presented by [Yao et al., 2010], shows a rule-based approach to describe images. This approach builds its sentences according to predefined rules, which equate to a combination of grammatical structures. Additionally, this algorithm provides a correspondent lexical tree

[1] Landmarks location that usually represent the coordinates for the eyes, nose and corners of the mouth.

of generated sentences.

Secondly, [Li et al., 2011] proposed a more natural sentence generator. The algorithm uses n-grams with the aim of providing more complex textual descriptions, which generate sentences from scratch. This method is seen as the second option when compared to the methods presented before because it predicts individual words one after another.

This kind of method ensures that the generated sentences are correct in most cases. Despite the reliability of the generated sentences, this method do not provide the most natural language descriptions. The results seem more mechanical and the variety is not consistent with natural speech.

Apart from that, the most recent works have brought advances and improvements in the performance of this kind of task. For instance, the work proposed by [Kiros et al., 2014] presents an encoder-decoder approach, with the Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN) method as the base to generate sentences. In addition, [Zhang et al., 2017b] modified the original GAN [Goodfellow et al., 2014] to generate realistic text through adversarial training. This approach used an LSTM model as a generator and a CNN model as a discriminator. The LSTM model predicts the next word regarding the context-response pair.

To conclude, it can be said that the NN has considerably improved the quality of mechanically generated sentences. They are currently of greater complexity and variety. Although, the degree of complexity in the generated sentences is higher, they still have a considerable percentage of errors, which means that they are still far from equalizing the human capacity in this subject. NN algorithms require a certain amount of similar data to train the model and this requisition is, in our case, a barrier.

Our approach focuses on the textual description of static images and uses a rule-based methodology similar to that of [Yao et al., 2010].

3.3 Natural Language Understanding

NLU is one specific area of the NLP field. This subfield is intended to gather context knowledge about phrases and documents. There are certain applications that are used in hopes to attempt to understand humans. For instance, the work proposed by [Kollar et al., 2010] uses NLU to build a bridge of communication between humans and robots through spatial descriptions. This method captures information about the environment and procedure indications which then

guide himself through the path described by the human.

Previous investigations that focuses on parsing long texts often works with word frequency [Suen, 1979], [Ramos et al., 2003], [Matsuo and Ishizuka, 2004]. Although this technique shows that is fairly accurate, it focuses only on the word itself. When the task is NLU, there are several steps to follow because the words can have ambiguous meanings. Thus, it is necessary to decompose the text into lexical semantics to eliminate ambiguity and obtain a precise understanding of it. The work developed by [Bajwa and Choudhary, 2006] presents a ruled-based approach to understanding the context behind a text.

A simple and more direct NLU approach is keyword extraction. It represents a commonly used method that performs with high accuracy, and its accessibility is a strong point in its favor. Additionally, it works considerably well once it analyzes text in search of particular words that represent global issues. Generally, keyword extraction applications, like ours, do not need context because its text is already limited to one type of context or its purpose can be to retrieve information about a text even without reading it. This NLU technique achieves success by using methods like lexical chains [Ercan and Cicekli, 2007], conditional random fields [Zhang, 2008] or co-occurrence distribution [Wartena et al., 2010]. Note that this last paper has an advantage in comparison to the other since it uses the relation between words to choose the best keywords in order to characterize the text.

A similar NLU approach is key-phrase extraction [Frank et al., 1999]. This paper works under Term Frequency-Inverse Document Frequency (TF-IDF) algorithm. The algorithm's goal is to find which phrases retrieve more information about the document. Therefore, with a set of phrases all text can be, for instance, summed up or subjected to a query search. This technique can be used for keyword extraction as seen in [Ramos et al., 2003]. This technique needs a corpus to relate, in an inverse way, the frequency that a word/phrase appears on a document and in the corpus. Nevertheless, [Matsuo and Ishizuka, 2004] present their approach with comparable results to the method aforementioned with the advantage of not needing a corpus.

Although our model follows a keyword extraction approach, it has no direct relation to the previously mentioned methods because it uses a toolkit to do it.

3.4 Face Images Generator

Although image generation is not a fairly new research area, the breakthrough came with [Goodfellow et al., 2014] when he introduced the GAN method. This method simplified the

resolution of this problem as well as provided state of the art results. Previous methods, such as Restricted Boltzmann Machines [Hinton and Salakhutdinov, 2006], were used for the same purpose. However, the training process is more complex than with the GAN method because it requires pre-training to achieve good performance. Additionally, it only presents reasonable results for data without much variety.

In most cases, algorithms that generate images use unsupervised learning, like [Le, 2013]. Nonetheless, [Odena, 2016], [Springenberg, 2015] already demonstrated that it is possible to have promising results in generating images from semi-supervised data.

Likewise, our purpose is to generate face images that capture a variety of possible features according to the training data, which means that the images are generated with the additional information that is associated with them. GAN and consequently its derivations have been the key to solving problems such as ours. For instance, the papers written by [Mirza and Osindero, 2014], [Odena et al., 2016] and [Gauthier, 2014], use the conditional GAN method to generate images from either associated class labels or labels that somehow describe an image. Interestingly, the derivations of the original method, GAN, achieved astonishing results in generating photo-realistic images with high-quality resolution [Ledig et al., 2017], [Zhang et al., 2017a].

Despite the good results presented by GAN and its derivations, there are other methods that have achieved satisfactory results in generating quality images from desired facial characteristics. For instance, PixelCNN [van den Oord et al., 2016] and CNN [Cate et al., 2017] architectures present alternative approaches to solve our task. Both of these papers present algorithms that are able to generate realistic images from labels or classes.

Our model follows an approach similar to those previously mentioned in that it use the conditional GAN technique. However, the associated information with the images and the input is directly related to the information available in the database used to train the model.

4 Representing Visual and Textual Data

In our work, we employ two distinct datasets that represent, respectively, the images data for the attribute prediction model and the textual descriptions data that will be used for the attribute extraction model. Each of them will be described below.

4.1 Visual Data

We have evaluated our approach using the attribute prediction model on the publicly available dataset CelebA [Liu et al., 2015]. This dataset was composed using 10,177 celebrity faces compiled into a total of 202,599 images. Each image in CelebA is annotated with 40 binary attributes. Besides that, each image has five annotated landmark locations: two for the eyes, one for the nose, and two for the corners of the mouth.

CelebA is usually used to study facial attributes in the wild, which means under unconstrained conditions. Moreover, all the images were collected from the Internet.

4.2 Textual Data

Face2Text is an ongoing project that aims to collect a dataset of the natural language descriptions for human faces [Gatt et al., 2018]. The released version contains 400 randomly selected images from the Face in the Wild dataset [Huang et al., 2007]. Each image has been described, at least three times by each voluntarily participant.

The descriptions will be used as an input for the NLU model. The attributes will be extracted from the sentences.

5 Methodology

In this section, we tackle the methodology process for the whole system that was implemented.

5.1 Face Attribute Recognition

The aim of this model is to predict attributes from a given face in which these attributes represent features on a face image. As previously seen, CNN models represent the state of the art in tasks such as recognition, localization, and classification. To achieve our goal, we use as a base to our recognizer, the VGG-16 pre-trained model [Simonyan and Zisserman, 2014]. This model is well known on the image recognition research area due to its prizes in the sophisticated ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. This publicly available model offers the capacity of classifying objects in images and, for that reason, we choose to adapt it to our purpose of predicting attributes from a face image. The architecture of this model can be seen in Figure 5.1.

The model takes an RGB image with a resolution of 224×224 pixels as input. To respect the

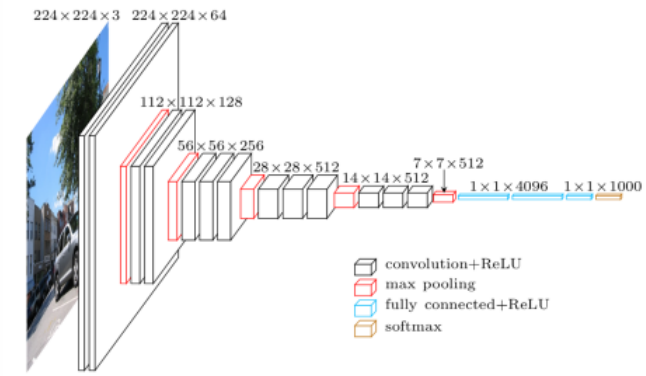


Figure 5.1: VGG-16 Model Architecture. The original image is published in [Blier, 2016]

parameters of the pre-trained model, each image had to be resized because the images in the data set have a different resolution.

Owing to the extent of this thesis, along with the short period of time to conclude it, we chose to predict 3 of the 40 available attributes from the list of annotated features. The chosen attributes to be predicted were eyeglasses, gender, and hat. Note that these attributes, as mentioned in the Section 4.1, are binary, which means that positive values represent the presence of this attribute and negative values the opposite. Except for the gender attribute, that positive values determine that the face belongs to a male and negative values to a female.

With the focus of implementing the feature extractor, we added, to the pre-trained model, three dense layers. Each layer was responsible for predicting one of the three attributes in the face image. Therefore, we froze the training for all the layers corresponding to the VGG-16 model and trained only the last three layers of the model. Note that these three layers were trained simultaneously and not individually.

In order to train each feature that we wanted to recognize equally, the dataset needed to be balanced. This meant that it needed to contain the same amount of images for each combination of the three attributes. The task of balancing the dataset is due to the fact that the model, during training, selects a batch of images at random. As a consequence, this selection is mandatory if we want each combination to have the same probability of being chosen. Otherwise the model trains the most common combinations of attributes or the most common attributes better.

The training and test sets represent respectively 60% and 40% of the total amount of images that fit the selection, which equates to 1000 training images and 672 test images. Each combination of the three attributes appears in the same amount in either the training set or the test set.

As was seen before, the attributes are classified as binary values. Having this, the layers needed to be trained with the Tanh activation function, as it is the only activation function that can predict values between [-1, 1]. Subsequently, since we have results within a restricted set of values, our problem is no longer a classification problem, but rather a regression problem. Therefore, during training, we use a mean absolute error loss metrics, in which it represents the average of the absolute difference between the predicted values and the real values. All the differences have the same weight on the final average. Additionally, the model was trained under the Adam optimizer, and its specifications were already studied previously in this paper.

5.2 Natural Language Descriptions

This model belongs to the first part of our system and proceeds the model presented before. Its purpose is to generate text descriptions that describe the output images of the first model. Once our recognizer only predicts three attributes the text descriptions will be deterministic. Although the sentences for describing at most three attributes are simple, there are a lot of different ways to say the same thing.

In this task, we followed a rule-based method to build a model able to generate different text descriptions. To create diversification in the descriptions, we developed a model with three different structures. Each structure has the goal of generating sentences accordingly to the number of attributes that are predicted on the face image. Basically, if the model receives an attribute, it will generate a text description based on a specific rule-based structure. Additionally, if it receives two attributes, it will generate the text description based on a different structure, and the same will occur if it receives three attributes as input.

Each of the models has one thing in common, which is the beginning of the description. Thus, the model selectively chooses one of the three options available at the beginning of the sentence. Moreover, another common aspect is that, for each description, the gender attribute is always the first characteristic to be described. As a result, every text description starts in the same way, which means with one of the optional beginnings followed by the gender attribute. We decided to restrict textual descriptions to this principle because we have an attribute that describes a person physically and the other two describe attachments that this person may have. Note that, when the model receives only one attribute, this refers to the gender of the person on the image.

The generated text descriptions, as evident, increase the level of complexity proportionally with the increase in the number of attributes. They follow a set of rules and these rules will be explained in detail in the following steps:

- The beginning of each text description follows a set of rules. First, it chooses one of three possible options. After that, and accordingly to the chosen begin for the description, the model automatically adds some words to the sentence. For instance, if the text descriptions start with a pronoun the model will add a determinant to the sentence, on the other hand, if the text descriptions starts with a determinant the model will add either a verb and a determinant or a preposition and a determinant. This process happens for every beginning independently of the number of attributes that the text will describe. As a result,

the model outputs one of the three possible beginnings for the text description.

- As mention before, the first attribute to be presented is always the gender attribute, no matter how many attributes will be described.
- The hat attribute always comes after a determinant.
- With the purpose of giving more variety to the generated descriptions. If the model receives two or three attributes, the second attribute may come after a verb or a preposition.
- For textual descriptions with three attributes, the sentences also vary in the sense of modifying which of the attributes appears first in the description.

5.3 Natural Language Understanding

The intended goal to achieve with this model is to create the first half of the second part of our system. Its target is to interpret and identify which attributes, from our list of attributes, are present in a text description.

Although it seems a simple problem to be solved, it presents two significant difficulties. First, the words that are in our list of attributes can be expressed in many ways, whether by synonyms or expressions or, even identify the lack of their presence through antonyms. Second, certain words can have multiple senses, which means that the same word in a different context can mean a different thing that is understood in the first place by the machine.

Despite the model not having a direct context understanding approach, we develop a model that analyzes the linguistic content and recognizes the semantic similarities of the possible keywords selected by the model. In other words, it identifies the words and set of words that are in the text description and belong to the list of attributes, together with, the words that are similar and have the same meaning as the attributes in our list.

Therefore, the NLTK[1] Python toolkit was used, which was able to provide a practical solution for the task. For that reason, the model follows several steps in order to accomplish the purpose.

- Initially, the text is analyzed with the intention of simplifying the description. This step removes all the stop words and words without meaning to the classification of a face. In this way, all the possible candidate words and set of words to be classified are gleaned as an attribute.

[1] `https://www.nltk.org/book/`

- Secondly, we check all synonyms, hypernyms[2] and hyponyms[3] of the candidate attributes. This way, we can identify the attributes that are present in the text description without being the same exact word.
- In the next stage, we match the list of candidate words, along with, the list of similar words and set of words to our list of attributes.
- Finally, our model outputs a vector or 40 binary values. A positive value indicates that the correspondent attribute was positively identified in the text description by our model. All the other attributes are automatically assigned as missing, which means that they will be attributed negative values.

Note that our model works with binary attributes, and as a result, we are obligated to assume that if the attribute is not identified on the text description, it means that the face does not have that attribute. This is not always true. First, because the person that wrote the description maybe though it would not be important to describe it. Also, our model can wrongly miss an identification and, in the end, assign a negative value to it.

Our model is prepared to identify the antonyms. We executed this step since the text descriptions can mention opposite face attributes and those should be the ones with negative values on the output vector. However, our model is trained to receive binary values and does not have an undetermined value for the attribute.

Bear in mind that our model will be directly influenced by the way people describe the face images. In other words, the model will respond much better if a person describes a face by using attributes rather than describing their emotions, or trying to identify where that person is from, or even characterize their personality by looking to the face image.

5.4 Face Images Generator

Our model follows a similar approach as [Gauthier, 2014] as we also develop a conditional GAN. This model intends to generate face images with a desired combination of attributes automatically. Therefore, the generator receives, in addition to the random noise, additional information regarding the features of the image. The additional information is a vector with binary values that identifies the 40 possible attributes that can be presented in that face image.

[2] A word with broad meaning, which can be characterized as a general term in which specific words belong to that category.

[3] A word with a specific meaning that can be cataloged within a category or general term.

The discriminator provides two results: one related to the decision that represents whether the image is considered real or fake, as well as a vector corresponding to the prediction of the 40 attributes that characterize the face image. In Figure 5.2 the architecture of this model is presented.

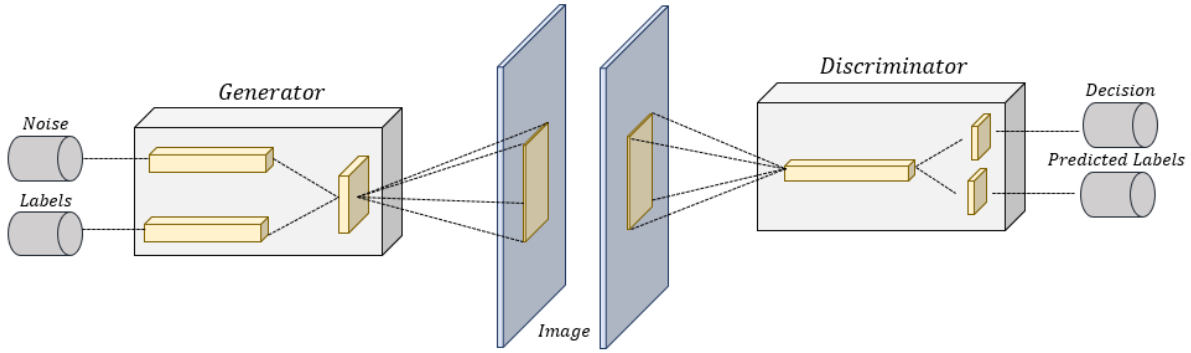


Figure 5.2: Conditional GAN Architecture Implemented

The training was performed with a random selection of 50 000 images from the dataset. Once again, the training received RGB images. However, the resolution of the output images is 56×56 pixels. Before passing the image of the face as input for the model, we proceed to cut the image. In order to do this, we first resize the image to a resolution of 250×250 pixels and then crop the image so that the face is positioned in the center of the image. In this way, we ensure that the attributes are better recognized. Also, we used the Tanh activation function due to the same reasons as explained before in Section 5.1.

In fact, these models have a training performance check. Typically, the model performs a proper training if the loss values for the generator, as well as for the discriminator, converge to a specific value. The fact that they reach stable values means that the model has reached its peak of learning and both the generator and the discriminator no longer have the capacity for improvement. Additionally, it is normal for the loss values to have peaks throughout the training process, as the jumps represent that the model is trying to improve itself. In Figure 5.3 we can observe the values for the losses during the training process of our model.

As we can see the model trained during 400 000 epochs and at each epoch both the generator and the discriminator were trained. Therefore, both parts of our conditional GAN model trained equally. Moreover, when the generator trained, the discriminator froze from training and vice-versa. This way, we ensure that when one part of the model was training, the other part did not suffer any alterations that could influence the performance of the model. Furthermore, having this type of procedure, we ensure that none of the parts that make up the model improves itself

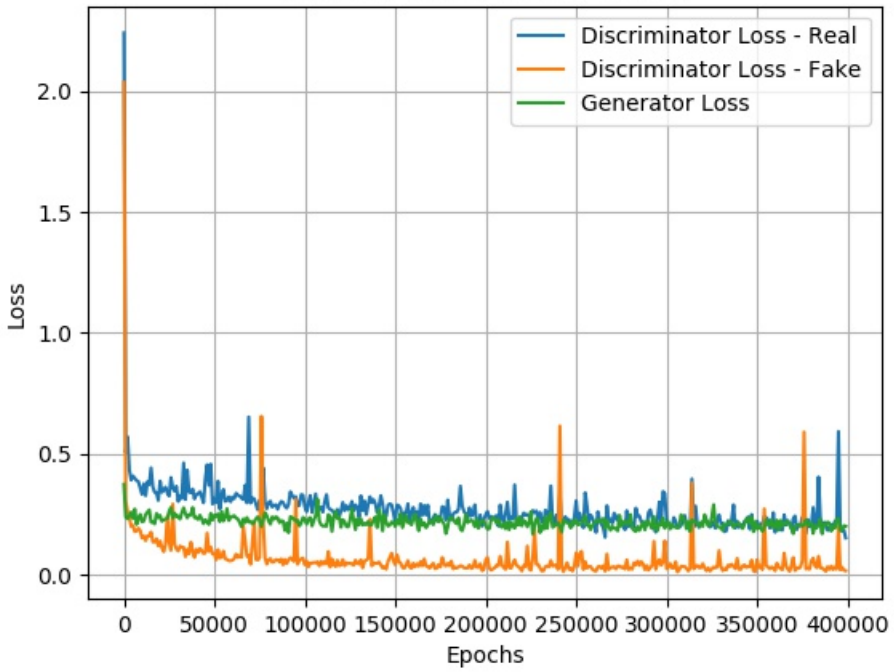


Figure 5.3: Variation of Losses over Epochs

by worsening the other. Regarding training technicalities, the method used the train on the batch, with a batch size of 32 images per epoch.

It needs to be noted that the amount of training data and the batch size used, along with, the resolution of the generated images were due to computational limitations. Otherwise, we would have opted for the use of all images available in the database and for a higher resolution for the generated images.

Besides, the model learning spectrum is limited since we used 50 000 images and 10 000 images for training and testing, respectively. This means that the model will learn to generate the most common combinations of the 40 face attributes. However, even if we used the entire dataset for training, it would not be possible to train all possible combinations equally because even if we use at least one image for each possible combination of the 40 attributes, we will end up with $2^{40} = 1.0995 \times 10^{12}$ images.

In Figure 5.4 we show an example of a set of twelve automatically generated face images.

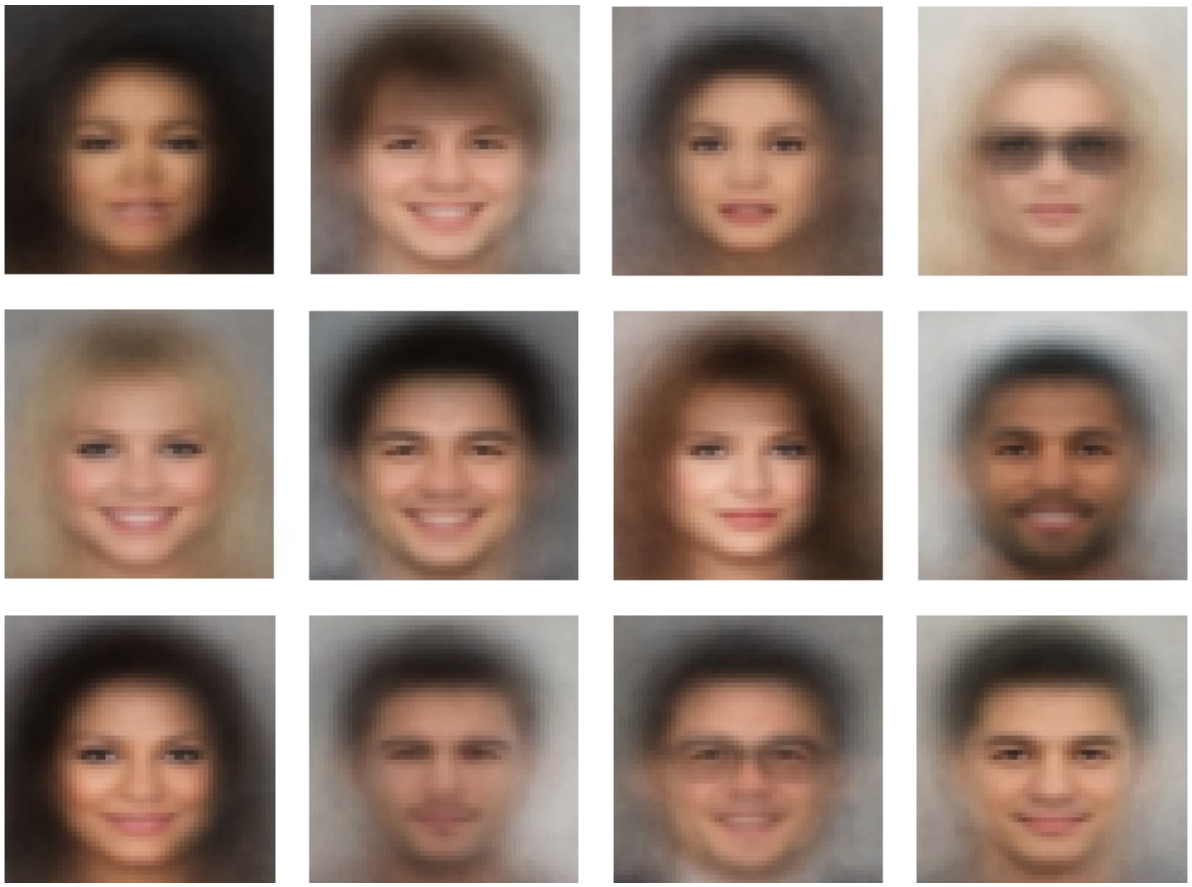


Figure 5.4: Example of a Set of Generated Face Images

6 Evaluation

In this section, we demonstrate the evaluation procedure together with the results achieved for each of the implemented models.

6.1 Face Attribute Recognition

As presented in Section 3.1, our investigations demonstrate that [Liu et al., 2015] demonstrates the state of the art results and they are referenced as the "LNet + ANet" model. Additionally, our model directly relates with [Zhong et al., 2016], and its results are denoted as "Off-the-Shell CNN". Therefore, we compare our results with both previously presented methods on Table 6.1.

The results of our model represent a train and test on a limited amount of images since we needed to balance the dataset, as mentioned in Section 5.1. On the contrary, both models that we use as a comparison had trained and tested their models on all the available images of the CelebA dataset. Therefore, this issue represents the biggest difference between our model and the other two presented as a comparison.

Models	Eyeglasses	Gender	Hat
LNet + ANet	99	98	99
Off-the-Shell CNN	99	99	96
Ours	79	83	94

Table 6.1: Comparing Attribute Prediction Accuracy on CelebA

Although our results are lower compared to the results of the other two models, it needs to be noted that our model was submitted to a significantly lower amount of data. Since we wanted to train each attribute equally, we ended up limiting the learning ability. The amount of images that our model was trained on, represents less than 1% of the available amount of images on

CelebA. As a result, the model failed to evolve enough as it did not have a considerable amount of images to achieve the performance levels as the presented results.

During training, an error is counted if the prediction is incorrect. Thus, we can have false positive predictions and false negative predictions. False positive predictions represent the predictions that were predicted as positive, and their real value is negative. False negative predictions represent the predictions that were identified as negative and their real value is positive. To be more demonstrative, we will present in Figure 6.1 one example per attribute that relates itself with each of the previously mentioned types of errors. In each image, the left part shows an example of a false positive prediction and the right side shows an example of a false negative prediction.

(a) Glasses attribute (b) Gender attribute (c) Hat attribute

Figure 6.1: Example of Prediction Errors

6.2 Natural Language Descriptions

This model was evaluated by 22 individuals with professional working proficiency in the English language. The evaluation demanded that each person analyze 30 sentences per each possible combination of the three attributes, creating a total of 240 automatically generated text descriptions. The evaluation was carried out in order to determine how many descriptions were correctly written according to the English grammatical structure.

In Table 6.2 we show the average accuracy for each group of generated descriptions. The first two groups correspond to the generated sentences of the gender attribute, which can express either if the face is a man or a woman. Secondly, groups three to six represent the combination between the gender attribute with either the eyeglasses attribute or the hat attribute. Finally, the last two groups indicate that both the glasses and the hat attribute are present and one of the attributes of gender.

Group	Accuracy (%)
1	94.4
2	94.8
3	74.5
4	76.2
5	73.9
6	75.6
7	72.9
8	74.8

Table 6.2: Generated Text Description Grammatical Accuracy per each Possible Combination of Attributes

In Table 6.3 we can verify the average performance of our model for each ruled-based text generation since the model have some changes when it receives one, two or three attributes. Therefore, this table shows the results for the correctly generated descriptions according to English grammar per number of attributes.

Number of attributes	Average accuracy (%)
1	94.6
2	75.1
3	73.9

Table 6.3: Generated Text Description Average Grammatical Accuracy per each Possible Combination of Attributes

As explained in section 5.2, the accuracy of generated text descriptions decreases with the increase in sentence complexity. This is because, as complexity increases, we also want to increase the variety in the descriptions generated. Therefore, as complexity and variety increases, accuracy decreases.

Our natural language descriptions model demonstrates an average accuracy of 79.7% for the correct generation of sentences. In the Table 6.4, we show the examples of descriptions that were considered, by the people who evaluated the model, grammatically correct and incorrect.

Number of attributes	Grammatically	Example
1	Correct	"I observe a woman."
	Incorrect	"I see a male."
2	Correct	"I see a lady wearing a hat."
	Incorrect	"An image with a man of contacts."
3	Correct	"I observe a girl wearing sunglasses and a hat."
	Incorrect	"I saw a guy of a sun hat and eyeglasses."

Table 6.4: Examples of Correct and Incorrect Generated Text Descriptions

As observed in Table 6.4, the sentences are simple and descriptive enough to detect the attributes. However, we are aware that it would be impractical to apply this concept to, for instance, ten different attributes.

6.3 Natural Language Understanding

It should be noted that the evaluation procedure had the objective of identifying which attributes were either in the text descriptions as in the list of attributes of the dataset since the goal of the model was to retrieve the attributes on the same page as the evaluation and it would not make sense to make a comparison where the aim was different. Whereas, the text the description could identify other pertinent attributes on the face image that were not present on the list of attributes. Therefore, the evaluation is considered to be of a subjective type once it was executed by reading the text descriptions and identify, by hand, which attributes were either in the description, as well as, on the list of attributes.

The model was evaluated on 100 text descriptions from the Face2Text dataset presented on Section 4.2. In Table 6.5 we show an example of the evaluation that was conducted for two clear text descriptions and, consequently, the results of our model for those sentences.

Text Description	**Retrieved attributes**	
	Human	**Model**
"A young woman with long light brown hair cut in layers and parted in the middle, a small nose and a nice open smile."	Young Gender (female) Brown hair Smiling Mouth slightly open	Young Gender (female) Brown hair Smiling
"A dark-skinned man of possibly American origin with short dark hair which is square-like on his forehead. Sharp eyebrows, long face and large nose and mouth in proportion to his eyes. Ears are quite low on the head and has a small mustache. Vaguely smug expression on his face possibly from a graduation picture or award ceremony."	Gender (male) Black hair Big nose Mustache	Gender (male) Mustache

Table 6.5: Example of the Evaluation Procedure

The model achieved a precision of 74.3% in detecting the keywords in the text descriptions along with 7.7% in identifying mismatched attributes that are not present in the description. We believe that the performance of this model is considered suitable for its purpose. The main reason for its relatively good result is that humans have an extraordinary context perception that

is developing from a young age. Nonetheless, it has been demonstrated that it is complicated to overcome this gap.

6.4 Face Images Generator

The model was tested on 10 000 randomly selected images in the CelebA database. We submitted this model to two distinct evaluations. Although this model has the aim to output a face image, it is directly related to the attributes that the face exposed. Our conditional GAN outputs either the image and the prediction values for the 40 attributes. Therefore, we decided to evaluate the model both in the quality of face images generated and in the attribute prediction accuracy for these same images.

The first evaluation has passed the images through a face detector that outputs the confidence values for the detection of a face. The model used was implemented by [Rosebrock, 2018] and uses OpenCV and DL for detecting only faces on video and static images. Two examples are shown in Figure 6.2. The confidence values for the face detection are quite high. On the left side of the figure is the highest confidence value (99.95%) and on the right side the lowest confidence value (81.34%) when testing our model. With just the lower value, we are verifying high confidence for facial detection. Most probably it is due to the low resolution of the generated images (54×54 pixels) or to the fact that the images only contain faces.

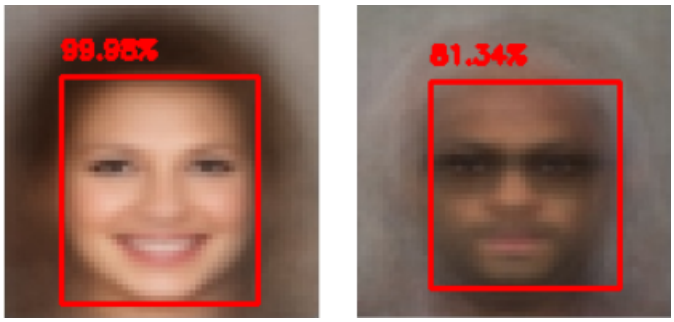


Figure 6.2: Face Detection with Output of Confidence Values using OpenCV and DL

The second evaluation intended to check a comparison between the inputted attributes and the predicted attributes objectively. Since the model is trained under the Tanh activation function, the predictions will be between [-1, 1]. That being the case, the evaluation occurred by thresholding the predictions by the signal of the predicted value and compare directly to the signal of the real value. Our evaluation demonstrates that our model has a general accuracy of 99.5%. This means that our discriminator is working well and it is predicting all the 40 attributes at a

high rate.

We took this evaluation further, and we showed in Figure 6.3 the number of predictions that were between the five different groups of values. The groups were divided according to their prediction values. So the groups were: above or equal to 0.99, under 0.99 and higher or equal to 0.95, under 0.95 and higher or equal to 0.9, under 0.9 and higher or equal to 0.8, under 0.8 and higher or equal to 0.5, and finally, under 0.5. Note that these groups represent the absolute value, which means that all the predicted values in this task were considered positive. Our intention with this part of the evaluation was to show that despite our initial evaluation being by thresholding the predicted values by the signal, they are in 83.5% of the cases close to the real value that was inputted. Additionally, we want to demonstrate that if our model predicts the attribute values with high accuracy, it means that when generating faces with a specific combination, the model will respond accordingly to our needs. However, it is logical to say that it will work better for the most common combinations.

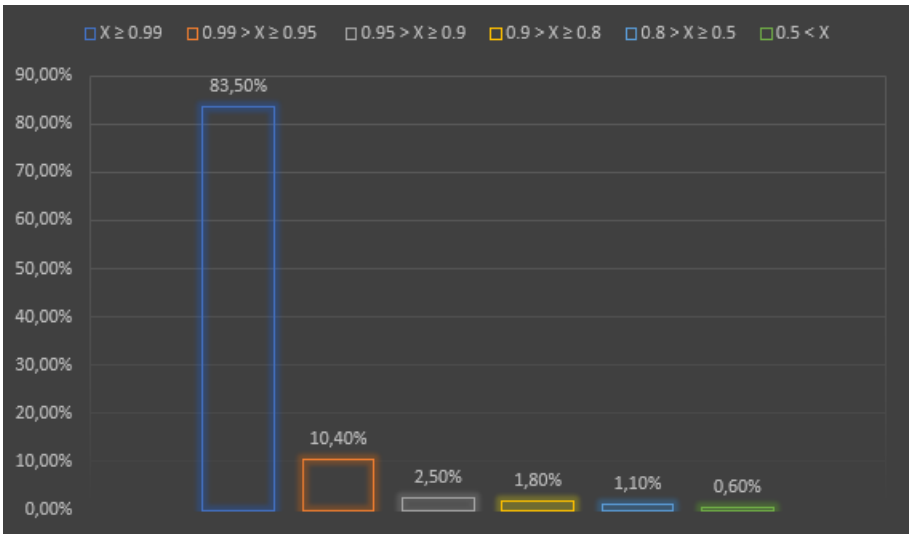


Figure 6.3: Graph Illustrating the Predicted Values by Several Groups of Values

6.5 Unified System

In this section, we will show some examples of how the unified model would work. First, we will present examples of the face-to-text system. And then, we will show examples of the text-to-face system.

In Figure 6.4 we present two different face images. Both face images show the gender attribute and the eyeglasses attribute. Although, the Figure 6.4b shows an additional feature, which is the hat attribute.

Therefore, for the figure 6.4a our face attribute recognizer model successfully predicts the attributes of gender and glasses and the NLG model generates the following textual description: "This person is a gentleman with glasses.".

On the other hand, for Figure 6.4b our face attribute recognizer model predicts the attribute gender, hat and glasses successfully, and at the same time the NLG model generates the following correct textual description: "This individual is a gentleman with a hat and sunglasses.".

(a) Facial Image with Gender and Eyeglasses Attribute

(b) Facial Image with Gender, Eyeglasses, and Hat Attribute

Figure 6.4: Facial Images Used to Evaluate the Unified System

Contrarily, the results for the text-to-face model are as follows:

1. Text description: A dark-haired woman with a surprised open-mouth smile. She is wearing make up and long earrings. She has a sun tanned skin and dark red lipstick.

    - Our NLU model extracted the following attributes from the previous text description: Gender (female), earrings, lipstick, and smiling.

    - Our conditional GAN model generated the face image of Figure 6.5 accordingly to the previous list of attributes:

Figure 6.5: Generated Face Image Accordingly to the First Text Description

2. Text description: This looks like a relaxed young man with a little smile with closed lips but playful eyes. His goatee, branded beanie and hoodie all place him in a young generation though he must be in his thirties. He is in uniform as a young film star.

    - Our NLU model extracted the following attributes from the previous text description: Gender (male), young, goatee, and smiling.

    - Our conditional GAN model generated the face image of Figure 6.6 accordingly to the previous list of attributes:

Figure 6.6: Generated Face Image Accordingly to the Second Text Description

3. Text description: Elderly man in a suit, looks like he is of Mediterranean origin, wearing a mustache and spectacles. He has thinning dark hair.

    - Our NLU model extracted the following attributes from the previous text description: Gender (male), mustache, eyeglasses.

    - Our conditional GAN model generated the face image of Figure 6.7 accordingly to the previous list of attributes:

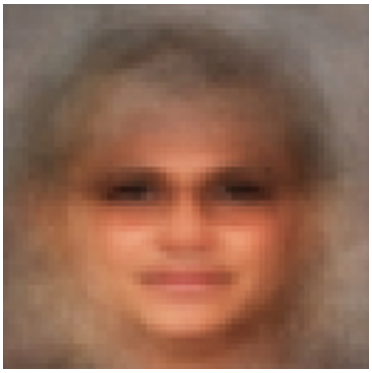


Figure 6.7: Generated Face Image Accordingly to the Third Text Description

Note that the text descriptions previously used to generate the face images have been randomly removed from the Face2Text database presented in Section 4.2.

7 Conclusion

The ambition of this thesis was to find a solution for a face to text and text to face system. Despite similar research for describing global images and vice-versa, our investigations demonstrate that, at this time, there are not similar approaches to describing faces images together with generating face images from a text description. Therefore, we managed to integrate vision and language in a unique way, and the most important achievement was to create a functional system.

Firstly, we demonstrate, once again, the effectiveness of CNN models when applied to recognition problems. The results are not identical to similar approaches, principally, due to the amount of data used. Furthermore, we showed the power of generative networks for purposes of generating images with the implementation of a conditional GAN, and above all the model had remarkable results.

Regarding NLP methods, we used simple and basic techniques to solve either the NLG and NLU problems. We are also aware that the NLG model that is responsible for generating the text descriptions is the weak point of our system. However, we did not have many options to solve this task. Nevertheless, we presented multiple NLG approaches that can easily be implemented for a better solution. On the other hand, the NLU model had significant results in the task of removing keywords from the text. We think that this is the right method to solve our problem since our goal is to detect which attributes are being described in the text. We proved that the technique works and, for that reason, it should continue to be studied for improvement.

Generally, the results on the CelebA dataset demonstrate that our method can generate deterministic and objective face descriptions, as well as generate realistic face images with a specific combination of attributes from a given text description.

In short, we solved a complex and ambitious system. Therefore, it has some details that can, and should be improved. We implemented a robust solution and genuinely believe that the work developed during this thesis should be an ongoing study to enhance the global performance of the method to the problem at hand.

7.1 Future Work

Since this is an area with ongoing research, as mentioned before, there are several hypotheses to improve the performance of the whole system. As such, the following additional research is proposed to enhance the solution discussed throughout the dissertation:

- Evolve the first part of the system so that the model predicts all the 40 binary attributes, as well as develop another natural language descriptions model. During this thesis we showed in Section 3.2 methods can be applied to solve the generation of text description task.
- Use a more extensive dataset annotation with attributes that identify more objective features such as eye color, skin color or age group identification attributes. This way the method can improve the description of a face and generate with more accuracy the face image.
- Manipulate the solution to work with attributes that have an undefined value. Thus, if the attributes have a midterm value, the generation of facial images could be more exact. This alteration would permit a representation with attributes that are present in the face, attributes that are not, and features that are not explicit on the description.
- Enhance the NLU model to have the capability of relating the missing attributes with the given attributes. [Torfason et al., 2016] proposes an approach that could be applied to this problem.
- Work with a dataset that also has annotations for emotions. Thus, the text descriptions and the generation of faces can be more descriptive and accurate.

Besides, there is another specification that can be added to the system. For instance, if a particular description is not detailed enough to extract the more critical aspect from it, such as gender, hair or age. The method could have the intelligence to choose according to the most common results on that same attribute. However, it ought to be kept in mind that this only would be appropriate to be implemented if the system was to be applied in a small environment, such

as big cities or small countries.

8 Bibliography

[Bajwa and Choudhary, 2006] Bajwa, I. S. and Choudhary, M. A. (2006). A rule based system for speech language context understanding. *Journal of Donghua University (English Edition) Vol*, 23(06).

[Barbu et al., 2012] Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., et al. (2012). Video in sentences out. *arXiv preprint arXiv:1204.2742*.

[Berg and Belhumeur, 2013] Berg, T. and Belhumeur, P. (2013). Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–962.

[Blier, 2016] Blier, L. (2016). A brief report of the heuritech deep learning meetup. url = https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/.

[Bojarski et al., 2016] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[Campbell et al., 2002] Campbell, M., Hoane Jr, A. J., and Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, 134(1-2):57–83.

[Cate et al., 2017] Cate, H., Dalvi, F., and Hussain, Z. (2017). Deepface: Face generation using deep learning. *arXiv preprint arXiv:1701.01876*.

[Channarukul et al., 2003] Channarukul, S., McRoy, S. W., and Ali, S. S. (2003). Doghed: a template-based generator for multimodal dialog systems targeting heterogeneous devices. *Companion Volume of the Proceedings of HLT-NAACL 2003-Demonstrations*.

[Chollet, 2017] Chollet, F. (2017). *Deep learning with python*. Manning Publications Co.

[Chung et al., 2012] Chung, J., Lee, D., Seo, Y., and Yoo, C. D. (2012). Deep attribute networks. In *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, volume 3.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Deng, 2012] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142.

[Ercan and Cicekli, 2007] Ercan, G. and Cicekli, I. (2007). Using lexical chains for keyword extraction. *Information Processing & Management*, 43(6):1705–1714.

[Fedder, 1991] Fedder, L. (1991). Generating sentences from different perspectives. In *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*, pages 125–130. Association for Computational Linguistics.

[Frank et al., 1999] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *16th International joint conference on artificial intelligence (IJCAI 99)*, volume 2, pages 668–673. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Gatt et al., 2018] Gatt, A., Tanti, M., Muscat, A., Paggio, P., Farrugia, R. A., Borg, C., Camilleri, K. P., Rosner, M., and van der Plas, L. (2018). Face2text: Collecting an annotated image description corpus for the generation of rich face descriptions. *arXiv preprint arXiv:1803.03827*.

[Gauthier, 2014] Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014(5):2.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Havaei et al., 2017] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., and Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

[Huang et al., 2008] Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

[Huang et al., 2007] Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

[Hurlbut et al., 2011] Hurlbut, P. W., Penrose, R., and Hameroff, S. (2011). How long until human-level ai ? results from an expert assessment.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[Karlik and Olgac, 2011] Karlik, B. and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122.

[Kemelmacher-Shlizerman et al., 2016] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. (2016). The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882.

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

[Kiros et al., 2014] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

[Kollar et al., 2010] Kollar, T., Tellex, S., Roy, D., and Roy, N. (2010). Toward understanding natural language directions. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pages 259–266. IEEE Press.

[Krishnamoorthy et al., 2013] Krishnamoorthy, N., Malkarnenkar, G., Mooney, R. J., Saenko, K., and Guadarrama, S. (2013). Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, volume 1, page 2.

[Kulkarni et al., 2011] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer.

[Kumar et al., 2011] Kumar, N., Berg, A., Belhumeur, P. N., and Nayar, S. (2011). Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977.

[Kumar et al., 2009] Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE.

[Le, 2013] Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Ledig et al., 2017] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*.

[Li et al., 2011] Li, S., Kulkarni, G., Berg, T. L., Berg, A. C., and Choi, Y. (2011). Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228. Association for Computational Linguistics.

[Liddy, 2001] Liddy, E. D. (2001). Natural language processing.

[Liu et al., 2015] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[Luo et al., 2013] Luo, P., Wang, X., and Tang, X. (2013). A deep sum-product architecture for robust facial attributes analysis. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2864–2871. IEEE.

[Manyam et al., 2011] Manyam, O. K., Kumar, N., Belhumeur, P., and Kriegman, D. (2011). Two faces are better than one: Face recognition in group photographs. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–8. IEEE.

[Matsuo and Ishizuka, 2004] Matsuo, Y. and Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169.

[Mirkovic and Cavedon, 2011] Mirkovic, D. and Cavedon, L. (2011). Dialogue management using scripts. US Patent 8,041,570.

[Mirza and Osindero, 2014] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[Mitchell et al., 2012] Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., and Daumé III, H. (2012). Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756. Association for Computational Linguistics.

[Odena, 2016] Odena, A. (2016). Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*.

[Odena et al., 2016] Odena, A., Olah, C., and Shlens, J. (2016). Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*.

[Parikh and Grauman, 2011] Parikh, D. and Grauman, K. (2011). Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE.

[Parkhi et al., 2012] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. (2012). Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505.

[Ramos et al., 2003] Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142.

[Reed et al., 2016] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

[Rosebrock, 2018] Rosebrock, A. (2018). Face detection with opencv and deep learning. url = https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

[Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Song et al., 2014] Song, F., Tan, X., and Chen, S. (2014). Exploiting relationship between attributes for improved face verification. *Computer Vision and Image Understanding*, 122:143–154.

[Springenberg, 2015] Springenberg, J. T. (2015). Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Suen, 1979] Suen, C. Y. (1979). N-gram statistics for natural language understanding and text processing. *IEEE transactions on pattern analysis and machine intelligence*, (2):164–172.

[Sun et al., 2016] Sun, S., Chen, W., Wang, L., Liu, X., and Liu, T.-Y. (2016). On the depth of deep neural networks: A theoretical view. In *AAAI*, pages 2066–2072.

[Sun et al., 2013] Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3476–3483. IEEE.

[Torfason et al., 2016] Torfason, R., Agustsson, E., Rothe, R., and Timofte, R. (2016). From face images and attributes to attributes. In *Asian Conference on Computer Vision*, pages 313–329. Springer.

[Tretyakov, 2004] Tretyakov, K. (2004). Machine learning techniques in spam filtering. In *Data Mining Problem-oriented Seminar, MTAT*, volume 3, pages 60–79.

[van den Oord et al., 2016] van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798.

[Vaquero et al., 2009] Vaquero, D. A., Feris, R. S., Tran, D., Brown, L., Hampapur, A., and Turk, M. (2009). Attribute-based people search in surveillance environments. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–8. IEEE.

[Wartena et al., 2010] Wartena, C., Brussee, R., and Slakhorst, W. (2010). Keyword extraction using word co-occurrence. In *2010 Workshops on Database and Expert Systems Applications*, pages 54–58. IEEE.

[Xu et al., 2015] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

[Yan et al., 2016] Yan, X., Yang, J., Sohn, K., and Lee, H. (2016). Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer.

[Yao et al., 2010] Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508.

[Yegnanarayana, 2009] Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.

[Zhang, 2008] Zhang, C. (2008). Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180.

[Zhang et al., 2017a] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D. (2017a). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915.

[Zhang et al., 2017b] Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., and Carin, L. (2017b). Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*.

[Zhang et al., 2014] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014). Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer.

[Zhong et al., 2016] Zhong, Y., Sullivan, J., and Li, H. (2016). Face attribute prediction using off-the-shelf cnn features. In *Biometrics (ICB), 2016 International Conference on*, pages 1–7. IEEE.

[Zhou and Hovy, 2004] Zhou, L. and Hovy, E. (2004). Template-filtered headline summarization. *Text Summarization Branches Out*.

Appendix A

List of Attributes

5 o clock shadow
Arched eyebrows
Attractive
Bags under eyes
Bald
Bangs
Big lips
Big nose
Black hair
Blond hair
Blurry
Brown hair
Bushy eyebrows
Chubby
Double chin
Eyeglasses
Goatee
Grey hair
Heavy makeup
High cheekbones
Male
Mouth slightly open
Mustache
Narrow eyes
No beard
Oval face
Pale skin
Pointy nose
Receding hairline
Rosy cheeks
Sideburns
Smiling
Straight hair
Wavy hair
Wearing earrings
Wearing hat
Wearing lipstick
Wearing necklace
Wearing necktie
Young