MASTER THESIS

Estimating Image Aesthetic Value using a Content-Based Convolutional Neural Network Architecture

João Marcello Schubnell Abreu de Rezende Lima

Brasília, December 16, 2019

UNIVERSITY OF BRASILIA

FACULTY OF TECHNOLOGY

UNIVERSITY OF BRASILIA
Faculty of Technology
Department of Electrical Engineering

MASTER THESIS

Estimating Image Aesthetic Value using a Content-Based Convolutional Neural Network Architecture

João Marcello Schubnell Abreu de Rezende Lima

Master Thesis

Examination Board

MYLÈNE CHRISTINE QUEIROZ DE FARIAS, Dra., ENE/UNB
Supervisor

ALEXANDRE RICARDO SOARES ROMARIZ, Dr., ENE/UNB
External Examiner

CRISTIANO JACQUES MIOSSO RODRIGUES MENDES, Dr., FGA/UNB
External Examiner

FICHA CATALOGRÁFICA

LIMA, JOÃO SCHUBNELL
Estimating Image Aesthetic Value using a Content-Based Convolutional Neural Network Architecture [Distrito Federal] 2019.
xvi, 59 p., 210 x 297 mm (ENE/FT/UnB, Master, Electrical Engineering, 2019).
Master Thesis - University of Brasília, Faculty of Technology.
Department of Electrical Engineering

1. Aesthetic Quality Assessment
2. Convolutional Neural Networks
3. Computer Vision
4. Machine Learning

I. ENE/FT/UnB
II. Title (series)

BIBLIOGRAPHIC REFERENCE

LIMA, J. S. (2019). *Estimating Image Aesthetic Value using a Content-Based Convolutional Neural Network Architecture* . Master Thesis, PPGEA.DM-738/19, Department of Electrical Engineering, University of Brasília, Brasília, DF, 59 p.

ASSIGNMENT OF RIGHTS

AUTOR: João Marcello Schubnell Abreu de Rezende Lima
TÍTULO: Estimating Image Aesthetic Value using a Content-Based Convolutional Neural Network Architecture .
GRAU: Master ANO: 2019

João Marcello Schubnell Abreu de Rezende Lima
Department of Electrical Engineering - ENE - FT
University of Brasília (UnB)
Campus Darcy Ribeiro
CEP: 70919-970 - Brasília - DF - Brasil

Dedico este trabalho à minha amada Clara.

Agradecimentos

À minha orientadora Doutora Mylène Farias pelos anos de orientação com muita atenção e paciência. A sua confiança no meu potencial e a sua experiência como pesquisadora e professora, além das inúmeras reuniões e conversas, possibilitaram que este trabalho fosse concluído com sucesso.
Aos colegas do GPDS, Vinícius, Dário, Henrique, Rafael, Daniel, Lucas, Sana e Muhammad, que estiveram presentes para ajudar com diversas dúvidas, além de sempre estarem dispostos para uma conversa sobre a vida dentro e fora da Universidade.
Aos colegas do Ipea, em especial, Jader, Leonardo e Daniel, pelas inúmeras conversas sobre computação e estatística que contribuíram para o enriquecimento deste trabalho de forma imensurável. Além dos momentos de descontração que trouxeram tranquilidade quando todos os experimentos pareciam estar dando errado.
Aos amigos de longa data que durante esse período possibilitaram momentos de descontração e muito ajudaram a aliviar angústias e trazer risadas, em especial ao Bernardo por sua amizade e sinceridade insubstituíveis.
À minha família, Paulo, Maria Ignez e Maria Paula, por todos os anos de apoio e amor incondicional, que me trouxeram até aqui, esta conquista é tanto minha quanto sua.
À minha amada, Clara, pelo seu amor e companheirismo. Estes anos não teriam sido possíveis sem você ao meu lado, durante todos os momentos difíceis e todas as conquistas. Você foi e sempre será a luz no meu caminho. Com muito amor, obrigado.

João Marcello Schubnell Abreu de Rezende Lima

ABSTRACT

In the last decade, there has been an increasing interest in the employment of machine learning methods in computer vision applications. The most popular applications range across several different areas from object classification to biomedical imaging. Among all areas, the quality assessment area has attracted some attention due to its applications in digital image editing, search engine, and network optimization. Several works have highlighted the potentials of employing no-reference image quality assessment methods, but most of them only consider the image attributes and often ignore their aesthetic quality. The aesthetic quality of an image is an important factor that drives the observers interests and plays a key role in visual communication. In this work, we analyze previous studies that tried to quantify aesthetic quality by using hand-designed image descriptors and machine-learning methods, more specifically the deep learning method. We also discuss the difficulties and most important factors when developing aesthetic assessment systems. Later on, we propose a method that takes into account the image content to choose the most suited deep learning architecture. We discuss the implications and results of this novel approach and how they can be further improved. Finally, we present some ideas that can help other researchers make sense of the features learned by convolutional networks.

Keywords: Digital Image Processing, Convolutional Neural Network, Aesthetic Quality Assessment, Content-based Methods, Computer Vision, Machine Learning, Deep Learning

RESUMO

Na última década, a propagação de métodos de aprendizado de máquina despertou o interesse de pesquisadoras em investigar suas aplicações no campo de visão computacional. As implementações mais populares abrangem diversas áreas de estudo, de classificação de objetos até imagens biomédicas. Dentro destas áreas o estudo de qualidade de imagens atraiu muita atenção devido as suas aplicações em edição digital de imagens, otimização de motores de busca e otimização de redes de computadores. Diversos trabalhos destacaram o potencial de sistemas classificadores de imagens sem referência, contudo a maioria destes trabalhos se concentraram nos atributos de qualidade da imagem e descosideraram a sua estética. A estética é um atributo importante que provoca o interesse de uma observadora e tem um papel importante na comunicação visual. Neste trabalho, nós analisamos diversos estudos que desenvolveram métodos para quantificar a qualidade estética de uma imagem, tais métodos abrangem desde descritores desenvolvidos à mão quanto descritores desenvolvidos utilizando aprendizado de máquina, mais especificamente aprendizado profundo. Nós discutimos também as dificuldades e os fatores mais importantes que devem ser levados em conta no desenvolvimento de sistemas de classificação estética. Em seguida, nós apresentamos um método que leva em consideração o conteúdo de uma imagem na escolha de uma arquitetura de rede. Nós discutimos as implicações e resultados deste novo método e como é possível aprimorar ainda mais estes resultados. Por fim, nós apresentamos algumas idéias que podem lançar uma luz sobre os atributos que redes convolucionais aprendem durante o processo de treinamento.

Palavras-Chave: Processamento Digital de Imagens, Redes Neurais Convolucionais, Avaliação Estética, Conteúdo de Imagem, Visão computacional, Apredizado de Máquina, Aprendizado Profundo

SUMMARY

List of Figures

LIST OF TABLES

1 INTRODUCTION

1.1 THE STUDY OF BEAUTY

For millennia, philosophers, researchers and artists have debated about the concepts of beauty and where it emerges from. From architecture to painting some concepts have been shared among many distinct areas, but with common goals of provoking reactions on viewers. Common questions about aesthetics are: "Beauty is in the person or in the object?" and "Does the aesthetic response varies over time and across people?"

The study of aesthetic allows researchers to solve problems on visual ugliness and make design choices more appealing. This is important because the aesthetic of places, objects, and images are often correlated with the user's overall evaluation of such places, objects and images. In order to achieve these aesthetic goals, common techniques are employed throughout the different fields, such as rule of thirds, symmetry, and the golden ratio. In architecture, we can see these rules being applied in the design of buildings and places, to create more pleasing spaces like the ones depicted in Figure 1.1.

In image, and more specifically photography, there are some common aesthetic rules that are similar to the rules used in other fields, but in photography some rules are more relevant than others. In photography, concepts like break of symmetry, leading lines, and rule of thirds are well known compositional elements that are known to attract the observer's attention and make the image more pleasing and interesting. Figure 1.2 [1] depicts how the elements in a picture can be positioned in order to highlight elements of interest. These elements are far from being the only important aesthetic rules. It has been observed that, for different types of photographic styles and different image content, some rules are more relevant than others. For example, in portraits it is common to use a low depth-of-field or contrasting colors to highlight natural elements and make the image more appealing.

This difference between the effect of contents in the perception of beauty will be further

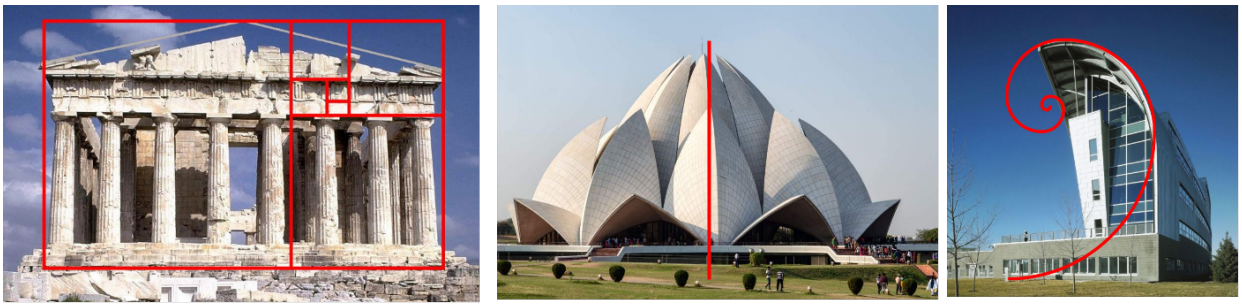


Figure 1.1: From left to right, we can observe classical examples of aesthetic rules in the shapes of buildings. Such choices are common but not necessarily strict.

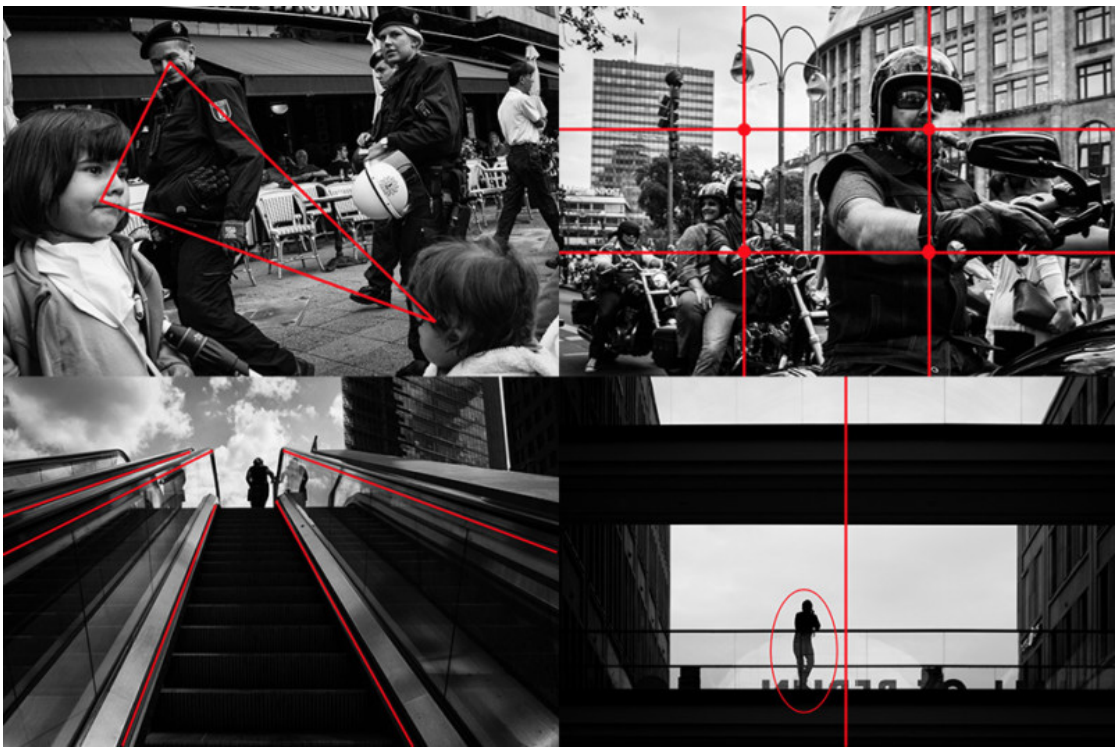


Figure 1.2: Some common techniques among professional and amateur photographers are compositional, such as the use of shapes, rule of thirds, leading lines and break of symmetry. [1]

explored in this work, along with other concepts that might influence how images are evaluated and how we can use computer vision techniques to assess the aesthetic quality of an image.

1.2 IMAGE QUALITY AESTHETIC ASSESSMENT

The evaluation of the aesthetic quality of an image is a problem that puzzles researchers of many areas. Aesthetic can be broadly defined as the branch of philosophy that deals with beauty. It can be analyzed in many different knowledge areas, such as painting, writing, music, and so forth. The present work analyzes the aesthetic in photos, more precisely, it will examine how a computer might be able to perceive the aesthetic value of an image.

The assessment of an image aesthetic quality value is affected by different elements. Certain well established photographic rules are known to influence the perception of an image, like for example the rule of thirds, the golden ration, the image composition, the lighting, the contrast, the color harmony, and the depth of field [3, 4]. Such photographic rules are actively employed by professionals as ground rules that help to improve the appraisal of their work. However, the influence of these rules photographic rules onto the image perception is not entirely comprehended.

Many studies have tried to understand the psychological impact of visual aesthetics on human observers [5–16]. In one hand, the signals produced by the light hitting the cornea are processed in a similar way by the observer's brain, despite their cultural background. On the other hand, a

Figure 1.3: The photo of a beautiful Macaw highlights the use of contrasting colors and low depth of field to drive the eyes of the observer. Photo by Andrew Pons.

person's background, experience, and education level affects the way an image is perceived and how its aesthetic quality is assessed. As a result, both psychological and physiological factors build the perception of the aesthetic quality of images.

Many attempts to develop algorithms to assess the aesthetic quality of an image were made. The main challenges faced by researchers are: the difficulty in modeling both the personal and cultural influence of the person, the lack of images datasets with human-based annotations, and lack of knowledge on how the different genres and styles are perceived by observers and how they might affect the perceived image aesthetic. In the past few years, to improve the design of automatic aesthetic quality assessment new image datasets were made available, such as AVA [17], TID2013 [18], and ILSVRC-2010 [19]. Also, the improvement of computational power and the popularization of deep-learning methods that can be used in image processing application. All these factors have contributed in the design of automatic aesthetic quality methods, with new studies showing the power of machine learning methods.

The improvement of aesthetic assessment systems has impacts in many fields. The use of these kind of systems allow developers to create better tools to assist photographers in all the stages of

a picture, from the moment an image is being captured to its final editing. A photographer might use an assistance system that recommends better angles given the identification of the subject being captured and, during the post-processing stage, a content-based system can recommend crops or color transformations to further highlight the image.

In the Networking field, a content-based aesthetic system can be used to better encode a video signal. This can be done by identifying the most aesthetic pleasing regions in each frame before encoding the signal. Along with signal quality systems, this novel approach has the potential to greatly improve the final user experience.

In the next chapters, we develop a deep-learning based method able to assess the aesthetic quality of an image and we introduce the requirements to use machine learning methods. Our novel method takes advantage of the image's content to better assess its aesthetic quality. Our approach aims to shed light over the influence of the content in the process of learning and further understand which learned features are more relevant for images with distinct contents. We explore how other researches tried to approach the aesthetic assessment problem both with hand-designed and deep-learned feature extractors and the advantages and shortcomings of both of them. Later we describe the methodology and training process employed during our experiments. We present the results that inspired us to take a content-based approach in order to achieve better results. Moreover we study the influence of an image's content during the assessment of its aesthetic value and how some content might be better analyzed by simpler models. Finally, we discuss the results achieved by our proposed method and the difficulties we found through out our study and how we might solve some of the difficulties.

2 MACHINE LEARNING

The learning problem arises from the attempt to understand phenomena that are mathematically and analytically difficult to define, but can be understood when analyzing their data. The data provides an intuitive sense to the observations and this intuitive sense can lead to meaningful conclusions about the phenomena. The assessment of the beauty of an object, scenery, or person is one of such phenomenon. While it is possible, and quite easy, to find a group of people that agrees with the beauty of a portrait, such as the beauty of the portrait of Antonia Zárate by Francisco Goya, depicted in Figure 2.1, or the portrait of Sharbat Gula taken by Steve McCurry, depicted in Figure 2.2, a more challenging task is to define why they feel attracted to such portraits. Such question helps us understand how certain phenomena are easily grasped by their data, but are significantly harder to define. Learning algorithms can provide a way to systematically analyze data from events of interest while also providing tools to understand these events, like the question "Why such image is so beautiful?"

In this chapter, the main ideas that provide the theoretical foundation to machine learning are presented, like for example the necessary elements to design such algorithms and the mathematical background that makes the learning problem feasible. In addition, a baseline framework that

Figure 2.1: Francisco José de Goya y Lucientes (1746 - 1828) was one of the greatest portraitist of all times. In his work reproduced here is shown the portrait of actress Antonia Zárate a successful actress and singer from 19th century Spain. [20]

Figure 2.2: Steve McCury was commissioned by National Geographic in 1984 to portrait the refugee camps along Afghan-Pakistan border. There he was able to capture one of the most recognizable portraits of the 20th century, the picture of Sharbat Gula or as she came to known *The Afghan Girl*

support a more pragmatical analysis of the different learning algorithms employed by Yaser S. Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin in their book Learning From Data [2] is presented. Furthermore, important features that compose the state-of-the-art algorithms and the systems employed in our experiments are explained, like for example convolutional blocks.

2.1 THE LEARNING PROBLEM

The main challenge, while addressing the task of modeling the perception of beauty by an individual or a group, is the broadness and complexity of the criteria used by each one to define what is beautiful. The effort to come up with an analytic solution that takes into account all this criteria might render the task impossible. Nonetheless, the plethora of data about images nowadays in the form of datasets, such as AVA [17] and websites like photo.net [21] and 500px.com [22], might give a glance on a way this challenge can be dealt. These datasets and websites gather information, like the number of likes received by an image or the scores given to the images by users, which can be used as a proxy to the true beauty value of an image. Moreover, descriptors can be used to turn images into vectors, such as the ones proposed by Datta *et al.* [23], in order to create an easier representation of the image. Finally, the relationship between the information taken from the dataset images and the descriptor vector can be devised to serve as an approximation to the underlying aesthetic evaluation. This relationship can be translated by how well different scores are predicted and, based on its performance, how it can be adjusted and further improved. The potential of this approach comes from the fact that all this process can be automated and no additional analysis from the different variables is needed. To achieve this potential, the learning algorithm analyzes classified samples and identifies common features among them that affects the final result and, based on that, it changes the model to improve the results.

2.2 TYPES OF LEARNING

There are three major approaches to the learning problem that function as frameworks to different applications. The approaches are named supervised learning, unsupervised learning, and reinforcement learning. Each of these approaches emerged to deal with different situations and the most important differences between them have to do with how the data regarding an underlying process is presented. In this work, the main focus is placed on the supervised learning approach. In this type of learning, the training data contains the target output for a given input. In the case of beauty evaluation, the data is given in the format (image, score). Later in this chapter, the choice for this kind of approach will become clear.

In a supervised learning approach, the necessary components can be summarized by the diagram in Figure 2.3, developed by Yaser S. Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin [2]. In this diagram, the input $\mathbf{x}$ is the image that is being evaluated represented by a descriptor

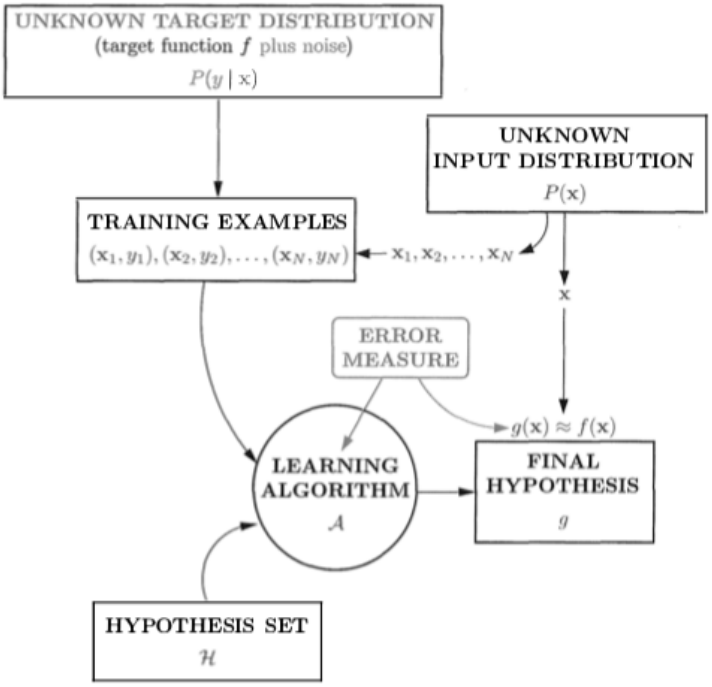


Figure 2.3: The diagram shows the components and steps required in an Supervised Learning problem as presented by Abu-Mostafa *et al.* [2].

that can be a vector, a matrix, or a tensor. At the same time, the target distribution $P(y \mid \mathbf{x})$ is the distribution of scores we are trying to learn given an input $\mathbf{x}$. This distribution is comprised by a target function f plus a noise. The noise is meant to address the problem where two images with the same descriptor might have two different scores, therefore it is added to the target function. The training examples are derived from an input distribution $P(\mathbf{x})$ with their respective score (y) values. The learning algorithm A and the hypothesis set $\mathcal{H}$ together are called Learning Model. Finally, after the learning algorithm explores the hypothesis set it outputs the best approximation to the target distribution it could find, the final hypothesis represented by g.

The hypothesis set is comprised of the many different models, and their respective parameters, employed to represent the target distribution, such as Support Vector Machines (SVM), Neural Networks (NN), and Decision Tree, to name a few. The learning algorithm represents the way in which the Final Hypothesis g will be chosen from the hypothesis set. In the case where we have a single model, such as a Neural Network, a Backpropagation Algorithm may be employed as learning algorithm to choose the best weights of a NN that fits the model to the data available. At the end, a final hypothesis is chosen and becomes the model that better represents the target distribution $P(y \mid \mathbf{x})$. It is important to highlight that this whole process is performed on the training examples available and a few extra steps are required to validate the statistics presented by the final hypothesis and draw conclusions from it. Later in this chapter, these steps will be presented.

Before discussing the feasibility of learning, three requirements should be met in order to employ a machine learning method to model the target distribution. This requirements are: first,

the target function must have a pattern, e.g: "Are there patterns of composition, color, content or any other element in an image that indicate that an image is beautiful?". Second, the pattern being studied should be hard to pin down mathematically, otherwise an analytical solution might be a more efficient approach. Third, there must be enough data available regarding the problem being analyzed.

2.3 THE REQUIREMENTS OF LEARNING

Of all three requirements of learning introduced in the previous paragraph, only the availability of data is indispensable to start a learning process. The other two requirements, although not indispensable, are still important because they help us make sense of the problem being analyzed.

For example, a random process by definition does not have a pattern, therefore rendering the task to find one impossible. The perception of beauty on the other hand is not a random process and in this work we believe that this perception follows a pattern, even if said pattern is restricted to a small group of people. As introduced earlier in Chapter 2, even with a small group, it is a demanding challenge to pin down mathematically or analytically a function that is able to describe beauty in an image. With all that, two of the three requirements of learning how beauty is assessed are met.

The last and most important requirement, data availability, is addressed by the different images datasets available with rich annotations about their images. An important element of this work must be highlighted, this work starts from the premise that said annotations reflect the true beauty of the images and this is, in fact, one limitation as we would benefit greatly from a more diverse dataset created by a diverse range of people from different backgrounds. This work does not try to state what is considered beautiful but rather how an application might learn how to perceive it in a curated set of images. Further works are encouraged in order to validate, both technically and ethically, the employment of machine learning algorithms in vision applications regarding beauty of images.

At the end, with all three requirements met, it is still crucial to address how it is possible to infer statistics generated by a model on datapoints it has not seen and, moreover, why these predictions are trustworthy.

2.4 THE FEASIBILITY OF LEARNING

To address the problem of the capability of a machine learning method to analyze a certain dataset and draw reliable conclusions on datapoints not seen before, effectively the feasibility of learning, a probabilistic tool is employed. By choosing a probabilistic standpoint we can apply well known tools, such as the Hoeffding Inequality[24].

Before diving into the benefits of employing the Hoeffding Inequality, it is important to highlight two important quantities. The in-sample error, E_{in}, and the out-of-sample error, E_{out}. While searching for a hypothesis to represent the *unknown* target function that rules an observation such as a function that takes an image and determine if it is aesthetic pleasing or not, it is necessary to evaluate how well the hypothesis performs. From this point on the term target function will be used interchangeably with target distribution. This change does not affect the generality of the following conclusions. In the case of a supervised learning environment, the samples have known outcomes, i.e. either they are labeled aesthetic pleasing or not. Based on how well the hypothesis predicts the aesthetic class it is possible to define the E_{in} as the following equation:

$$E_{in} = \frac{\text{Number of images correctly classified}}{\text{Total number of images}}. \tag{2.1}$$

The out-of-sample error ensures how well the chosen hypothesis generalizes, but due to the unknown nature of the target function that models the aesthetic assessment it is not possible to directly determine it. For these reasons, the Hoeffding Inequality plays a key role in the feasibility of learning.

The Hoeffding Inequality[24] quantifies the relationship between the in-sample error in an experiment and the out-of-sample error. In other words, the error in the whole population. In some places the out-of-sample error is also called the generalization error. In the particular case of an image aesthetic assessment system, the Hoeffding's Inequality translates how well the assessment system will perform when presented with unseen data. The following equation introduces the Hoefding's Inequality:

$$\mathbb{P}[|E_{in} - E_{out}| > \epsilon] \leq 2 \cdot e^{-2\epsilon^2 N}, \tag{2.2}$$

where the $\mathbb{P}[\cdot]$ stands for the probability of an event. In this case, the probability of a bad event, that is: "the in-sample error is further from the out-of-sample error by a positive value ϵ, a "tolerance", chosen at will". On the RHS of the inequality stands the upper-bound of the probability. The upper-bound is comprised only by the positive interval ϵ and the sample size N that generated the in-sample error measurement.

The upper-bound ensures that the chance of a bad event will decrease with a large sample size, this means that the in-sample error value will be closer to the out-of-sample error value. Moreover, it is possible to define how close the aesthetic assessment is by adjusting the tolerance ϵ value. A lower tolerance yields a greater chance that the bad event will occur and it has to be counter weighted by a greater sample size to oppose this effect.

Before moving on, it is important to note that the Hoeffding's Inequality is only able to address how close the values from E_{in} and E_{out} are. It does not tell how one might be able to lower the value of E_{in}. To achieve a lower in-sample error two important techniques, Regularization and Validation will be introduced later in this chapter.

Although the Hoeffding Inequality allow us to estimate the value of E_{out} based on the value of E_{in} and, consequently, how the model will perform on new data, it has major drawbacks. The *Hoeffding Inequality* only works for a sample randomly chosen a single time, that means that every time a new hypothesis is being analyzed and a new prediction is being generated the *Hoeffding Inequality* results are tampered and becomes biased towards the dataset. To account for multiples tries a small but significant modifications is in order. The following equation will introduce this modification

$$\mathbb{P}[|E_{in} - E_{out}| > \epsilon] \leq 2 \cdot \mathbf{M} \cdot e^{-2\epsilon^2 N}, \tag{2.3}$$

where the value M accounts for the number of hypothesis tested. It is easy to see that, if a high number of hypothesis are tested, the RHS of equation 2.3 grows and the probability of a bad event grows along with it. In fact in a machine learning experiment the hypothesis set grows to **infinite** and the probability in equation 2.3 becomes meaningless.

To prevent the M value from growing while the model is being trained the concept of a growth function is introduced. This growth function measures how different the hypothesis in the hypothesis set $\mathcal{H}$ truly are and sets a boundary to the growth of M.

2.5 THE GROWTH FUNCTION AND THE VC DIMENSION

The *growth function* is a combinatorial quantity that measures the effective number of hypothesis in a hypothesis set $\mathcal{H}$. The effective number of hypothesis represents how many distinct hypothesis generate the same results during learning.

For example, consider an image being analyzed by a model, like the perceptron depicted in Figure 2.4. If two Perceptrons, H_1 and H_2, with distinct weights, predict the same outcome, like both predict that the image has a high aesthetic value, it means that both Perceptrons have produced effectively the same hypothesis. Therefore, the two hypothesis in this case will be counted as one and M will not grow.

From the previous example, it is easy to see that, in the case of a binary classification problem, the maximum number of distinct hypothesis must be at most 2^N, where N is the total number of samples in the training set. This quantity is much lower than all possible M values in the Hoeffding's Inequality and it helps to set a bound on the growth of the RHS on equation 2.3. The following definition defines the growth function,

Definition 2.1 ***Growth Function*** *The growth function is defined for a hypothesis set $\langle$ by*

$$m_{\mathcal{H}}(N) = \max_{x_1,\ldots,x_N \in \mathbb{X}} |H(x_1, \ldots, x_N)|, \tag{2.4}$$

where $|\cdot|$ denotes the cardinality (number of elements) of a set.

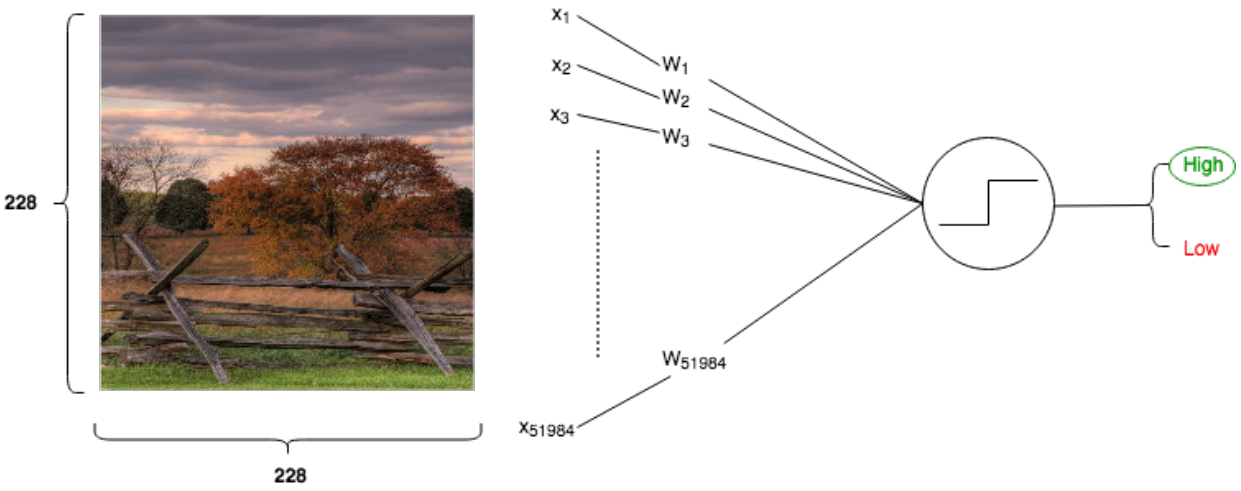


Figure 2.4: An aesthetic classifier example. The perceptron weights are represented by a $\mathbb{R}^{\{51984\}}$ vector. Each vector defines a hypothesis but when two distinct hypothesis lead to the same result, in this case the class *High*, the hypothesis are effectively the same from the sampled data point of view.

The growth function in definition 2.1 tells us that if there is at least one combination of datapoints $x_1, \ldots, x_N$ that will maximize the number of effective hypothesis, then that is the value of the growth function. When a hypothesis set $\mathcal{H}$ is able to generate all possible combinations of outcomes in a dataset of size N it is said that the hypothesis set *shatters* the $x_1, \ldots, x_N$ samples. For the binary output case the growth function is at most 2^N for a given N.

On the other hand, if a hypothesis set is not able to shatter any combination of N datapoints, then the number of samples N becomes the break point for the hypothesis set $\mathcal{H}$. In the binary output case the existence of a break point means that $m_{\mathcal{H}}(N) < 2^N$. The following definition defines the break point,

Definition 2.2 ***Break Point*** *For a given hypothesis set $\mathcal{H}$ and a number of samples N the break point is the quantity where the hypothesis set $\mathcal{H}$ is no longer able to shatter the N datapoints.*

The break point is an important parameter that relates to the complexity of a hypothesis set and its capacity to fit to the data. Hypothesis sets with higher break points are able to fit more datapoints. Moreover, in a hypothesis set with a break point, the growth function is used to replace M in the Hoeffding's Inequality and plays a key role restraining the effective number of hypothesis.

In fact if there is a break point *k* it is possible to show that the growth function is bounded by a polynomial. The following theorem shows the polynomial bound of a growth function from a hypothesis set with a break point,

Theorem 2.1 *If $m_{\mathcal{H}}(k) < 2^k$ for some value k, then*

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

for all N. The RHS is polynomial in N of degree $k-1$.

The theorem 2.1 is important because the growth function that replaces M in the RHS of the Hoeffding's Inequality may be seen as a polynomial if the hypothesis set has a break point. This realization makes it easier to work with the growth function because it no longer needs to be pinned down exactly and only with the break point it is possible to estimate how large the training set must be in order to ensure that E_{in} will be close to E_{out} with reasonable confidence.

The following equation shows the growth function replacing M in equation 2.3,

$$2 \cdot m_{\mathcal{H}}(N) \cdot e^{-2\epsilon^2 N}. \tag{2.5}$$

When the hypothesis set has a break point the growth function in equation 2.5 will be a polynomial in N of degree $k-1$. That means that the exponential will eventually damper the effect of the polynomial and lower the RHS of the Hoeffding's Inequality.

Now it is possible to define the VC Dimension,

Definition 2.3 *The Vapnik-Chernvonenkis dimension, d_{VC}, of a hypothesis set $\mathcal{H}$ is the largest value of N that can be shattered by the hypothesis set, i.e: $m_{\mathcal{H}}(N) = 2^N$ for the binary classification problem.*

The VC Dimension can be compared with the break point by the following equation $k = d_{VC} + 1$. The theorem 2.1 can be rewritten as:

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d_{VC}} \binom{N}{i} \tag{2.6}$$

Finally, with the definition of the VC dimension and how it relates to the growth function. It is possible to define a generalization bound replacing M in equation 2.3 and find E_{out}, taking into account the effective number of hypothesis in $\mathcal{H}$. The following theorem introduces the generalization bound known as VC generalization bound.

Theorem 2.2 (VC generalization bound) *For any tolerance $\delta > 0$,*

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(2N)}{\delta}}$$

with probability $\geq$ 1 - δ

The VC Generalization bound result is applicable to all kinds of input spaces, hypothesis set, learning algorithms and target distributions. Due to its generality, one might conclude correctly that it is a loose bound for the generalization problem, but it is still significant because it shows that it is feasible to learn a hypothesis while exploring an infinite set without rendering E_{in} meaningless. Moreover the bound is equally loose for all models, that way it is possible to compare different models while approaching the problem at hand. With this two factors in mind it is possible to employ the VC bound as a guideline for generalization.

2.6 TEST SET

The VC Bound is an amazing guideline during the training process but due to its looseness it gives us a poor estimation to E_{out}. To bypass this issue a common practical solution employed in real life situations is to split the dataset into two parts: a training set and a test set. The training set will be employed during the training process to update the weights of the model, while the test set will be used to generate the final estimation for E_{out}.

Since the test set is not part of the training process and the final hypothesis g is already chosen, the effective number of hypothesis when estimating E_{out} is just 1. Therefore the original Hoeffding's Inequality can be used as generalization bound instead of the VC bound. The following equation, derived from the Hoeffding's inequality, enables the calculation of the generalization bound:

$$E_{out}(g) \leq E_{test}(g) + \sqrt{\frac{1}{2N} \ln \frac{2}{\delta}}. \tag{2.7}$$

The choice for the Hoeffding Inequality gives a much tighter prediction for E_{out}. The eq. 2.7 shows that in a test set of 2.000 samples, the E_{test} value has a probability $\geq 99\%$ of beign within $\pm$ 3.7% from E_{out}.

2.7 FRAMING THE FEASIBILITY OF LEARNING

In section 2.5 the generalization problem was addressed and the concept of the VC dimension was introduced. This settled the problem and gave us tools to estimate E_{out} systematically. The following questions frame the feasibility of learning and provide a way to analyze the learning problem.

1 - Is it possible to ensure that E_{out} is close to E_{in}?

2 - How small can E_{in} get?

The first question was answered in the previous section but in order to address the second question two key concepts are introduced: regularization and validation. These two concepts complement each other and grant us the tools to find the model that better predicts the target function. Moreover, along with the VC bound, they allow us to know that even in the worst case, where the model was not able to capture the underlying distribution, we can still be sure that the best result in the chosen Hypothesis Set was found.

2.8 OVERFITTING AND REGULARIZATION

Overfitting happens when the model fits the data available so well that it no longer provides a good indication to the out-of-sample error. When the model is trying to find the best fit it will try to fit all datapoints in the training set with its available degrees of freedom, but the problem arises because most datapoints have some degree of noise called stochastic noise that leads the model off target. This can be seen when very powerful models are used with a small dataset. Moreover, another type of noise present in a learning problem is the deterministic noise and it is related to the target complexity. In more complex target functions the difference between the final hypothesis and the target function represents the deterministic noise. The graph in Figure 2.5, based on the image from the book "Learning From Data" from Abu-Mostafa *et al.*[2], captures a hypothesis in red that overfits the data available from the target function. The stochastic noise is represented by the offset in the data from the target function and the deterministic noise is the area between the red curve (the final hypothesis) and the blue curve (the target function).

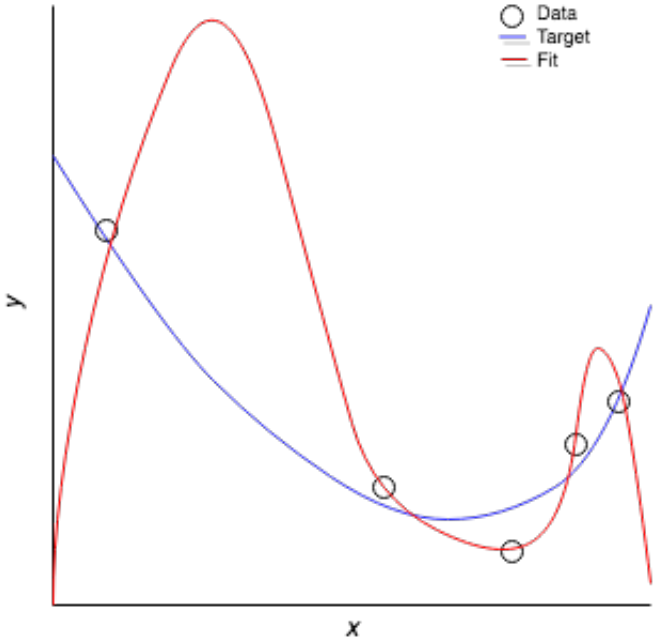


Figure 2.5: The model is complex enough to fit all data available in the dataset, but this greatly harms the E_{out} result.

Finally, the E_{out} error can be decomposed into three elements. These elements are shown in the following equation:

$$E_{\mathcal{D}}[E_{out}] = \sigma^2 + bias + var, \tag{2.8}$$

First, the noise, represented by σ, is the off set value between the target function and the data in figure 2.5. Second, the bias is directly related to the deterministic noise and represents the difference between the target function and the final hypothesis. Third, the variance captures how our model behaves with small variations to the data available.

Regularization is a set of tools employed to prevent the model of overfitting the data available, as shown in Figure 2.5. It works by setting constraints to the learning algorithm and favoring simpler models over complex ones, this may seem counter intuitive but, by employing regularization techniques, the harm done by the restraint is greatly surpassed by the benefits of not fitting the noise.

In a learning setting, the model learns by minimizing the in-sample error. This minimization is guided by a learning algorithm that evaluates how much a hypothesis is wrong and updating the model towards the correct answer. Distinct models have distinct learning algorithms, such as: PLA, Pocket and Gradient Descent. Nonetheless all this learning algorithms work towards the same goal, i.e: to find a set of weights that will make the hypothesis closer to the target. This set of weights are the effective responsible for the shape of the curve, like the shape of the red curve in Figure 2.5, therefore the in-sample must be a function of the weights. In fact, this idea is what allows the learning algorithm to work. For each type of a target function there are different choices of in-sample errors, for the case of a continuous target function, like the blue curve in Figure 2.5, a common error measures is the squared error measure represented by the following equation:

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(h_n(\mathbf{w}) - y_n)^2, \tag{2.9}$$

where $h_n(\mathbf{w})$ is the predicted value for the datapoint n, while y_n is the correct value, or ground truth, for the same datapoint n. By summing the difference between the predicted value and the ground truth for all N datapoints available in the dataset and dividing by N, we have the expected value for the in-sample error. This expected value is a function of the weights that control the shape of the hypothesis, therefore in order to prevent the model from overfitting the dataset, regularization techniques restraint the values that $\mathbf{w}$, in equation 2.9, can assume.

One of the most widely used regularization techniques is called weight decay. It adds a value to the in-sample error that prevents its weights to vary during the learning process. It works by making the learning algorithm trying to minimize an error augmented by this value instead of the original in-sample error. This is shown in the following equation:

$$E_{aug}(\mathbf{w}) = E_{in}(\mathbf{w}) + \lambda\mathbf{w}^T\mathbf{w}, \tag{2.10}$$

where the λ parameter is simply called *regularization parameter*. By choosing it correctly, it is possible to greatly improve the generalization error. In fact, Figure 2.5 illustrates a model, in red, overfitting the target function, in blue, while the same model with a $\lambda = 0.0001$ is depicted in Figure 2.6 performing much better. The Figures 2.5 and 2.6 illustrate the benefits of choosing a good regularization parameter. A good choice for the regularization parameter can make an algorithm succeed or fail and in order to achieve success it is necessary to introduce the concept of Validation.

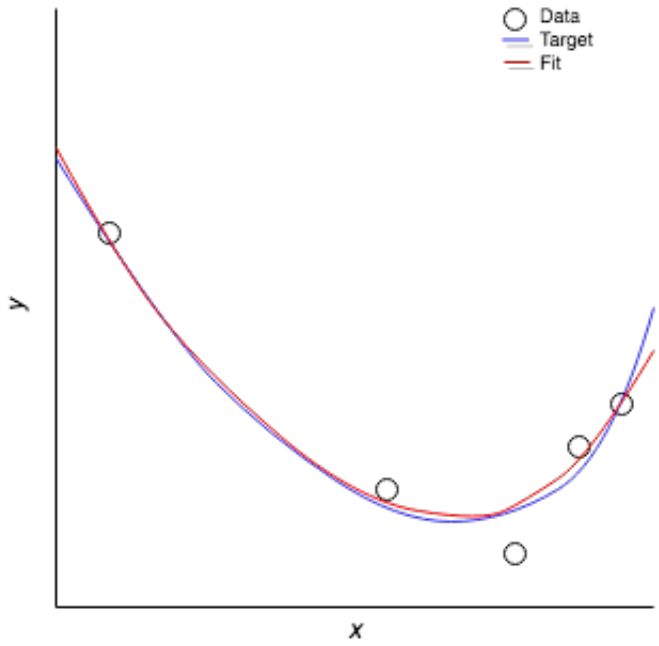


Figure 2.6: The regularized version of 2.5 with a regularization parameter equals to 0.0001.[2]

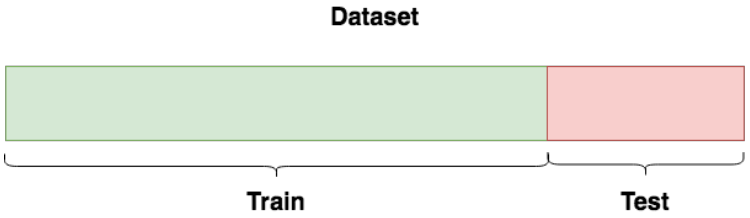
2.9 VALIDATION

In section 2.8, we introduce the idea of regularization to prevent a complex model of overfitting the dataset. To achieve that, an optimal value for λ was chosen. But in order to evaluate the effects of the regularization parameter, the target function responsible for generating the dataset was presented so it was possible to check that the hypothesis was approaching the target, in other words E_{out} was getting small. In a real learning environment this does not happen, there is no target function available to calculate the E_{out} value. This is where validation comes into play.

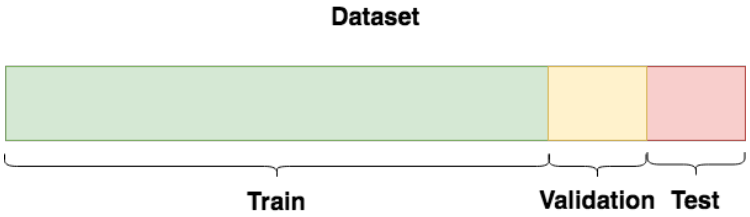
The Validation gives a proxy for the E_{out} by sorting out part of the training set before the training starts. This way the dataset is now divided three fold, instead of only into two parts. This new part is called validation set and Figure 2.7 illustrates the two types of division.

The division shown in Figure 2.7(a) made the use of the Hoeffding Inequality, which was described in Section 2.6, possible. Since the Test Set is never seen by the model during training, it works as a good proxy for the final estimation of E_{out}. But now a new set is necessary to estimate the value of E_{out} in order to choose the regularization parameters. The division shown in Figure 2.7(b) allows us to search the regularization parameter without meddling with the model while it is learning. This is important because if the model is changed during training by choosing the best regularization parameter, the VC dimension is no longer properly accounted for because we are helping the model to learn. This way the VC Bound gets looser and we no longer are able to make any statement about E_{in} and E_{out}.

The validation set allows for any one to perform *model selection* without the fear that the in-sample error does not follow the out-of-sample error. Model selection is the generic term given



(a) Training and Testing set division.



(b) Training, Validation and Testing set division.

Figure 2.7: The Validation Scheme further splits the dataset into a new part that will be used to perform the validation of the model. The two figures depict two distinct split schemes.

when parameters like the regularization parameter are searched outside the training step, moreover is possible to search through completely different models, given that they are minimizing the same error and training with the same data, to be selected during model selection and choose the one that better generalize. The parameters like the regularization parameter or the dropout rate (another regularization technique employed in Neural Networks) are commonly called Hyper Parameters. They receive this name to contrast with the weights of the model, that are considered Trainable Parameters.

2.10 MACHINE LEARNING MODELS

After introducing important concepts about machine learning, how one is able predict something outside the dataset and how models can be selected to generalize better it is time to introduce the models that are employed in the aesthetic assessment problem.

2.10.1 Neural Networks

The Neural Networks are a class of models that received great attention on the last decade due to their capability to model complex functions and their applications in time series predictions and vision problems. Besides their capability, neural networks have great learning algorithms to fit the training data that allows for a quick computational optimization.

This type of model is based on the Perceptron, more precisely, the Multi-Layer Perceptron, which can approximate more complex functions than the Perceptron alone. For example, no single perceptron is capable of learning an hypothesis boundary such as the one in Figure 2.8

depicted as the black lines that separate the -1 region, in purple, from the +1 region, in green. The black lines represent the final hypothesis boundary between the classification regions of the model, in purple and green. On the other hand, a MLP, like the one in Figure 2.9, is capable of learning such boundary by combining Perceptrons and employing a Perceptron Learning Algorithm (or PLA).

The first layer in the MLP is known as the Input Layer, while the second and third layers, represented by the Perceptron 1 and the Perceptron 2, form the Hidden Layers. At last, the final circle, or node, represents the Output Layer that is responsible to output the outcome as +1 or -1 in Figure 2.8.

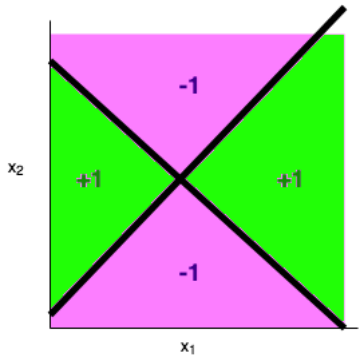


Figure 2.8: The boundary delimited by the black lines represents a XOR Bound. It divides the input region into two regions representing the output of a model. A simple model such as the Perceptron is not capable of drawing complex lines like these, therefore MLPs are used to model more complex bounds.

The main issue behind the employment of the MLP lies on its $sign(\cdot)$ nodes, represented by the graphs inside Perceptron 1 and Perceptron 2 nodes. Because of this type of node the Learning Algorithm employed by the MLP becomes a hard combinatorial optimization problem that requires a large amount of computational power as the MLP gets larger. A more smooth, differentiable function that approximates $sign(\cdot)$ allows for an easier optimization problem and this is where Neural Networks emerge.

The Neural Network model employs a soft differentiable function, the hyperbolic tangent $\theta(x) = tanh(x)$, in place of $f(x) = sign(x)$, that allows for a more efficient learning algorithm, the *gradient descent*. The gradient descent is a technique for minimizing a twice-differentiable function, such as the in-sample error, E_{in}, represented by the difference between the hypothesis output, $\mathcal{H}(x)$ in figure 2.10, and the ground truth. This technique updates the weights, represented by the arrows in figure 2.10. Each arrow represents a linear operation between the output of the previous node and a trainable weight. For example, in the case for the second node in first hidden layer its output is represented by the following equation:

$$y = \theta(x) = tanh(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_d \cdot x_d). \tag{2.11}$$

Each layer ℓ has its own weights matrix W^ℓ with dimensions $(d^{(\ell-1)}+1) \times d^{(\ell)}$, where $(d^{(\ell-1)}+1)$ is the number of nodes in the previous layer and $d^{(\ell)}$ is the number of nodes in current layer.

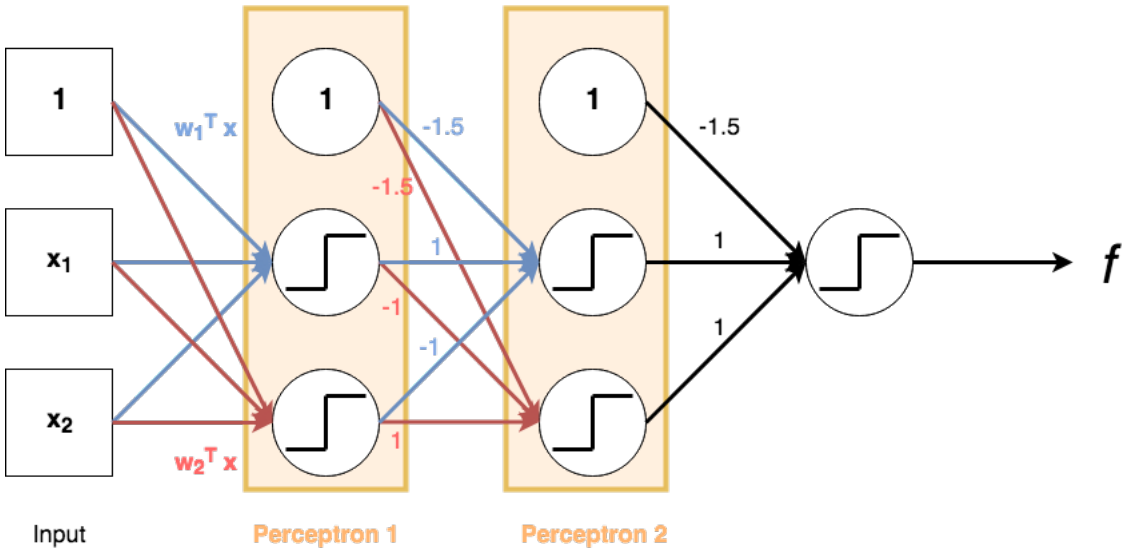


Figure 2.9: The Multi-Layer Perceptron is more complex than the Perceptron. It is able to model more complex functions by stacking Perceptrons into different layers. The Perceptrons in the MLP are represented here by the rectangles with each circle representing a node that will perform a linear operation between the values from the previous layer.

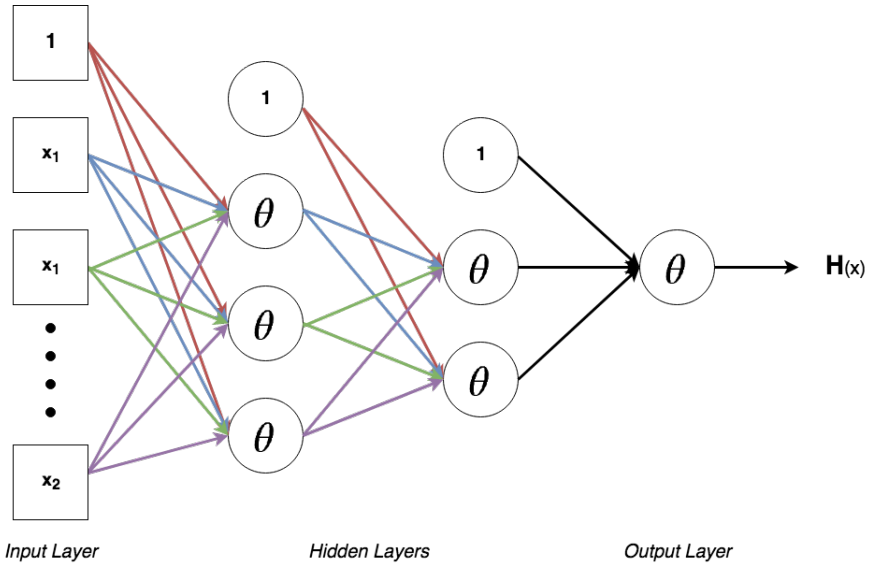


Figure 2.10: The neural network is composed of layers of nodes, represented by the circles, that perform a linear operation on their input signal. Later on, an activation function is applied to each node and the results propagate to the next layer.

In the first hidden layer in Figure 2.10, the weight matrix connecting the previous layer (the input layer in this case) and the first hidden layer would have a dimension of $(d+1)\times 3$. Collectively, all weights in a Neural Network can be represented by $\mathbf{w} = \{W^1, W^2, \dots W^L\}$, where L represents the total amount of layers without counting the input layer in a Neural Network. In the case for the Neural Network in Figure 2.10, L is equal to 3.

In order to update the weights, and effectively learn, the gradient descent updates them by making them take step in the negative gradient direction, as seen in the following equation:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E_{in}(\mathbf{w}(t)). \tag{2.12}$$

The value $\mathbf{w}(t)$ represents the current value for the weights, η represents the Learning Rate and $\nabla E_{in}(\mathbf{w}(t))$ is the gradient of the in-sample error in regards to the weights of the network. As previously mentioned in Section 2.8, the in-sample error depends on the target function being learned.

2.10.2 Convolutional Networks

The development of the convolutional layer was one of the greatest steps in the computer vision field. Although the roots of the development of convolutional layers may be traced as far back as 1959 in the works of David Hubel and Torsten Wiesel [25], in which they analyzed how the brain process images, later works were responsible to introduce the convolutional layers as they are known today. In 1980, the researcher Dr. Kunihiko Fukushima first introduced in his work [26] the Neocognitron Network. In his work, the modern approach for a convolutional layer was first described but only later in 1998 Dr. Yann LeCun [27] would introduce the implementation of a learning algorithm to train a network with convolutional layers.

2.10.2.1 How Convolutional Layers Work

Much like the Neural Networks evolved from the MLP models, the Convolutional Networks can be seen as a evolution of the Neural Networks. The convolutional layer performs, as the name implies, a convolutional operation between two matrices, as depicted in Figure 2.11 where a section of the black and white picture is represented as the binary input matrix. The convolutional operation works as a linear operation between the weights in the filter and the overlapped section of the image. The filter matrix then moves across the image and builds a descriptor of the image and ideally this descriptor will reflect important features of the image. The filter represents the Weights Matrix from the Neural Network that, just like in the Neural Network, will be randomly initialized and will be updated through gradient descent.

The motivation to employ this convolutional approach lies on the structural quality of an image. When perceiving an image, a human being does not notice each pixel individually, but rather the image as a whole. Therefore, the use of filters along with a convolutional approach

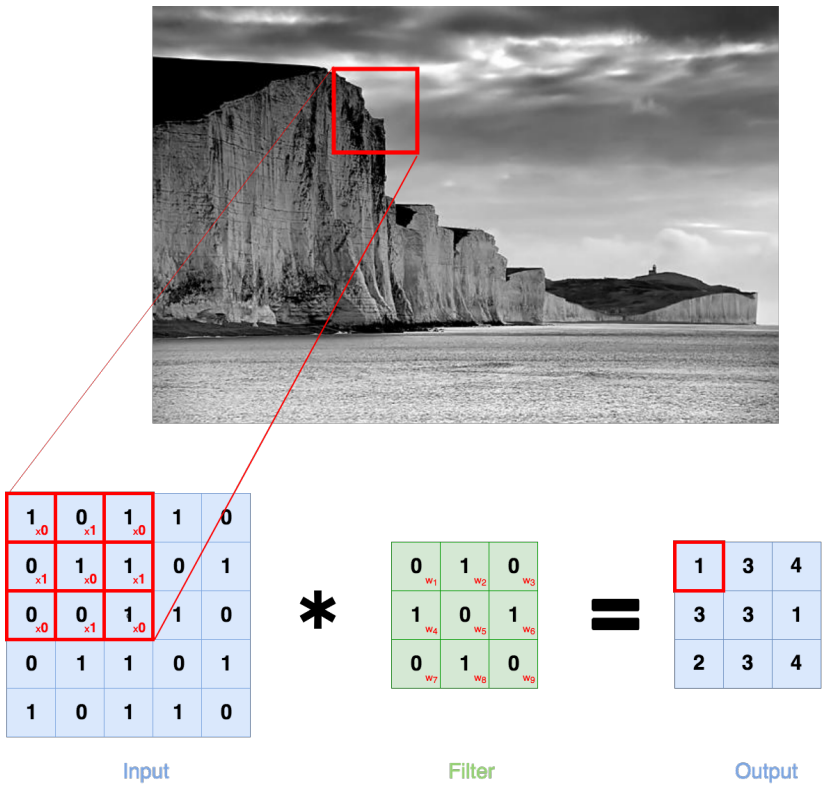


Figure 2.11: The convolutional layer is able to capture the spatial relation between the pixels of an image. In this image each block in the input represents a pixel, while the filter is depicted as the green boxes.

allows the network to capture the spatial relationship between the pixels and improve the filters at each step of the gradient descent. The Figure 2.11 depicts a section of the image as input along with the filter and the result of the convolution between them. The filter sweep the whole image creating a descriptor of the original image as a result. Moreover, the concept of digital image filters is the basis of digital image processing, where filters like the Mean Filter, the Gaussian Filter and Edge Detection filters, as the ones depicted on figure 2.12, are common place, this further supports the idea of employing convolutional layers to extract features.

Just like traditional filters, the filter in a convolutional layer will try to learn the weights in order to detect features by employing a backpropagation algorithm along with gradient descent.

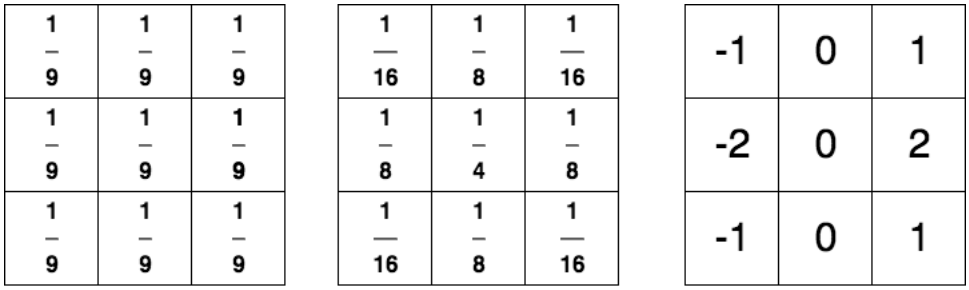


Figure 2.12: From left to right: Median Filter, Gaussian Filter and Vertical Edge Detection Filter. Traditionally, filters were handcrafted based on the characteristics being searched but, as soon as the characteristics searched became more and more complex, the use of feature detectors presented an efficient solution for high-level analyzes.

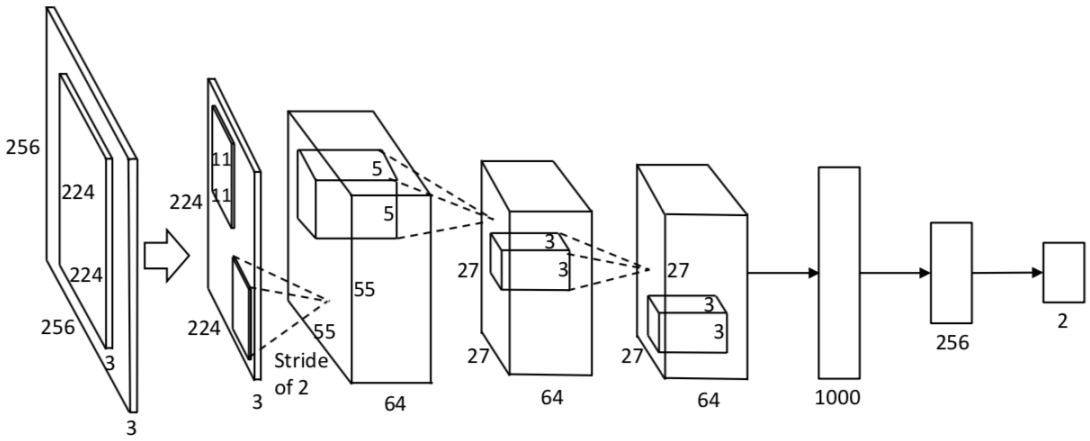


Figure 2.13: An example of a typical CNN architecture scheme with a 256x256x3 input and a two-label output [29].

The idea is depicted in Figure 2.14; from Courville, Goodfellow and Bengio book *Deep Learning* [28], where each layer learns more complex features than the previous one, like edges and corners all the way to faces and objects. Each layer takes advantage of the previous features to build a new image "view".

Finally, after the image flows through the convolutional layers, it is flattened and a fully connected layer is employed to perform a classification or a regression, depending on the task goal. This configuration is illustrated in Figure 2.13 , where the blocks represent the convolutional layers and the rectangles the fully connected layers . For this reason the convolutional layers are sometimes seen as Feature Extractors, because they learn the most relevant features by a learning algorithm, and why Neural Networks are seen as Classifiers, as a generic term to highlight their goal to take the extracted features and perform some classification or regression.

2.10.3 Modern Convolutional Networks Architectures

Four distinct Convolutional Networks will be employed in this work, they are: AlexNet [30], VGG16 [31], ResNet [32] and MobileNet [33].

The first one, AlexNet, is arguably the architecture that galvanized the interest in Convolutional Networks among the Computer Vision community. It was introduced by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) and it won the challange by achieving a top-5 error, where the predicted class for the input image is not in the top-5 classes predicted by the model, of 15.5%. The second place at the time achieved 26.2%. The architecture is depicted on figure 3.3 and by all means is the simplest architecture of all four aforementioned. Nonetheless Krizhevsky *et al.* popularized important techniques in computer vision, such as Dropout and Image Augmentation, besides the use of ReLU (Rectified Linear Unit) in place of the TANH function on the activation nodes. The use of ReLU was found to decrease the training time significantly while also introducing nonlinearities to the model that helps it learn more complex target functions.

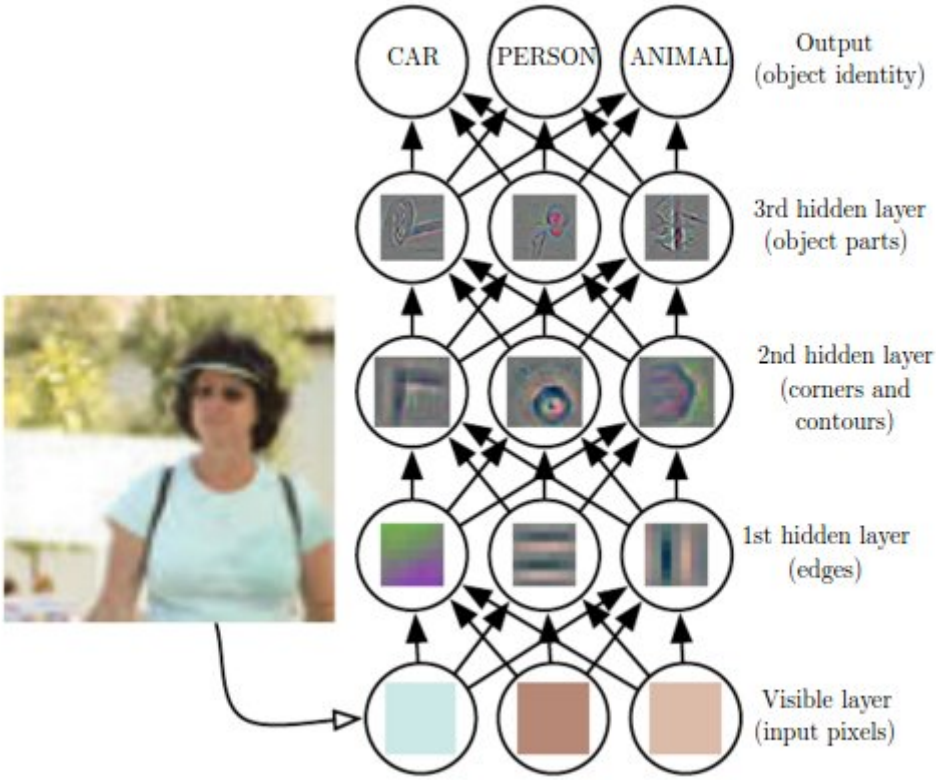


Figure 2.14: As the model gets more and more layers, at each one of them, the filters are learning more complex features.

The second one, VGG16, built upon the ideas from AlexNet and was introduced by Karen Simonyan and Andrew Zisserman. They expanded on the idea of employing convolutional layers and created a network with 16 layers, 13 convolutional layers and 3 fully connected layers, as seen in Figure 2.15. The VGG16 was capable of achieving a top-5 error of 7.3% in the ImageNet challenge, showing that the error rate could be improved by increasing the depth of the architecture.

The third model, ResNet, is the deepest model of all four, containing a total of 152 layers. The model was able to achieve a top-5 error of only 3.6% in the ILSVRC 2015 and win it. The model was developed by the Microsoft Research Asia team and its main contribution is what they called, Residual Block. With this many layers a problem arises called *Vanishing Gradient*, where the gradient of the loss starts to shrink to zero as it approaches the beginning of the model during backpropagation. This problem happens on sigmoid activation functions used on the layers nodes because they squeeze the input values to have values between 0 and 1, therefore the impact of the derivatives during backpropagation are diminished at each layer. This diminishing effect leads to very low gradient values to arrive in the initial layers affecting their training. The Residual Block proposed in [32] introduces a identity path, shown in Figure 2.16, between layers that allow for the gradient to flow back with more ease preventing vanishing gradient effect.

The last model is the MobileNet. This model was introduced by researchers at Google and

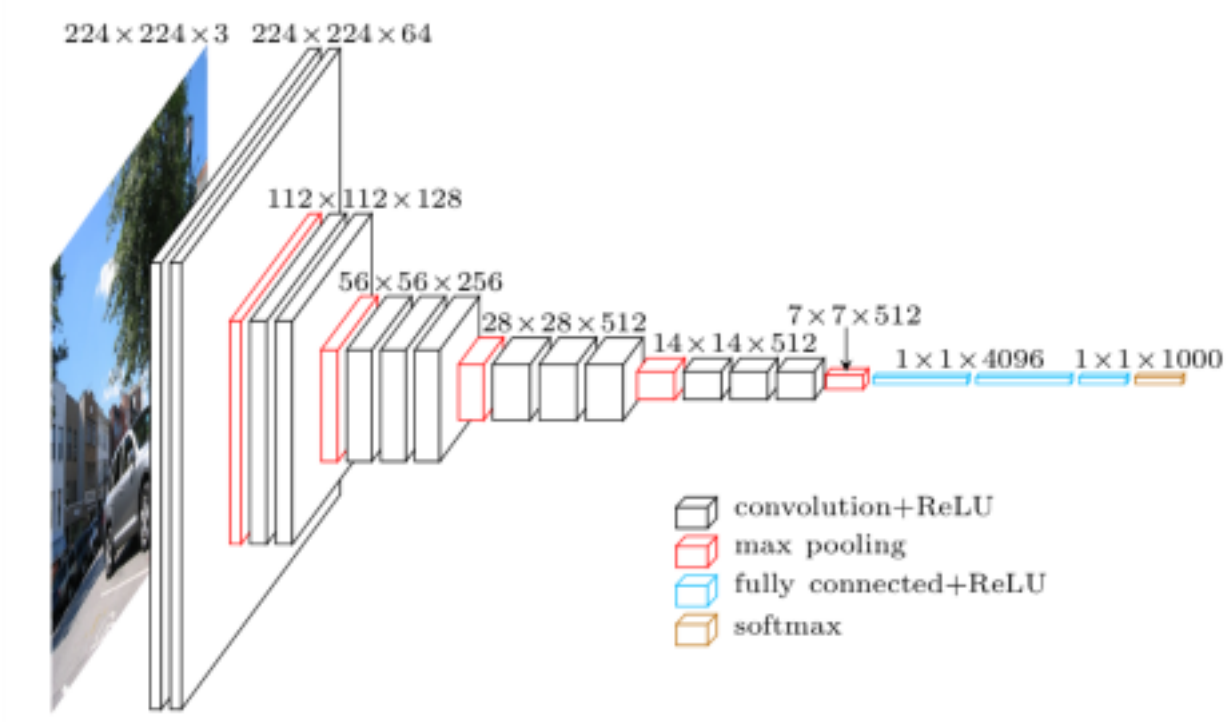


Figure 2.15: The VGG16 architecture displayed the benefits of employing a deeper architecture.

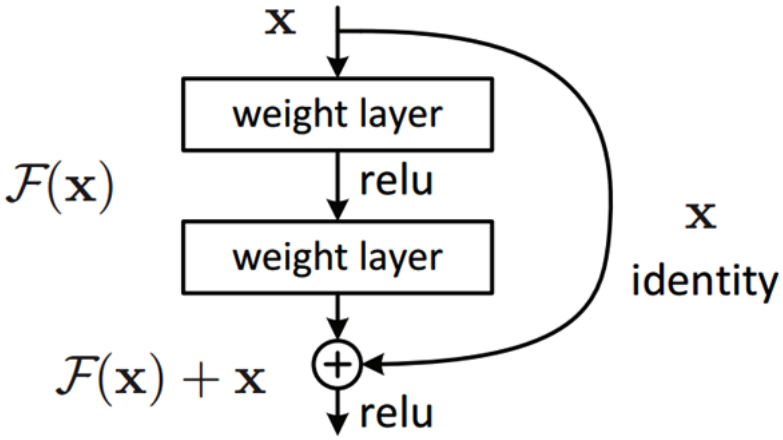


Figure 2.16: The Residual Block creates a residual mapping between the original input and the output of the convolutional layers.

its main contribution was the use of width multipliers and resolution multipliers to control the input width and resolution of each layer, respectively. This addition allowed the architecture to be much smaller than the previous ones while still performing better than other state-of-the-art architectures at the time it was launched. Moreover, the number of multiplications and additions are much lower than the previous ones allowing it to be employed in mobiles and embedded systems.

3 AESTHETIC ASSESSMENT SYSTEMS

In this chapter, we discuss the basic concepts necessary to understand and design an aesthetic assessment system based on machine learning techniques. More specifically, we describe current challenges, CNN architectures and the performance metrics, and present the basic CNN architecture in which the proposed methodology is based on.

3.1 INTRODUCTION

As depicted in Figure 3.1, an aesthetic assessment system can be divided into two parts: a feature extraction step and a decision step. The feature extraction step, as the name implies, is responsible for analyzing the image and extracting the features that better represents its aesthetic properties. During the decision step, this feature representation is used to assess the aesthetic value of the image.

Two types of feature extraction step can used to extract aesthetic features that will form the image representation. The first category comprises the traditional hand-designed extractors that extract low-level features, such as color histograms and salience maps, and high-level features, such as the golden ratio and the image composition. The second category comprises deep-learning

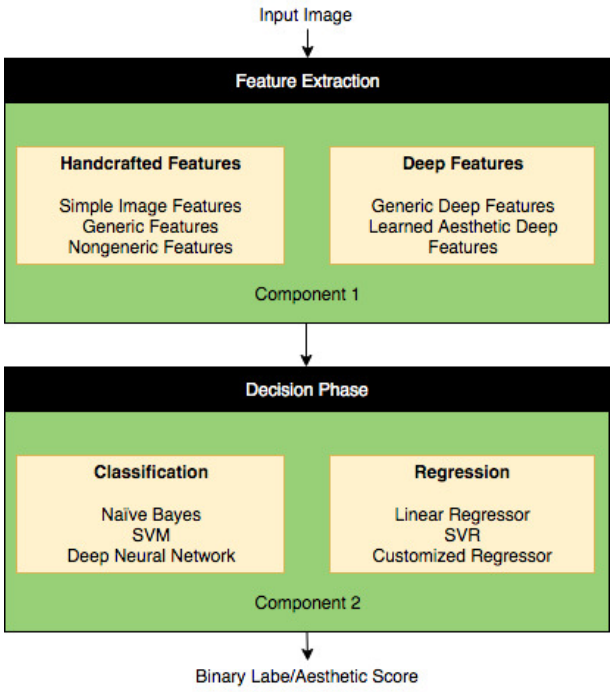


Figure 3.1: The pipeline shows the main two steps of an automatic aesthetic system, which are the feature extraction and the decision steps (adapted from [34]).

(DL) based methods that employ different types of neural networks to generate a feature vector that represents the image aesthetics. We refer to these two categories as Handcrafted Features and Deep Features approaches.

The decision step, on the other hand, can be either a classification system or a regression system. In a classification system, the objective is to classify an image into classes, such as high-aesthetic-quality or low-aesthetic-quality [23] [35], or to classify the contents of an image, for example images containing animals, people or landscapes. Regression systems aim to provide a score or a value to an image based on the features given by the previous step. The scores can reflect different aspects of the image, like the aesthetic appraisal of an image or its visual quality [36]. The choice between using either a classification or a regression system depends on the research problem addressed.

Finally, the system performance is evaluated using metrics, such as accuracy, F1 score or KL divergence. The designer of the system chooses the metric that is able to better illustrate the performance of the system to achieve the final goal. In a classification problem, for example, it is common to employ accuracy to determine the number of correct classifications compared to the total number of examples.

3.2 CHALLENGES

One of the first works to undertake the goal to evaluate how different features affect the aesthetic quality of an image was the work of *Datta et al.* [23]. In their work, the authors enumerate a number of challenges faced by automatic aesthetic systems. In this section, we discuss some of these challenges.

The first challenge faced by these systems is the lack of diverse datasets. This has a direct impact in the development of new algorithms and, most importantly, in their ability of generalization. One very popular dataset is AVA [17], which contains photos extracted from the site *Photo.net* and is mainly composed of photos taken by professional photographers or amateurs with a reasonable understanding of the photographic rules (i.e. a better understanding than the average person). Although this dataset was used by several authors, including Datta *et al.* [23], these factors generates an implicit bias in the aesthetic quality of the images that is caused by the background of the photographers and of the participants (photographers and amateurs) that evaluated the images. Because of this bias, it is important to bear in mind the limitations of an automated evaluation system designed using only the AVA dataset. To better generalize the designed system and improve its overall performance, researchers commonly use specific datasets to train the system and cross-validate it using data from a different dataset.

The task of explicitly hand-designing a machine learning model that is able to differentiate images with different aesthetic values requires a substantial amount of expertise from the designer. During the design, many different factors have to be taken into account, like for example what

are the most important visual features; how they interact with each other; and which ones are the most perceptually relevant. Also, the choice of visual features is often motivated by well known photographic rules, which has its advantages and disadvantages. For example, the reasons behind a certain photographic rule are often unclear and, therefore, it is hard to computationally model them. This problem often turn systems based on handcrafted features into simple predictors of the photographic rules that influence the aesthetic value. In their work, Datta *et al.* [23] studied how several visual features, such as light exposure, colorfulness, saturation, hue, and composition, affect the performance of an automatic aesthetic assessment system system.

In the past two decades significant advances were made in the machine learning field, which also contributed to the development of new (and better) image processing algorithms. In particular, deep-learning based methods have shown great results for data-driven approaches [29, 34, 37–39]. Additionally, the use of transfer learning [34] to develop new algorithms has been very effective, specially in cases where there is no large datasets that allow training the system from beginning to end. Nevertheless, the use of deep-learning in the development of automatic aesthetic assessment methods also faces challenges. One of these challenges lies on the difficulty to interpret the features being learned by the hidden-layers. This difficulty prevents researches on gain relevant insight that could improve the results of the models. Even though much effort was made to develop good practices [2, 28, 40], the interpretability of deep-learning models is an area that still needs more research and could greatly contribute to better results.

3.3 DATASETS

In this section, we describe the available datasets for image aesthetics, detailing their advantages and disadvantages.

3.3.1 AVA Dataset

The AVA (Aesthetic Visual Analysis) dataset was first introduced by Murray *et al.* in their paper *AVA: A large-scale database for aesthetic visual analysis* [17]. The AVA dataset is comprised of images extracted from the DPChallenge website [41]. The images are submitted by amateurs and professional photographers to the website and are subject to peer-review by other users. The users score the images on a scale of 1 to 10, according to how much they like the image. Moreover, the images can be submitted to different challenges organized at the website. These challenges have different themes and brief descriptions to guide the photographers, such as: *Artificial Lighting* and *Independence* depicted on figure 3.2, and the submitted images are scored based on their adherence to the theme besides their technical quality. All these features provide a richness of metadata that can be used in aesthetic assessment research.

The AVA dataset contains over 255,000 images, with an average 210 numerical voting scores per image, with its less voted image having 79 votes. Moreover, approximately 200,000 images

have at least one semantic tag (e.g, persons, science and technology, nature, etc.) describing the content of the image, and around 11,000 images have photographic styles tags (e.g. portrait, back and white, landscape, etc.). With respect to the statistics of the votes , 62% of the images have a Gaussian-like distribution of votes [17]. This behavior is observed on images with mean score vote value that is closer to the mean score of the dataset. Images with mean scores closer to the limits of the rating scale, have approximately a Gamma distribution of vote, with values skewed closer to the corresponding limit.

3.3.2 CUHKPQ

The CUHKPQ is a dataset assembled by Luo, Wei, Xiaogang Wang, and Xiaoou Tang [42] to serve as benchmark for their work. Their work analyzed how regional and global features affected the perceived aesthetic values of images. It was inspired by the idea that photographers employ different techniques depending on the nature of the content. This dataset was assembled with images taken by professional and amateur photographers and published in photography websites. The images were divided into seven distinct classes, according to their content: animal, architecture, human, landscape, night, plant, and static. The images were classified by ten observers as having either a high or a low aesthetic quality. However, an image would only be added to the the dataset if eight out of the ten observers agreed on these aesthetic classification of the image, e.g: only if an image received eight positive or negative votes it would be added to the dataset. . From the total amount of images gathered by the team, around 40% of them were excluded due to a divergence among observers, leaving a total of 17.673 images distributed among the seven distinct classes. Table 3.1 summarizes the distribution of images per class and aesthetic value (high or low). Notice that there is a larger number of low (aesthetically) quality mages than of high quality images.

Scene / Aesthetic	Animal	Architecture	Human	Landscape	Night	Plant	Static
High Quality	953	595	678	820	353	594	531
Low Quality	2,292	1,290	2,470	1,950	1,356	1,803	2,005

Table 3.1: Distribution of images in the CUHKP dataset among the different classes and aesthetic values.

As seen in Table 3.1, there is a great bias towards the low aesthetic class since there are three times more low quality images than the high quality ones. Although the CUHKP has a smaller number of images, when compared to the AVA dataset, the criterion by which the images were selected makes this a more reliable dataset, reducing the intrinsic dataset noise.

3.4 STATE OF THE ART

During the past decade, machine-learning techniques gained a lot of popularity due to their impressive results in competitions like ILSVRC. In this competition, deep-learning methods were able to achieve better results in object classification challenges than humans, a task that was previously considered impossible. Moreover, the past decade saw an increase in computational power that allowed researches to experiment with more complex models faster than before. Along with impressive results this sparked the interest of many research groups, including computer vision and image processing groups In this section we explore the main studies in image processing employing machine learning that inspired our work.

Datta *et al.* [23] have analyzed how the different visual features of an image and its adherence to photography rules, such as the rule of thirds, impact the perception of the aesthetic value of the image. This work is the first work that tried to evaluate the subjective aspects of aesthetics in a pragmatic way and to understand their individual contributions to the perceived image aesthetics. In their work, they use the *Photo.net* [21] as a data source for the photographic images. *Photo.net* contains images taken by amateur and professional photographers, who can upload their work and have it evaluated by their peers. The images are scored based on their *aesthetics* and *originality* and any user can cast a vote on any image based on this two factors. The votes range from 1 to 7, where 7 stands for a high-quality image and 1 to a low-quality image, then all votes are averaged to give an image its final score. In their experiments, only the aesthetics scores were used as label during training.

In their study, they built a classifier that performed a binary classification of the images, between high and low aesthetic value. They also employed a regression algorithm to predict the aesthetic score of a given image. But the main goal behind these classification and regression algorithms was to gain insights into the most important features that influence the aesthetic perception of an image. For this, image descriptors were hand designed to help better describe the image and, consequently, better understand their main characteristics. The authors argued that this approach would allow them to break down how the features affect the outcome, unlike the deep-learning approach that learn features by itself that are not easily interpretable.

The first step on the system proposed by Datta *et al.* is the feature extraction step. In their work, to establish what are the characteristics of a high-quality aesthetic image, they present a series of features that are perceived (by viewers) as important. These features are: light exposure, colorfulness, saturation, hue, rule of thirds, familiarity, texture, size, aspect ratio, region composition, depth of field, and shapes. For each one of these features, an algorithm is proposed to extract the corresponding image feature. The algorithms extract low level features and the use pixel information and distribution to compose the representation of each feature. For example, the average pixel intensity represents the light exposure, while the image colorfulness is represented by the Earth Mover's Distance [43] applied to the relative color distribution histogram. The total amount of features extracted in the experiment was 56. Each one of the features have its value normalized to the [0,1] interval to prevent that the decision algorithm privileges features

with large range values.

In the decision step, Datta *et al.* choose SVM and Classification Trees to perform the classification. To perform the regression they choose a Regression Tree. They tested and trained their model using a 5-fold cross-validation.

The 56 features extracted from the images were evaluated individually to verify how well each feature is able to describe the image aesthetic and, therefore, how well it is able to classify the images into having a high or low quality aesthetics. The best result, when a single feature is used, corresponded to a 59.3% accuracy. Then, the authors applied a feature selection method to identify the most relevant features for the problem. The top 15 most relevant features were then combined to produce a classifier, which achieved a 70,12% accuracy. Later on, the researchers combined all 56 features and the Classification Tree achieved an accuracy of 62.3%.

Even though the result using all 56 features was worse, the algorithm provided interesting insights on how each feature might affect the aesthetics perception of an image. Some features, like the depth of field (denoted f_{54} and f_{55}) and the familiarity of an image (denoted f_9), presented very low losses (up to 9%) on their nodes. This implies that these features seem to be very important when assessing an image aesthetic. These results are interesting and encourage further analyzes on how different images attributes might influence the perceived image aesthetic. Also, we wonder if there is a correlation between the performance of certain features and the photography attributes.

A Convolutional Neural Network (CNN) architecture, or some of its many variations, is the go-to architecture employed by many computer vision or image processing applications. Figure 2.13 shows as an example of a typical single-column CNN that takes a 256x256x3 input and has a two-label output. The block-like components represent the convolutional layers and the rectangles represent the fully connected layers. For example, the work of Lu *et al.* [29] uses a CNN, trained on the AVA dataset, to classify images into high and low quality images [17]. Their work uses the style tags to simultaneously take advantage of what they call global and local views. This approach is inspired by the notion that an image appraisal can be assessed considering both its global composition and its individual elements. The global view represents the normalized image input, while the local view is a random crop of the original image. The architecture employed in this study has a fixed input size, which makes it possible to learn better systems parameters. Also, a fixed input size is useful because the global view must be normalized to a fixed size. The authors propose different normalization techniques, such as center-crop, warp, and padding. Then, they analyze the performance of these techniques using cross-validations tests. The best performing normalization techniques are used to design a double-column CNN that takes the global and local views as input. Moreover, the authors analyze the influence of the style tags on the performance of their proposed architecture. Their system was able to classify each image in a very short period of time (around 1.5 seconds), which can be further reduced by employing better hardware. Their work presented interesting results. For example, they observed an improved system performance when the style tags are used together with the proposed double-column architecture. In this work,

we have used a similar idea to design the proposed methodology that is described in Chapter 5.

Talebi *et al.* [44] evaluated if it is possible to use a previously trained CNN to train an automatic quality aesthetic assement model for images. Their model, named NIMA (Neural Image Assessment), blindly (without any reference) estimates the aesthetic and technical qualities of an image. Moreover, their model aimed at predicting the distribution of votes received by an image. The authors analyzed if this approach has advantages when compared to simply trying to predict the image average vote. It is worth pointing out that many previous studies approached this task as a classification problem with two classes: high and low aesthetically pleasing images. Previous studies did not provide a prediction of more meaningful statistics. Notice that predicting a distribution gives interesting information about the image, like the standard deviation of the votes, which tells the general agreement of the viewers about the quality and aesthetics of specific images.

The novelty introduced by Talebi *et al.* is the use of a "classifier" to predict the distribution of votes. The distribution is represented by a mass function $p = [p_{s_1},...,p_{s_T}]$, with $s_1 \leq s_j \leq s_N$, where s_j represents the jth score bucket and T represents the total number of score buckets. To predict the distribution, the authors selected state-of-the-art classification architectures i.e., VGG16, Inception V2 and MobileNet. These architectures were trained on the ImageNet dataset, with their last layer being replaced by a Softmax activation layer of 10 neurons with randomly generated weights. The 10 neurons represent the total number of vote buckets, T, of the AVA [17] and TID2013 [18] datasets.

The researchers also employed an EMD-based loss function to take into account the inter-class similarities, since predicting a score closer to the ground-truth is better than predicting a further one. This type of loss function rewards these predictions, where a Softmax Cross-Entropy punishes misclassifications equally, ignoring the proximity to the ground-truth and losing important information inherent to the nature of the scores.

Moreover, Talebi *et al.* applied well known regularization techniques, namely Dropout and Image Augmentation, along with Momentum, a technique to speed up the learning process. The first regularization technique, Dropout, randomly deactivate certain nodes in a chosen layer of the model, by doing this the model become effectively simpler at each training batch and helps the final model mitigate overfitting. The second technique, image augmentation, applies transformations to the images, such as affine transformation, image flip and noise-addition. This transformations create new images to the model and expand the available training samples. Finally, momentum allows the optimization algorithm, such as the SGD, to take into account the previous gradient directions estimated in previous batches of images, this in turn helps accelerate the gradient vector faster towards the optimal value.

The models were implemented in TensorFlow, with 80% of images used for training and 20% used for testing. A few hyper-parameters were found to improve the training process. The learning rate at the convolutional layers was set to $3 \cdot 10^{-7}$ and to $3 \cdot 10^{-6}$ at the FC layers. A exponential decay factor of 0.95 was used for all learning rates, after every 10 epochs. The

Name	Description	Values
Learning rate	Convolutional Layers	$3 \cdot 10^{-}7$
	Fully Connected Layers	$3 \cdot 10^{-}6$
Exponential Decay		0.95
Weight and Bias Momentum		0.95
Dropout		0.75

Table 3.2: Parameters of the CNN architecture used by Talebi *et al.*

weight and bias momentum were set to 0.95 and a dropout layer was added before the last 10 neurons layer with a 0.75. The optimization method chosen was the SGD. Only a horizontal flip was applied as data augmentation, since other types of operations could change key image characteristics that affect the aesthetics, like for example image composition and rule-of-thirds. The experiment setup is summarized in Table 3.2

The results obtained by Talebi *et al.* are comparable to state-of-the-art algorithms. Nevertheless, it is worth pointing out that previous works were trained and tested using the AVA dataset, considering a binary classification of the aesthetic quality (high or low quality). Consequently, a 2-classes accuracy metric was employed by the majority of these works. NIMA, on the hand, aims at a more complex prediction. Since NIMA was also trained on TID2013, it also used correlation coefficients, such as LCC (Linear Correlation Coeficient) and SRCC (Spearman's Rank Correlation Coeficient), as performance metrics. The best result of NIMA on the TID2013 dataset achieved a LCC and a SRCC of 0.941 and 0.944, respectively. This value is slightly lower than the state-of-the-art model by Bianco *et al.*, named DeepBIQ, that achieved a LCC and a SRCC of 0.96 and 0.96, respectively.

Talebi *et al.*' models were later employed to steer two enhancement operators, called the multi-layer Laplacian technique and Turbo denoising. Both these operators contain parameters that control different aspects of an image, such as shadows, brightness, and smoothness. These parameters are selected based on the prediction performed by NIMA. An analysis of the images enhanced by this pipeline resulted in interesting conclusions. While performing smoothing employing the Turbo denoising operator, images with a high level of detail received a lower degree of smoothing. On the other hand, images with lower level of detail received a greater degree of smoothing by the same method. This difference suggests that the NIMA method was capable of recognizing this features on the image and, moreover, learn to preserve images essential features even when analyzing images with different levels. In the next section, we describe the architecture in which the proposed work is based on.

3.5 MULTI SCENE DEEP LEARNING MODEL (MSDLM)

The MSDLM is a deep learning model introduced by Wang *et. al.* to predict the aesthetic score of an image. Their experiment is divided into two steps: pre-training and training. During the pre-training step, the authors train a model based on *AlexNet*[30], as depicted in Figure 3.3. In this model, a new convolutional layer, named Scene Layer, is used in place of AlexNet's last convolutional layer (between the feature extraction section of the model and the classification section of the model), as depicted in Figure 3.4. The authors trained a model for each class (types of images) of the CUHKPQ dataset, which predicts whether an image has a high aesthetic value, represented by the value 1, or a low aesthetic value, represented by the value 0. After the models are pre-trained for each class separately, the weights from the Scene Layer are saved to be used during the training step. The assumption here is that Scene Layer works as a good descriptor of the corresponding CUHKPQ's class and is able to capture its main high-level features.

During pre-training, the weights of the feature extraction convolutional layer are initialized from the AlexNet's weights, while each Scene Layer and Fully-Conected Layer are randomly initialized for each different class. During the original pre-training, the Learning Rate for the original experiment was set to $1 \cdot 10^{-4}$ for the feature extraction layers, while the other layers Learning Rates were set to $1 \cdot 10^{-3}$. Moreover, a Learning Rates decay of 40% was performed every 8 epochs

During the training step, a model based on AlexNet is employed again with slight adjustments to its architecture. Instead of a single Scene Layer, after the feature extraction section, seven Scene Layers are connected between the feature extraction section and the classification section of the network, as depicted in Figure 3.5. They are connected in parallel between the two sections and a mean-pooling layer connects the seven parallel layers to the Classification section. The feature extraction section of the model is then initialized with the weights of the AlexNet model trained on the ImageNet Dataset. The seven Scene Layers are initialized with the weight from the Scene Layers Pre-Training step. At last, the Fully-Connected layers weights are randomly initialized.

3.6 COHEN'S KAPPA STATISTIC

The Cohen's Kappa is a statistic useful to compare how different techniques behave while performing the same categorical classification. It is also specially useful because it takes into account the observed and expected frequencies of the images classes. Moreover, the Cohen's Kappa is ideally suited for non-ordinal classes like the ones employed in this work. Cohen's Kappa is calculated as following:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{3.1}$$

where p_o stands for the fraction of images that where correctly classified and p_e stands for the probability that a given image would be randomly classified correctly. The way it considers the probability chance of a correct classification is the reason why it is useful to evaluate the performance of classification models with unbalaced training sets.

3.7 OUR PROPOSITION

The goal of this work is to analyze the use of different deep learning architectures to implement an automatic image quality aesthetic assessment system. We studied the state-of-the-art approaches proposed by Wang *et al.* [45] and reproduced their experiments to gain insights on possible improvements. Then, we introduce a scheme inspired by the results achieved that takes into account models trained on specific image content. We analyze options on how to improve the automatic quality aesthetic assessment system. Finally we present the results obtained and present some insights gained through out the experimentation process, introducing possible future works.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: The images 3.2(a),3.2(b),3.2(d),3.2(e) and 3.2(f) are examples of images from the AVA dataset. The image belongs to the "Artificial Lightning", "Independence", "Insect", "Planes Train and Automobiles", "Play" and "Portrait of a Camera" challenges, respectively.

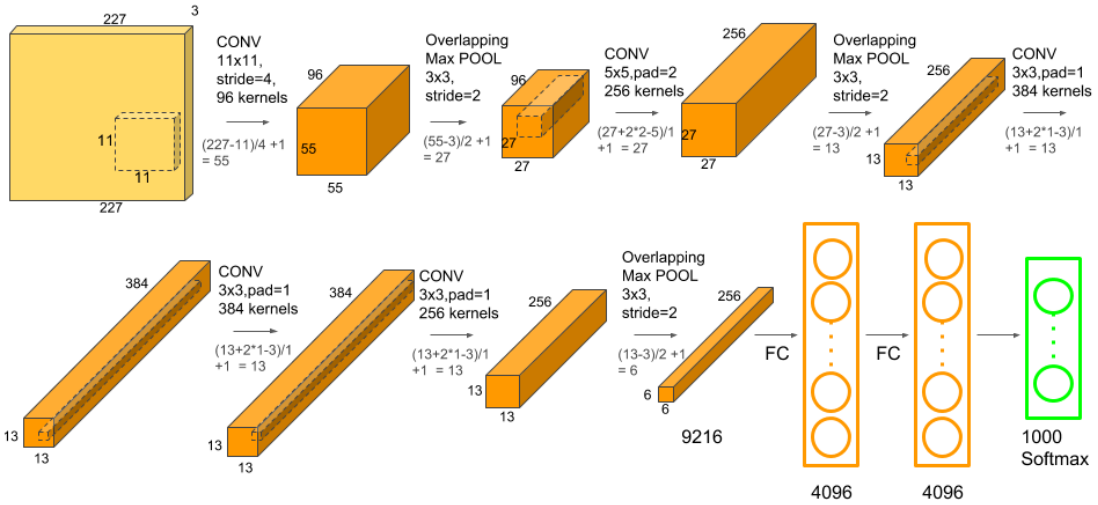


Figure 3.3: Original AlexNet Architecture [30] employed as baseline for the MSDLM architecture.

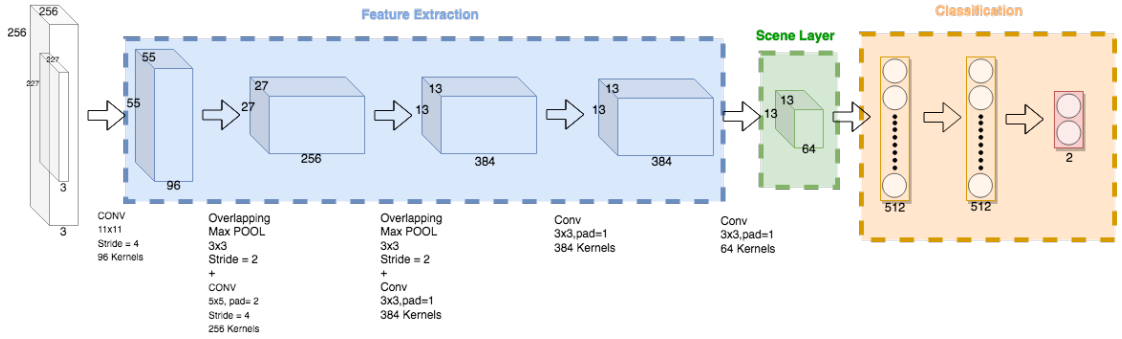


Figure 3.4: Original MSDLM pre-training setup, based on the AlexNet architecture, with the addition of a scene layer in the place of the last convolutional layer before the fully-connected layers.

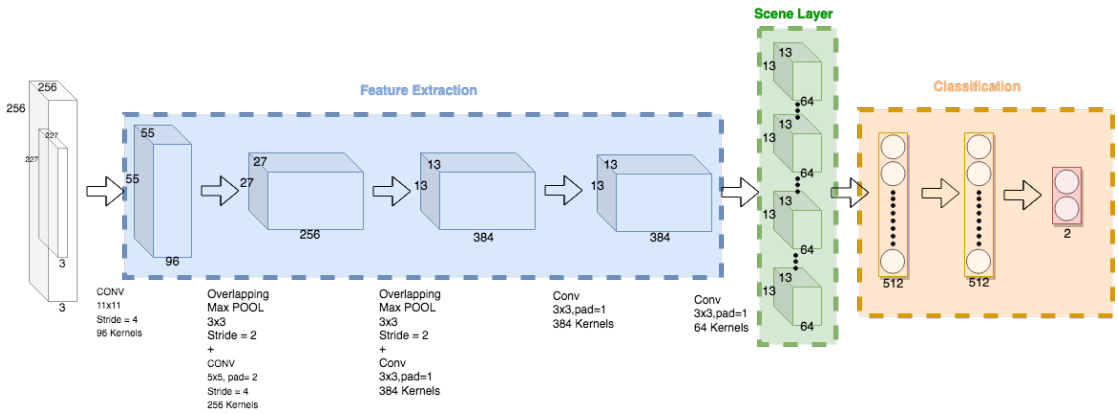


Figure 3.5: Training setup for the MSDLM, containing a Scene Layer composed of seven convolutional layers pre-trained on the their classes. After the Scene Layer, a mean-polling layer is employed to aggregate the results before the classification step.

4 PROPOSED METHODOLOGY

In this chapter, we discuss the proposed methodology, detailing the differences between the proposed method and the current state of the art methods.

4.1 MSDLM REPLICATION MAIN DIFFERENCES

While replicating the MSDLM original experiment, the original results were not achieved employing the original experiments setup. In order to approach the results the following adjustments were performed. The learning rate was chosen based on the learning rate range test presented by Leslie N. Smith [46]. For the pre-training step of the experiment, Figure 4.1 depicts the learning rate range test results presented. Based on the these results, the pre-training learning rate for the whole model was set to $1 \cdot 10^{-3}$. Moreover, a cyclical learning rate procedure [46] was used instead of the 40% decay proposed originally. The learning rate was cycled between a minimum value of $1 \cdot 10^{-3}$ and a maximum value of $1 \cdot 10^{-2}$. Finally, an early stopping procedure, with a patience set to 10 epochs, was employed to prevent overfitting and speed up the pre-training step.

To compensate for the higher number of low aesthetic value images on the CUHKPQ dataset, Wang *et al.* duplicated the high aesthetic value images spinning them 90º and 270º, using a technique called over-sampling. By doing this, both sets of images have similar sizes. In this work, the images were not duplicated, instead a sampler based on torch.utils.data.Sampler from Pytorch library was employed to weigh the images based on the number of images at each class. The sampler delivers the weights of each class to the dataloader and the dataloader uses it to assign a probability to each class that will guide how many images of each class will receive data augmentation. This approach is useful to better control the images being used during the training section of the experiment and the testing section and therefore avoiding Data Leakage between the sets.

The original experiment employs a Sigmoid function, $h_\theta(x) = \frac{1}{1+exp(x^T \cdot \theta)}$, as activation function on the last layer. Each x represents the output vector of the last layer. The output is then used to minimize the following log-likelihood function:

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} (y_i log(h_\theta(x_i)) - (1 - y_i) log(1 - h_\theta(x_i)), \tag{4.1}$$

where m stands for the total number of images in the dataset and each x_i is the output vector of the an input image i.

In this work a Log Softmax function, $h_\theta(x_i) = \log(\frac{\exp(x_i)}{\sum_j \exp(x_j)})$, is applied directly to the raw linear output ($x_i^T \theta$) of the last layer of the model. This Log Softmax Function outputs the log of a normalized probability, i.e the sum of the probability of all classes will are equal to one.

	Original Setup	Modified Setup
Scheduler	40% Decay Every 8 Epochs	Cyclical Learning Rate
Learning Rate	$1 \cdot 10^{-4}$	$1 \cdot 10^{-3} \sim 1 \cdot 10^{-2}$
Image Balacing	Replicate and rotate images	Sampler and random transformations
Early Stopping	Not applied	Applied with patience = 10

Table 4.1: Main differences between the original MSDLM setup, including the replication performed in this work.

This change, between Sigmoid and Log Softmax, was motivated by the way Cross-Entropy Loss is implemented in PyTorch. The PyTorch library optimizes its loss function by using the log of probability, this makes the computation more numerically stable.

Finally, the original article doesn't specify the learning algorithm employed to update the network weights, therefore based on the setup shown this work employs the Stochastic Gradient Descent with the Learning Rates scheme aforementioned in this section. The setup employed during the Pre-Training step was kept during the Training step. The main differences described here are summarized in Table 4.1.

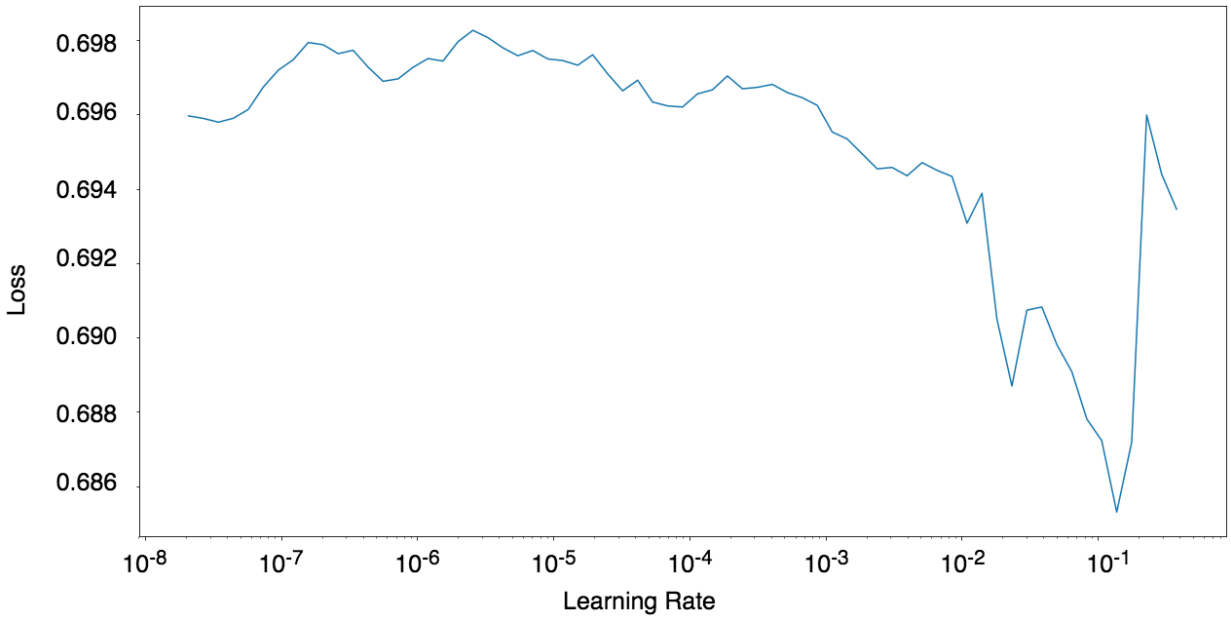


Figure 4.1: The graph presents one of the learning range tests conducted during our experiments. This method was proposed by Leslie N. Smith [46] as a way to better initialize the Learning Rate, arguably one of the most important hyper-parameters in a learning environment. Along with the Cyclical Learning Rate scheduler it allows the model to achieve better results faster.

The MSDLM Multiplexing setup was inspired by the results obtained during the replication of the pre-training step of the original MSDLM experiment. The results showed a higher degree of accuracy when the Scene Layer was being trained. This lead to the idea of using specialized aesthetic estimators based on the scene. Figure 4.2 shows the proposed pipeline to improve the results of the original experiment. First, the Scene Classifier classifies an image into one of the seven scene classes of the CUHKPQ, as previous described in Section 3.3.2. Then, based on

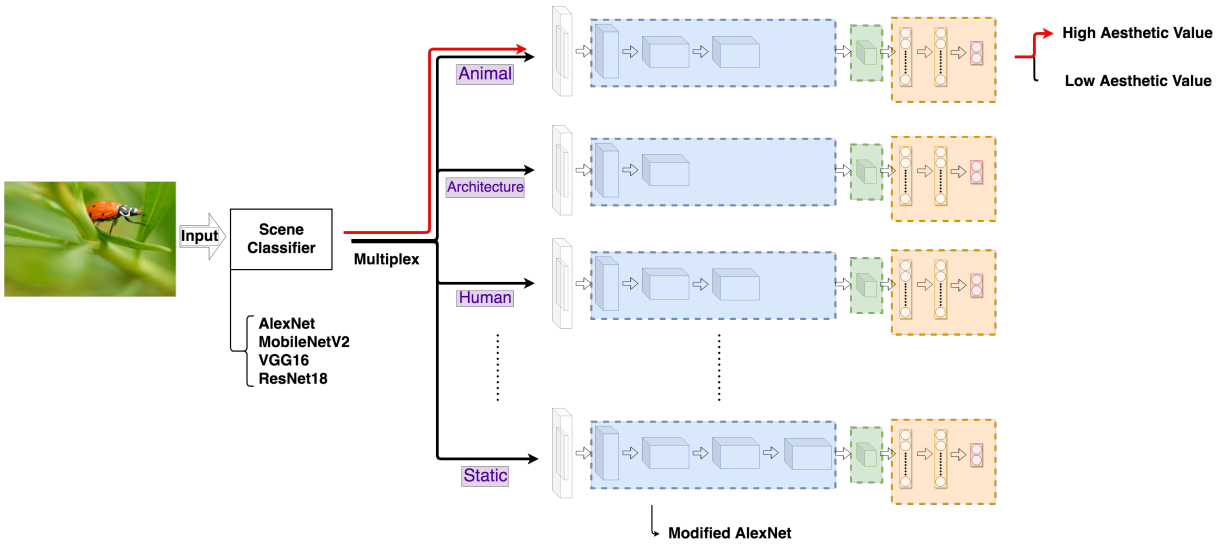


Figure 4.2: Proposed improvement of the MSDLM architecture based on a scene recognition step and a specialized aesthetic classifier.

the classification result, a specialized aesthetic estimator classifies the image into a high or low aesthetic class. As it will be seen later in Chapter 5, the results for the pre-training step show that the overall result are higher if one can take advantage of a specialized network. Moreover, in this proposed setup, the architecture of the network can be changed between scenes, giving it a better chance to detect inherent features of the image that indicate if it is aesthetically pleasing.

To take full advantage of this new setup we performed an ablation test for each scene class in the CUHKPQ dataset. In the general case, the idea behind an ablation test is to successively change aspects of the training process and through the validation analysis define how each change affects the overall result of the experiment. In the MSDLM Multiplexing setup, we chose AlexNet architecture, due to its simplicity. For each class, the network was trained, as depicted in Figure 3.4. At first, the network was setup with only one convolutional layer. At each training procedure, a new layer was added on top of the later until all twelve layers of the original AlexNet were added, including convolutional layers, polling layers, and activation layers. In order to effectively evaluate the performance of the setup, a validation step was performed. As previously explained in Chapter 2, it is possible to define when the proposed changes are starting to cause overfitting. Later, in Chapter 5 the effects of the overfitting will be presented and further discussed.

Finally, due to the detachment between the scene classification step and the aesthetic classification step, distinct architectures were trained to identify the correct image scene. In total, four models were employed for this task: the original AlexNet architecture, the MobileNet V2 architecture, the VGG16 architecture, and the ResNet18 architecture. In Chapter 5, the performance of each one of them will be presented and their results analyzed.

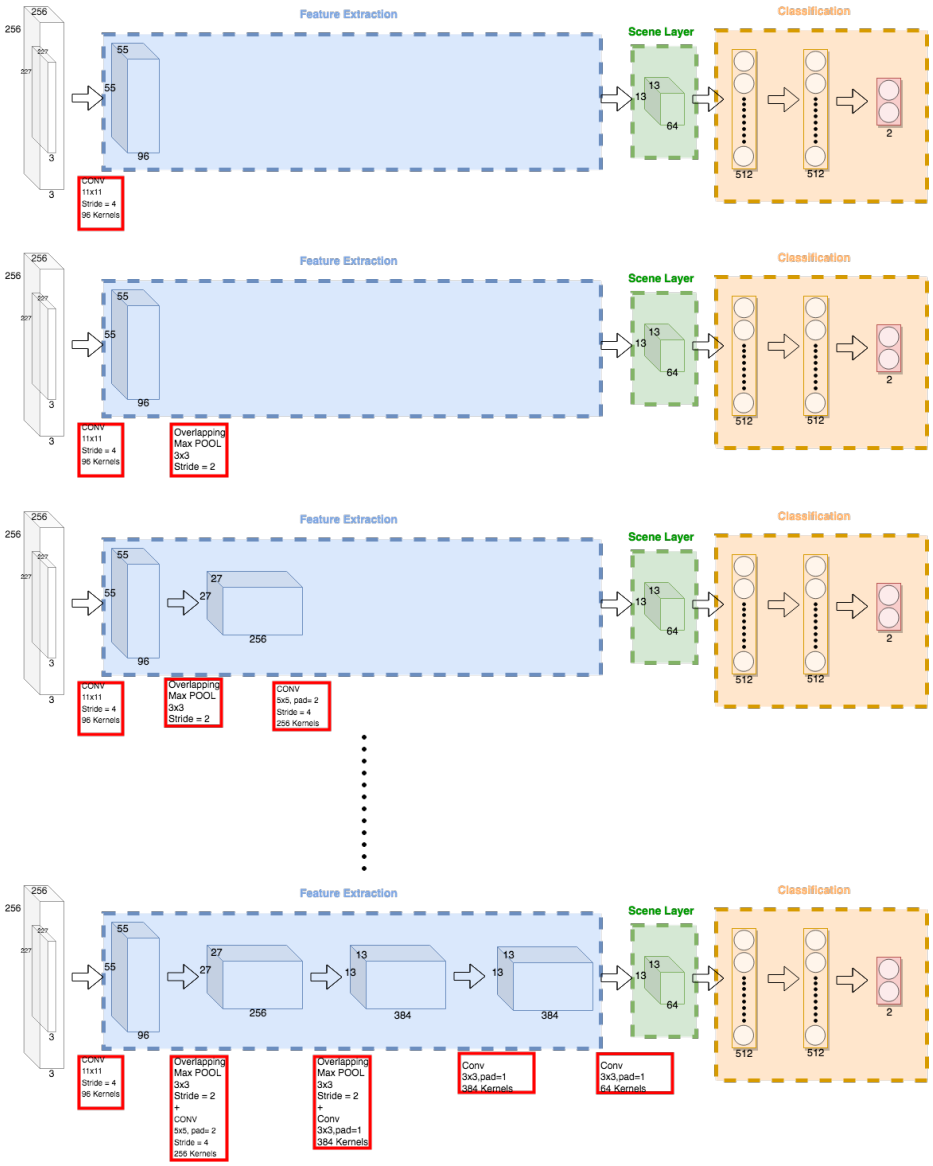


Figure 4.3: Proposed ablation test with stack layers one over the other, in order to evaluate the effect of the depth of the network on the generalization error.

4.1.1 Splitting and Sampling Technique

Table 4.2 presents the number of samples at each set for all classes. The difference between the number of images with high aesthetic and low aesthetic values is noteworthy. For the Night Scene class, there are around 3.91 times more images in the low aesthetic value group than in the high aesthetic value group. This high difference is harmful to the model training, which can become biased towards the class with larger number of samples. In order to prevent such bias, two techniques were employed sampling balancing and images transformations.

The Sampling Balancing technique works usually in two ways: under-sampling and over-sampling. As the name suggests, the under-sampling techniques consists of discarding a number of images of the dataset, chosen randomly, from the bigger class until it reaches the size of the

Aesthetic Label	Animal		Architecture		Human		Landscape		Night		Plant		Static	
	Low	High	Low	High	Low	High	Low	High	Low	High	Low	High	Low	High
Training Set	1856	764	1018	478	1992	536	1547	666	1062	271	1451	466	1597	429
Validation Set	225	94	131	65	263	61	206	84	146	37	169	73	190	41
Testing Set	211	95	141	52	215	81	197	70	148	45	183	55	218	61

Table 4.2: Number of images per class per aesthetic value. The number of low aesthetic samples in all classes are significantly higher than the high aesthetic label class. The effects of this difference on the models is further examined in Chapter 5.

smallest class. The over-sampling technique, on the other hand, takes the opposite direction. It multiplies the number of samples in the smaller class until it reaches the size of the bigger class. Both approaches are reasonable, but can yield serious problems. In the under-sampling approach, the dataset is not being fully used. This is a problem when a dataset is already small, causing a serious problem in the learning capacity of the model during training. In the over-sampling approach the repetition of the same image many times might create a bias in the model because the same sample is present multiple times.

As a compromise between the two approaches, a solution employing image transformations is employed. During the training and validation stages for all experiments on CUHKPQ, a sampler based on the `torch.utils.data.sampler.Sampler` class from the PyTorch package was adopted. PyTorch Library was first released on 2016. It is an open-source machine learning library focused on ease development of numerical algorithms, easily extendable packages and easy debugging capabilities. The PyTorch library was developed as a transcript of the Torch library that was originally written in Lua. PyTorch has two major features that renders it very useful in research. First, it has a tensor computing capability that easily lets a researcher take advantage of graphical processing units that accelerate the development of new models. Moreover, due to the Python focused nature of its development, Pytorch is able to work seamlessly with other Python packages, such as Numpy and Scikit-Learn. The second major feature that lends PyTorch library as a great option is its implementation of autodiff systems that allows the researcher to build very complex networks without worrying about the difficulty of develop the differentiation algorithms. This means that the backpropagation that generates the derivatives necessary to perform gradient descent and update the weight are performed automatically.

The `torch.utils.data.sampler.Sampler` works along a `Dataloader` that is responsible to load the samples to the model during training and validation. The `Dataloader` keeps track of the images that were already loaded during an epoch and, along with the sampler, it regulates the number of samples of each class. The idea is to apply image transformations to the under-sampled classes to create "new" samples. This way it is possible to use the full training set during training without throwing away any data. Moreover, this approach brings a two-fold benefit, besides the possibility to train the model with the complete dataset, it "increases" the size of the dataset by changing the image enough to seem like a new image to the model.

Finally, the images transformations applied were distinct between the different experiments.

For the case of the scene classification in the MSDLM Multiplexing, random resized crops, random flips, and random affine transformations were applied. For the aesthetic classification experiments in both the original MSDLM setup and the Multiplexing setup, the affine transformation was excluded. The reason for this is that the affine transformation alters the composition of the image and this is a key aspect to the aesthetic evaluation of an image.

5 RESULTS

In this chapter, the results for the experiments previously described are introduced and further analyzed, along with a more in-depth analysis of how we might improve the overall performance of the models. First, we present the best results achieved by the baseline algorithm, simply by employing the original AlexNet architecture. As depicted in Figure 3.3, the last layer is replaced by a two-neurons output to predict the aesthetic label. The goal for this first experiment is to test the performance of a simple architecture. This allows to better grasp the improvements of the proposed scheme versus Wang *et al*'s workand in our proposed scheme. Second, we analyze the best results achieved in our attempt to reproduce the original MSDLM experiment, whose modifications were highlighted in Section 4.1. Finally, we analyze the results obtained by the proposed scheme that employs a multiplexing approach. We will further discuss how we can improve the results and how we might achieve a higher performance.

5.1 BASELINE EXPERIMENT

The results achieved by the baseline algorithm are presented in Table 5.1. This table is divided into four parts, where we summarize the accuracy results for the best models for each experiment. In the first part we show the baseline experiment results, while in the second part the original MSDLM reproduction results. In the third part of the table, we show the results obtained with the proposed scheme and, for comparison purposes, in the last part of the table we present the results reported by Wang *et. al.* in their original work [45].

Results in the first part of Table 5.1 shows us that during validation we achieved a good result. But, during the test, the baseline model performed poorly. In fact, Figure 5.1 shows that the validation loss has its lowest value for the first epoch and, after this, it starts to increase. This is a clear sign that the model overfit the training set. Moreover, the Cohen's Kappa score obtained for the test was 0.0064, proving further evidence that the baseline model performed poorly. We believe that one of reasons for these poor results is that the number of samples available for training in the CUHKPQ dataset is low. Therefore, with this low number of samples, even an architecture as "simple" as AlexNet may overfit. We considered these results as the lowest scores and moved on to test the original MSDLM architecture to check if their model might performed as reported.

Table 5.1: Summary of the best accuracy achieved in the experiments. For the training stages, the validation results are presented. For the testing stages, the testing results are presented.

Scene	Animal	Architecture	Human	Landscape	Night	Plant	Static	Overall
Baseline Experiment								
Training:								
Best Validation Results								
AlexNet	———	———	———	———	———	———	———	89.69%
Testing:								
AlexNet	———	———	———	———	———	———	———	74.210%
Reproducing MSDLM Experiment								
Pre-Training:								
Best Validation Results								
MSDLM Cyclical LR	94.932%	93.96%	97.577%	95.56%	92.885%	92.219%	94.627%	———
Training:								
Best Validation Results								
MSDLM Cyclical LR	93.417%	88.776%	96.914%	93.103%	90.164%	90.496%	92.208%	91.092%
Testing:								
Results	———	———	———	———	———	———	———	89.220%
Multiplex MSDLM Experiment								
Pre-training								
Ablation Validation Results:	90.777%(7)	95.968%(9)	95.028%(7)	92.000%(7)	95.789%(7)	94.444%(8)	93.056%(4)	———
Testing Results								
Multiplex Pipeline	———	———	———	———	———	———	———	85.100%
MSDLM Reported Results[45]:								
Testing Results								
MSDLM Original	92.11%	91.50%	94.92%	91.77%	91.69%	91.92%	92.78%	92.59%

Testing Loss Results	
Baseline experiment	1.7293
Reproducing MSDLM Experiment	0.3036
Multiplex MSDLM Experiment	0.5427

Table 5.2: Summary of the Cross Entropy loss results during the testing step for the best performing model at each experiment. The original Wang *et al.* setup performed better than the Multiplex MSDLM.

Testing Cohen's Kappa Results	
Baseline experiment	0.64%
Reproducing MSDLM Experiment	73.05%
Multiplex MSDLM Experiment	47.96%

Table 5.3: Summary of the Cohen's Kappa results during the testing step for the best performing model at each experiment. The original Wang *et al.* setup performed better than the Multiplex MSDLM.

5.2 REPRODUCING WANG'S MSDLM EXPERIMENT

In this section, we analyze the results from our attempt to reproduce the original MSDLM experiment by Wang *et al.*. First, we analyze the performance of the pre-training step, then we analyze the training results, and finally the testing results. Figures 5.2 (a)-(g) show the results obtained during our attempt to reproduce the original MSDLM pre-training step.

The results in the original pre-training step shows us that, unlike the baseline, the pre-training models were able to train for a longer period of time before overfitting. Moreover, it is important to highlight that none of the models pre-trained for more than 20 epochs, meaning that due to the early-stopping setup they achieved their minimum loss value at most at the 10th epoch. This setup was applied to prevent overfitting and to allow us to experiment with different hyperparameters.

Some key differences might explain why the pre-training setup performed better than the baseline setup. The most important difference is that the baseline model was trying to learn a generic aesthetic classification, that is, it was trying to fit the results for all seven scenes, while the pre-training models were each trained only a single class at a time. This might indicate that it was easier for the model to find a beauty pattern when it was analyzing the same content. This observations motivated our approach to develop a system that would take further advantage of this specialized models, as explained in Section 4.1.

Finally, the number of image samples at each class does not seem to affect the pre-training step that much. The lowest loss among the pre-training models was achieved for the Human content scenes, where the training set has 2,528 image samples. While the loss for the model pre-trained on the Animal content scenes was almost double the loss, considering a training set of 2,620 image samples.

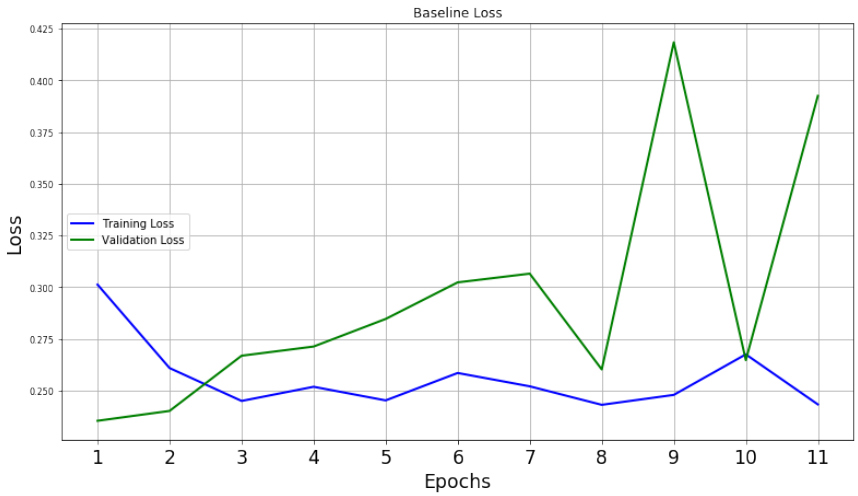


Figure 5.1: Training results for the best baseline model, showing that the model was overfitting the training set.

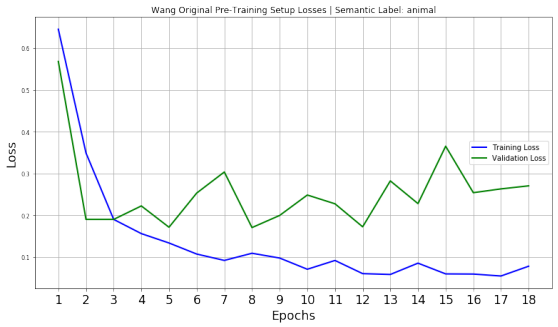
5.3 ORIGINAL TRAINING

As introduced earlier and illustrated in Figure 3.5, the MSDLM training pipeline stacks the scene layers pre-trained on each specific class of the CUHKPQ dataset. After achieving the results presented in the previous section, the weights of the scene layer were loaded into the model in Figure 3.5. The graphs in Figures 5.3, 5.4 and 5.4 show the overall results corresponding to loss, accuracy, and Cohen's Kappa, respectively, for the training and validation steps of the original MSDLM Pipeline.

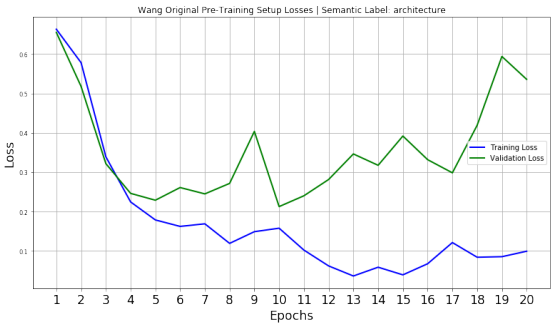
The loss results during the training step shows that best results were achieved fairly fast at the 6-th epoch. The blue graph in Figure 5.3 shows that the Training Cross Entropy Loss was decreasing during the whole experiment up to the 12th epoch but after the 6th epoch, the validation Cross Entropy Loss raises steadily, clearly showing that the model was overfitting the training set at this point. In comparison to the results of the pre-training step models in Figures 5.2 (a)-(g), the training step model achieved a higher validation loss (0.226) than the worst performing model (see Figure 5.2(b)) on the pre-training step (0.213).

5.4 MULTIPLEX MSDLM EXPERIMENT

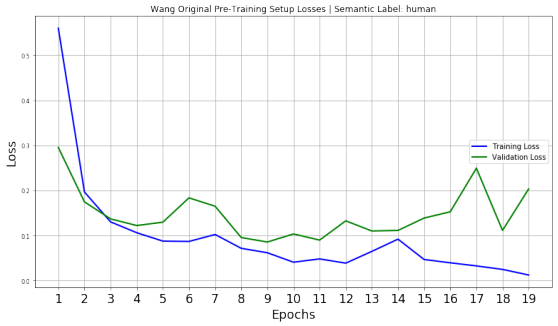
In this section, we analyze the performance of our proposed Multiplex MSDLM scheme. Due to the essence of our solution, as explained in Chapter 5, this section will be split into two. The first part describes the scene classification Algorithm, while the second describes the aesthetic classification algorithm.



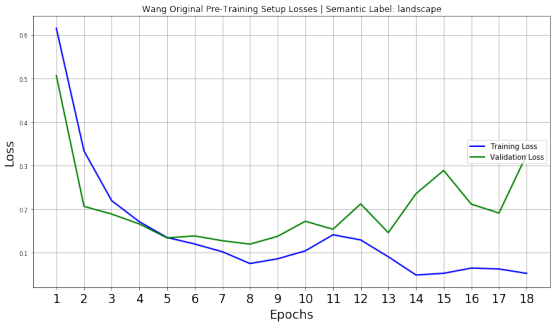
(a) Animal | Validation loss: 0.17 | Epoch 8



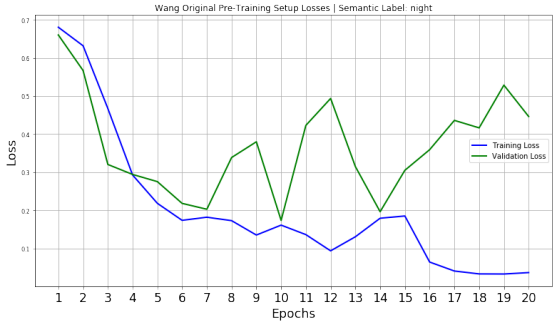
(b) Architecture | Validation loss: 0.213 | Epoch 10



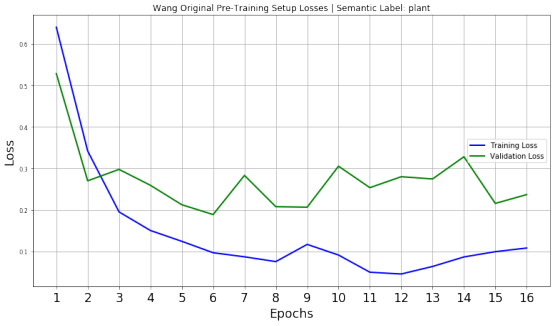
(c) Human | Validation loss: 0.086 | Epoch 9



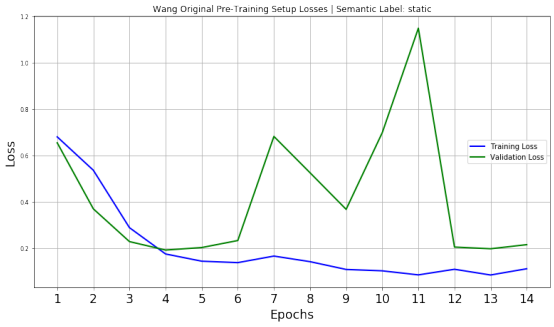
(d) Landscape | Validation loss: 0.12 | Epoch 8



(e) Night | Validation loss: 0.173 | Epoch 10



(f) Plant | Validation loss: 0.189 | Epoch 6



(g) Static | Validation loss: 0.192 | Epoch 4

Figure 5.2: Pre-Training results for the MSDLM Experiment. The losses show how each model learned each class. The difference between the loss behavior highlights how different image's content affects the learning process.

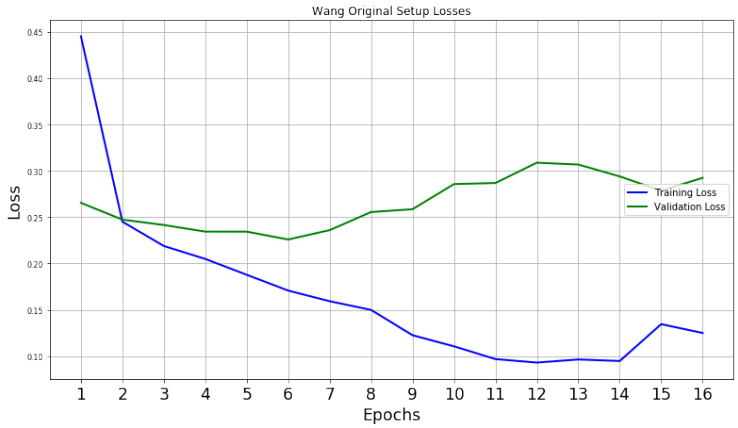


Figure 5.3: The graph represents the validation loss (Cross Entropy Loss) during the training step for the original MSDLM Pipeline setup. The best value was **0.226** at epoch 6

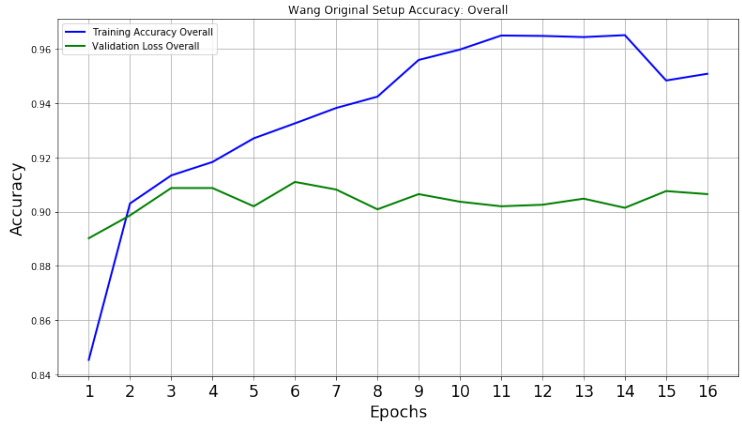


Figure 5.4: Overall accuracy score during the training step for the original MSDLM Pipeline setup. The best value was 91.092% at the 6th epoch.

5.4.1 Scene Classification

Due to the detached essence of the Multiplex MSDLM, it was possible to test different models to perform the scene classification task. Table 5.4 presents the best results for each of the different tested models. The results in this table show some interesting numbers. First, although the lower validation loss was achieved by the VGG16 architecture, the best accuracy result was achieved by the MobileNetV2 architecture. Second, the models VGG16 and ResNet18 achieved the best accuracy and the best loss at different epochs. This phenomena highlights a conclusion that might seem counter-intuitive. A decrease in loss, in this case of the Cross-Entropy Loss, is not always followed by an increase in the accuracy.

This mismatch between loss and accuracy happens because the classification model outputs the probability of an image belonging to one of the scene classes. The cross-entropy loss in this case takes the value of the log of the probability and calculates the error value. As the model learns, the cross-entropy loss value decreases but the predicted probability may not cross the classification threshold that divides the two classes, therefore the accuracy value is not updated

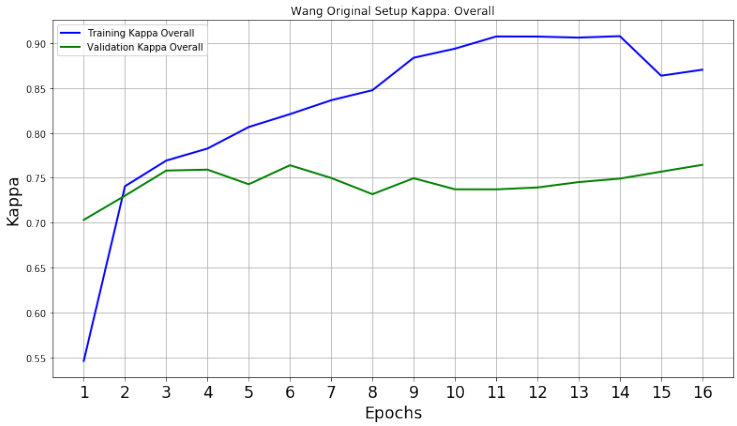


Figure 5.5: Overall Kappa score during the training step, for the original MSDLM Pipeline setup. The best value was 76.443%, which was achieved at the 16-th epoch.

Model	Loss	Accuracy	Kappa	Best Loss Epoch	Best Accuracy Kappa Epoch
AlexNet	0.707	77.08%	73.07%	34	34
MobileNetV2	0.598	**81.354%**	78.076%	16	16
VGG16	**0.595**	81.176%	77.859%	19	27
ResNet18	0.61	80.84%	77.458%	41	29

Table 5.4: Results for the scene classification stage.

even with a lower loss value. Tables 5.5 (a)-(b), show an example of the mismatch between Loss and Accuracy. In Table 5.5 (a) we observe that the model is improving, but it did not correctly classify Image 3 (because of a difference of 1%). In Table 5.5 (b), a higher loss model was able to correctly predict all three images.

In the end, the value that guided our choice was the Cross-Entropy Loss, which was used in all other experiments. But, due to the proximity between the results from the MobileNetV2 and the VGG16, we opted to use the MobileNetV2. This was motivated by the idea that a simpler model might generalize better than a more complex one. Since the MobileNetV2 has around 60 times less trainable parameters than the VGG16, this seemed like a fair choice given our computational set up. Table 5.6 summarizes the number of parameters and the time it took to train the best results, respectively.

5.4.2 Aesthetic Classification

As mentioned in Section 2.5, for each scene on the CUHKPQ dataset, a specialized model was trained to classify the aesthetic score of the image. For each scene, an AlexNet-type architecture was employed. After training each section, a new layer was added to the architecture and a new training was performed. The process was repeated until all twelve original layers of the AlexNet

	Ground truth			Prediction				
	Animal	Architecture	Human	$p(Animal)$	$p(Architecture)$	$p(Human)$	Cross-Entropy	Accuracy
Image 1	1	0	0	**0.80**	0.15	0.05	0.146	1
Image 2	1	0	0	**0.80**	0.10	0.10	0.145	1
Image 3	1	0	0	0.33	0.33	**0.34**	0.642	0

Loss = 0.311

Accuracy = 66.67%

(a)

	Ground truth			Prediction				
	Animal	Architecture	Human	$p(Animal)$	$p(Architecture)$	$p(Human)$	Cross-Entropy	Accuracy
Image 1	1	0	0	**0.50**	0.20	0.30	0.424	1
Image 2	1	0	0	**0.50**	0.25	0.25	0.423	1
Image 3	1	0	0	**0.40**	0.38	0.22	0.547	1

Loss = 0.465

Accuracy = 100%

(b)

Table 5.5: Table (a) shows an example where the model was able to achieve a lower loss than the model on table (b), but the accuracy on model (a) is lower than the model (b).

Model	Trainable Parameters	Duration of Training
AlexNet	57.032.519	175m 7s
MobileNetV2	**2.232.839**	**114m 52s**
VGG16	134.289.223	170m 2s
ResNet18	11.180.103	215m 56s

Table 5.6: The second column displays the number of trainable parameters on each model in the first column while the third column displays the amount of time it took to train each model. As expected the number of parameters on the MobileNetV2 is considerably lower than the other. From a VC dimension point-of-view the MobileNetV2 is simpler than the others and most likely to generalize better.

architecture were added. Figure 4.3 illustrates this processes. At the end, 13 architectures had to be trained for each scene, with each architecture being fine-tuned to a different parameter.

The graph in Figure 5.6 shows the results for one of the many different rounds of training. This graph captures the standard behavior for the ablation test. As one can see in Figure 5.6(a), as the number of layers increases (represented by the lines going from dotted to filled), the loss on the training set decreases more and more, down to the point where it overfits the training dataset. This overfitting behavior is highlighted in Figure 5.6(b), where the two models with more layers are performing worse than all others. This kind of behavior is expected, as mentioned in Chapter 2 an increase on the number of layers increases the VC dimension of the model. Therefore, the model will eventually overfit the data, given the low number of images available in the dataset. In the case of the scene Animal, there were only 2,620 images available for training.

Finally, after analyzing the ablation results we were able to choose the ideal number of layers for each class based on the validation loss. The third part of Table 5.1 presents the accuracy results for the models we choose based on the ablation test and, between parentheses next to each accuracy value, stands the number of layers that achieved the lowest validation loss. The results show that the best performing models were using around half of the total available layers on the AlexNet model.

5.5 GENERAL ANALYSIS OF THE MSDLM RESULTS

The results regarding the original MSDLM setup during pre-training and training confirmed to us some previous ideas. Based on the loss results we could observe that the models learning specific scenes achieved better results than the model that was trying to learn a generic classifier.

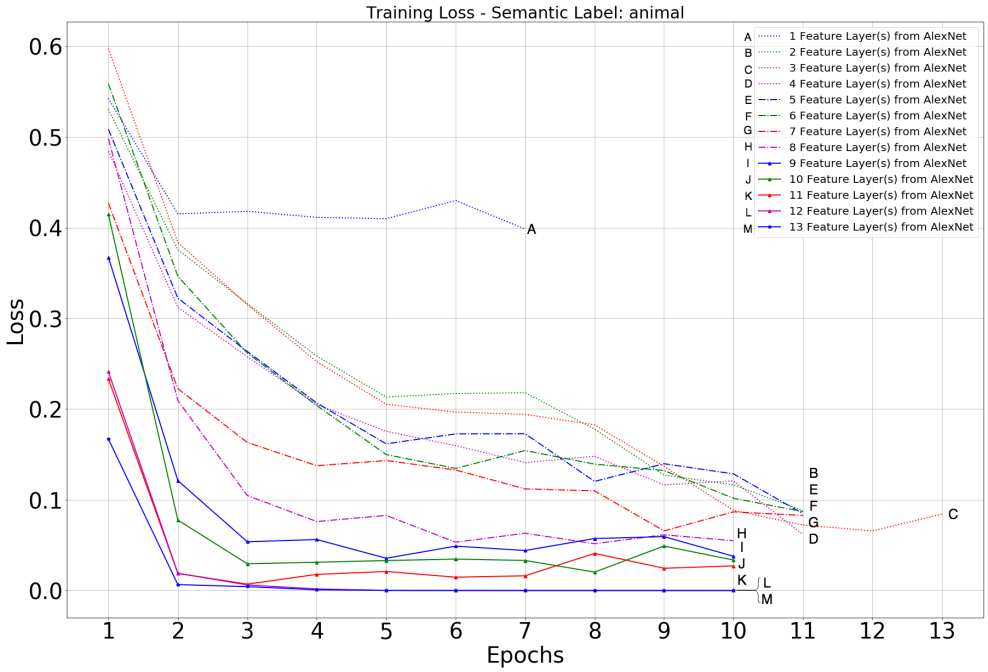
During the ablation test it was observed that distinct classes of the CUHKPQ achieved their best results using distinct number of layers in the feature extraction section. This make us believe that some image contents, like the Static class, are better perceived by lower level features than other. In order to gain further insight on this, the analyses of the last layers might give more information on the most important features learned by each model and ideas to improve upon.

The increase of the patience parameter did not yield any significant improvement to the results of the experiment. Therefore, this value was kept at a minimum of 10 epochs to prevent the experiment of being interrupted earlier than necessary. This decision was based on the validation results, which showed it would have been a mistake to use less than 10 epochs, due to the initial variations of the results. It is worth pointing out that larger patience values would take more time and resources during training. But, using 10 epochs proved to be a fair choice.

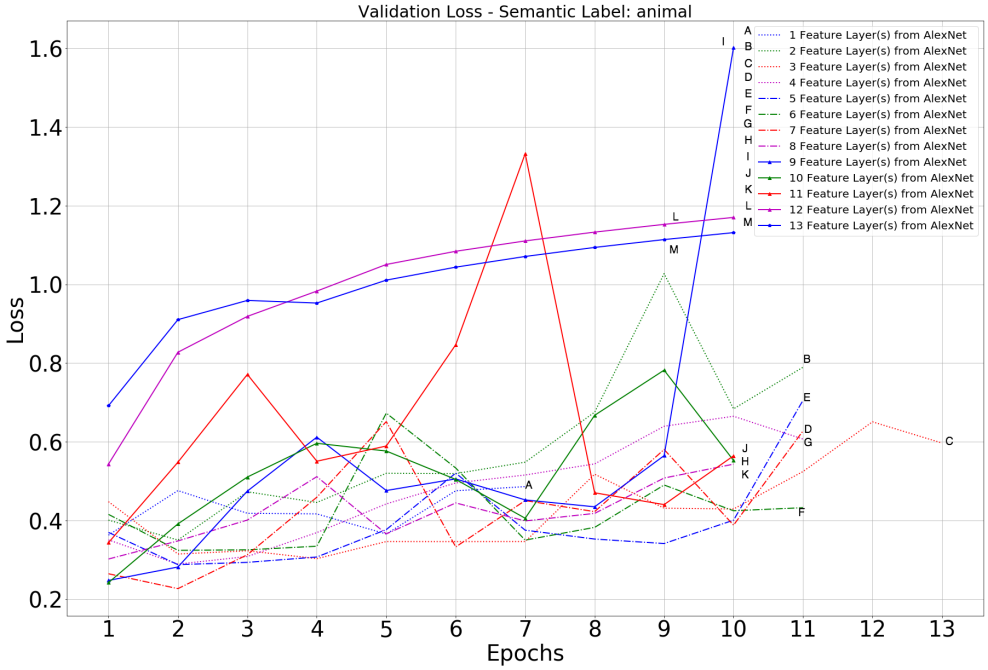
Among our experiments, the aesthetic classifiers fitted the sample images faster than the scene classifiers. It seems to us that the number of training samples could be one of the key factors for this behavior. This is based on the observations that during the scene classification experiments, which employs the "same" dataset with different targets, the models trained for a longer period before start overfitting. The major difference between the two setups data-wise was the number of images per class, where in the scene classification the low aesthetic value images and the high aesthetic value images were aggregated to perform the training. In order to evaluate this hypothesis it might be useful to train the same scheme on larger datasets.

The original Wang *et al.* paper [45] gives us some clues on how to proceed further. The authors tested their model on the AVA dataset and found that their accuracy score was lower. The reason might be the way the AVA dataset is built, which makes it much more susceptible to a higher variance error degree and, consequently, a higher error. Their results makes us believe that future aesthetic models will benefit greatly from a dataset with a more rigorous sample control, such as CUHKPQ, along with a higher number of samples.

Tables 5.1, 5.2, and 5.3 summarize all test results achieved by the best models in each exper-



(a) Training Results for the Ablation. Test



(b) Validation Results for the Ablation. Test

Figure 5.6: The ablation test was performed adding each layer of the AlexNet Architecture successively. The results presented are from the training process on the Animal scene layer, but the results are similar for the others scene. The graphs in figure (a) shows the training results for the ablation test for the Animal scene, as the layers are added the model fits more and more the training set. The graphs in figure (b) shows the validation results for the ablation test for the Animal scene. It is possible to see that with each new layer the model starts to overfit the training set and its loss increases during validation.

iment. The baseline model achieved very poor results, with a lower accuracy and a higher loss compared to the other two setups. Moreover, our approach of taking advantage of the specialized models based on the content class of each image performed better than the baseline model. But, it under-performed when compared to the original MSDLM setup.

Finally, the Cohen's Kappa results shown in Table 5.3 depict how poorly the Baseline performed. This behavior was expected because the testing set is not as augmented and balanced as the training set and the validation set. This means that the test set reflects better how the original distribution of images behave, with a bias towards low aesthetic images. The baseline model is most likely biased towards this value and, therefore, it has not actually learned how to evaluate image aesthetics. This result also highlights the importance of employing a metric like Cohen's Kappa. Unlike the accuracy, which only measures how many samples were right, Cohen's Kappa weighs the amount of samples at each class and captures the imbalance bias in the dataset.

6 CONCLUSION

This work aimed to evaluate different techniques to assess the aesthetic quality of images. It analyzed state of the art schemes and proposed a new method that could improve the aesthetic prediction results by leveraging on the use of a content-based approach. The results of our experiments shed light on the process and problems of an aesthetic quality assessment system.

During our experiments we evaluated the effectiveness of employing deep learning methods to extract features that could improve the overall prediction of aesthetic quality. We believe that our results prove that the use of deep learning methods are a great approach to the task and worth of further exploration. Moreover, based on our proposed method we could observe how the content affected the training process and how distinct convolutional networks architectures are more suited than others when assessing the quality of distinct contents.

We proved that images with distinct contents affect the outcome of an aesthetic classifier system by analyzing the testing results and observing how the number of layers influenced the final result. Moreover, the ablation test showed us that distinct images contents have different levels of feature complexities taken into account, this was concluded based on the idea that more layer generate more complex features.

6.1 FINAL THOUGHTS AND FUTURE WORKS

As Datta *et al.* points out in their work [23], the use of complex models such as CNNs turns the interpretation of their predictions harder than for other simpler models because the features that are learned across the different layers are not easily understandable. But, through our multiplexing approach, we gained insights that could guide researchers using deep-learning methods. We observed that the content affects learning and that certain contents can be evaluated using mostly lower level features. To gain more insight on what exactly features are influencing the model's decision it might be useful to perform a visual analysis of the CNNs.

During our experiments we were able to observe that the deep learning models were able to learn how to identify the aesthetic value of an image. Also, the use of deep learning feature extractors outperformed the use of handcrafted extractors. Some previous works took a step further and have employed both types of extractors to achieve promising results. By first using some well known feature extractors, such as the local binary patterns (LBP), and passing the results as input to CNNs, researchers were able to achieve interesting results. This might be a valid approach that is able to shed some light on what the networks are learning, giving that, regarding the features, CNNs are mostly black boxes .

In future works, we intend to explore other arrangements inspired by an ensemble of methods,

where an image will be evaluated by all the trained models. Also, we plan to analyze if a weighted voting scheme might yield better results and insights. Moreover, we intend to expand the ablation test scope to other types of architectures, such as ResNet and Inception. Such architectures are susceptible to ablation tests because they can only be analyzed by stacking their fundamental blocks, instead of stacking each layers separately.

Finally, in order to better analyze the most relevant aspects of an image, we plan to implement a Class Activation Maps (CAM) technique. This technique highlights the most important areas of an image and indicate which areas were most relevant to the prediction. We believe that such visualizing techniques can help understand the performance of the CNNs, making clear which parts of the images are taken into consideration when taking a decision.

Bibliography

[1] "Understanding basic aesthetics in photography," Aug 2016. [Online]. Available: https://petapixel.com/2016/08/08/understanding-basic-aesthetics-photography/

[2] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook New York, NY, USA:, 2012, vol. 4.

[3] M. Freeman, *The complete guide to light & lighting in digital photography*. Sterling Publishing Company, Inc., 2007.

[4] J. Itten, *Design and form: The basic course at the Bauhaus and later*. John Wiley & Sons, 1975.

[5] A. Chatterjee and O. Vartanian, "Neuroscience of aesthetics," *Annals of the New York Academy of Sciences*, vol. 1369, no. 1, pp. 172–194, 2016.

[6] G. T. Fechner, *Vorschule der aesthetik*. Breitkopf & Härtel, 1876, vol. 1.

[7] S. Zeki, "Clive bell's "significant form" and the neurobiology of aesthetics," *Frontiers in human neuroscience*, vol. 7, p. 730, 2013.

[8] T. Ishizu and S. Zeki, "The brain's specialized systems for aesthetic and perceptual judgment," *European Journal of Neuroscience*, vol. 37, no. 9, pp. 1413–1420, 2013.

[9] S. Brown, X. Gao, L. Tisdelle, S. B. Eickhoff, and M. Liotti, "Naturalizing aesthetics: brain areas for aesthetic appraisal across sensory modalities," *Neuroimage*, vol. 58, no. 1, pp. 250–258, 2011.

[10] L. F. Barrett, B. Mesquita, K. N. Ochsner, and J. J. Gross, "The experience of emotion," *Annu. Rev. Psychol.*, vol. 58, pp. 373–403, 2007.

[11] L. F. Barrett and T. D. Wager, "The structure of emotion: Evidence from neuroimaging studies," *Current Directions in Psychological Science*, vol. 15, no. 2, pp. 79–83, 2006.

[12] S. Kühn and J. Gallinat, "The neural correlates of subjective pleasantness," *Neuroimage*, vol. 61, no. 1, pp. 289–294, 2012.

[13] H. Leder, B. Belke, A. Oeberst, and D. Augustin, "A model of aesthetic appreciation and aesthetic judgments," *British journal of psychology*, vol. 95, no. 4, pp. 489–508, 2004.

[14] B. Wandell, S. Dumoulin, and A. Brewer, "Visual cortex in humans," *Encyclopedia of neuroscience*, vol. 10, pp. 251–257, 2009.

[15] M. W. Greenlee and U. T. Peter, "Functional neuroanatomy of the human visual system: A review of functional mri studies," in *Pediatric Ophthalmology, Neuro-Ophthalmology, Genetics*. Springer, 2008, pp. 119–138.

[16] P. Cavanagh, "The artist as neuroscientist," *Nature*, vol. 434, no. 7031, p. 301, 2005.

[17] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2408–2415.

[18] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti *et al.*, "Image database tid2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, vol. 30, pp. 57–77, 2015.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[20] "Portrait of Doña Antonia Zárate by Francisco Goya." [Online]. Available: https://joyofmuseums.com/museums/russian-federation/saint-petersburg/hermitage-museum/portrait-of-dona-antonia-zarate-hermitage-museum/

[21] "Photo.net Where Photographers Inspire Each Other." [Online]. Available: http://photo.net/

[22] "500px." [Online]. Available: https://500px.com/

[23] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Studying aesthetics in photographic images using a computational approach," in *European Conference on Computer Vision*. Springer, 2006, pp. 288–301.

[24] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[25] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[26] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[29] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 457–466.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

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

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[34] Y. Deng, C. C. Loy, and X. Tang, "Image aesthetic assessment: An experimental survey," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 80–106, 2017.

[35] Y. Ke, X. Tang, and F. Jing, "The design of high-level features for photo quality assessment," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 419–426.

[36] X. Sun, H. Yao, R. Ji, and S. Liu, "Photo assessment based on computational visual attention model," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 541–544.

[37] X. Lu, Z. Lin, X. Shen, R. Mech, and J. Z. Wang, "Deep multi-patch aggregation network for image style, aesthetics, and quality estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 990–998.

[38] S. Kong, X. Shen, Z. Lin, R. Mech, and C. Fowlkes, "Photo aesthetics ranking network with attributes and content adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 662–679.

[39] L. Mai, H. Jin, and F. Liu, "Composition-preserving deep photo aesthetics assessment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 497–506.

[40] A. Ng, "Machine learning yearning," *URL: http://www.mlyearning.org/(96)*, 2017.

[41] "A digital photography contest." [Online]. Available: http://www.dpchallenge.com/

[42] W. Luo, X. Wang, and X. Tang, "Content-based photo quality assessment," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2206–2213.

[43] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[44] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[45] W. Wang, M. Zhao, L. Wang, J. Huang, C. Cai, and X. Xu, "A multi-scene deep learning model for image aesthetic evaluation," *Signal Processing: Image Communication*, vol. 47, pp. 511–518, 2016.

[46] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.