UNIVERSIDADE D
COIMBRA

Leonardo Machado Alves Vieira

DEVELOPMENT OF AN AUTOMATIC IMAGE ENHANCEMENT FRAMEWORK

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems, advised by Professor João Nuno Gonçalves Costa Cavaleiro Correia and Professor Fernando Jorge Penousal Martins Machado and presented to Faculty of Sciences and Technology / Department of Informatics Engineering.

June 2020

This page is intentionally left blank.

Faculty of Sciences and Technology

Department of Informatics Engineering

Development of an Automatic Image Enhancement Framework

Leonardo Machado Alves Vieira

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems advised by Prof. João Nuno Gonçalves Costa Cavaleiro Correia and Prof. Fernando Jorge Penousal Martins Machado and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

June 2020

1 2 9 0

UNIVERSIDADE D
COIMBRA

This page is intentionally left blank.

Agradecimentos

Aprendi com esta dissertação muito mais do que esperava, e isso não teria sido possível sem todos aqueles que me ofereceram ajuda e motivação durante o seu decorrer, e a quem agora dedico este agradecimento.

Primeiramente, agradeço aos meus professores e orientadores João Nuno Correia e Penousal Machado, por me terem ensinado, acompanhado e guiado durante este ano, sempre e em tudo o que precisei.

Obrigado ao Tiago Gomes, Pedro Carvalho e Guilherme Silva por terem partilhado estes dois anos de mestrado comigo. Por todas as horas que passámos em momentos de trabalho, diversão, partilha e por todas as vezes que ofereceram risos e gargalhadas que me ajudaram a superar momentos em que me sentia em baixo. São verdadeiros amigos dos quais me orgulho imenso e que nunca esquecerei. Fico feliz que os nossos caminhos se tenham cruzado.

À Sara Silva, minha melhor amiga e companheira fiel durante os últimos três anos e meio. Obrigado por me fazeres ver o lado bom quando a minha motivação estava em baixo. Obrigado por estares sempre ao meu lado, incansável, a torcer por mim. Sempre com suporte, ajuda e abraços prontos a qualquer altura. Obrigado por me ouvires e responderes quando precisava de desabafar ou debater ideias. Por deixares um espaço para mim em todos os teus dias, por muito atarefados e trabalhosos que eles se mostrassem. Sem a tua presença, teria sido tudo tão mais difícil. Obrigado.

E finalmente, obrigado à minha família, com um muito especial, quente e forte abraço aos meus pais. Obrigado aos dois por serem os meus inabaláveis pilares durante toda a minha vida. Por me educarem, tomarem conta de mim e por me oferecerem oportunidades que não tenho como pagar. Por me aconselharem quando temiam que eu não tomasse a melhor decisão e por estarem incondicionalmente prontos para me amparar e acolher. O maior e mais forte obrigado.

A todas as pessoas e vivências que me fizeram crescer e aprender. Obrigado por me tornarem na pessoa que sou hoje.

Leonardo Machado Alves Vieira

This page is intentionally left blank.

Abstract

Image enhancement is an image processing procedure in which an image becomes better suited for a task, and so, it is very relevant across multiple fields, such as medical imagery, space imagery, bio-metrics, etc. Image enhancement can be used to alter an image in several different ways, for instance, by highlighting a specific feature in order to ease post-processing analyses by a human or a machine, or by increasing its human perceived aesthetic.

The main objective of this work is the development of a possible automatic image enhancement system, while having digital real-estate marketing as a case study, in the context of the project *"Indest - Indicador de composicion estética"*. We explored existing research in image enhancement and propose an end-to-end image enhancement pipeline architecture that takes advantage of both classical, evolutionary and machine learning approaches from the literature.

The framework is very modular as it can allow changes in its components and parameters. We tested it using a provided dataset of various real-estate pictures of different quality. The outputted enhanced images were evaluated using four image quality assessment tools and by conducting a user survey to assess their user perceived quality. We confirmed the initial presupposition that states that manipulating multiple image attributes at the same time is a complex problem. Also, looking at the survey results, we arrived to the conclusion that, in our scenario, similarity between an enhanced version and the original image, is more important to some extent, than improving its aesthetic value. This improvement can sometimes be exaggerated, causing the lost of useful contextual information or highlighting image defects. As such, a balance between similarity and aesthetic is desirable. Nevertheless, the attained results suggest that a modular and hybrid architecture like the one proposed, has potential in the area of image enhancement. Automatic image enhancement is also very closely tied with the capability of machine automated image quality assessment systems, and so progress in both areas are also intrinsically connected.

Keywords

Automatic Image Enhancement, Image Processing, Computer Vision, Machine Learning, Evolutionary Computation.

This page is intentionally left blank.

Resumo

Melhoramento de imagem é um procedimento da área de processamento de imagem onde uma imagem é manipulada de forma a que se adeque melhor a uma determinada tarefa, sendo por isso de grande relevância em múltiplas áreas, como por exemplo, imagem médica, imagem espacial, imagem biométrica, etc. Melhoramento de imagem pode resultar em diversos tipos de melhorias, por exemplo, o realce de uma características específica de forma a facilitar a subsequente análise realizada por uma máquina ou por um humano, ou melhoramento da sua percepção estética por um humano.

O objectivo deste trabalho é o desenvolvimento de um possível sistema para melhoramento automático de imagens, tendo como caso de estudo marketing digital de imóveis no contexto do projecto *"Indest - Indicador de composicion estética"*. Explorámos os estudos existentes em melhoramento de imagem e propomos uma pipeline com arquitectura *"end-to-end"*, que tira partido de técnicas clássicas, evolucionarias e de aprendizagem computacional presentes na literatura.

A estrutura apresentada é muito modular, pelo que permite a alteração de módulos e parâmetros. Os testes efectuados foram realizados sobre um conjunto variado de imagens de imobiliário, que nos foi fornecido. Os resultados dos testes foram avaliados utilizando técnicas de avaliação automática da qualidade de imagens e por um inquérito a utilizadores. Confirmámos o pressuposto inicial que indica que melhoramento de imagem manipulando múltiplos atributos, é uma tarefa complexa. Para além disso, olhando para os resultados do inquérito, chegámos à conclusão que, no nosso caso de uso, similaridade entre a imagem melhorada e a original, é algo mais importante do que puro melhoramento estético da imagem. Isto porque este melhoramento pode por vezes tornar-se exagerado, causando perda de informação contextual e realçando defeitos da imagem. Assim sendo, um balanço entre os dois é desejável. Apesar de tudo, os resultados obtidos sugerem que uma abordagem modular e híbrida como a apresentada, tem potencial na área de melhoramento de imagem. Melhoramento automático de imagem está também fortemente ligado à capacidade de avaliar correctamente e automaticamente a qualidade das imagens, e assim sendo, o progresso nas duas áreas está também intrinsecamente ligado.

Palavras-Chave

Melhoramento Automático de Imagem, Processamento de Imagem, Visão por Computador, Aprendizagem Computacional, Computação Evolucionária.

This page is intentionally left blank.

Contents

Acronyms

AHE Adaptive Histogram Equalization. 23

ANN Artificial Neural Network. 7

CLAHE Contrast Limited Adaptive Histogram Equalization. 23, 24

CNN Convolutional Neural Network. 7, 11, 17

EA Evolutionary Algorithms. 32, 34

GAN Generative Adversarial Networks. xviii, 7, 8, 10, 26, 46–48

GP Genetic Programing. xiv, xv, xvii–xix, 14, 32–37, 39–44, 48, 49, 51, 52, 54–56, 60

HVS Human Visual System. 5, 8, 23

IE Image Enhancement. xiii, 1–3, 5–11, 13, 14, 20, 22, 23, 34, 46, 48, 56, 59, 60

IQA Image Quality Assessment. xiii–xv, xvii–xix, 5, 6, 14, 15, 17, 18, 20, 21, 26, 31, 33, 34, 36, 46–49, 53, 54, 58–60

SSIM Structural similarity. xiii–xv, xvii, xix, 18, 19, 24–26, 34, 36, 39–41, 43, 45, 49, 51, 54, 56, 57, 59

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

Image Enhancement (IE) is an important and ever in demand branch of image processing and computer vision [3] that allows visual improvement to images by manipulating its various attributes. Since each digital image is essentially a matrix of discreet values, IE is done via mathematical techniques that alter those values in a specific way to achieve the desired result, resulting in an image that is usually in some way similar to the original one. In this chapter we will present the motivation for our work as well as the objectives we set and the planning we took to achieve them.

1.1 Motivation

IE has useful applications in several tasks and fields that benefit from having a way to pre-process an image in order to increase human or machine perception of key features. Such fields are medical imagery, space imagery, bio-metric, photography, video editing, among others [4, 5, 6]. Aesthetic improvement via IE is also one of the most popular uses of these techniques. A great example of such use of IE, is the post-processing that is applied to an image, before introducing it in a context where image appeal is crucial, like for instance, in advertisements.

There are multiple IE techniques that manipulate the image attributes in many different ways. Those attributes can influence the image brightness and colors, but also perform operations that increase its resolution or correct visual errors like noise. When an IE mechanism is used in a controlled environment, where the input images have similar features, and the desired output is also very well defined, the IE system can be tailored to the scenario it is being designed for. However, there are also situations where there is very little control over the input image attributes, and the desired output may vary. In such situations, tailored approaches would not work, and versatile systems, which combine image processing with meta-heuristics or machine learning, perform better, because they are able to adapt to each individual image by taking in consideration the image data before altering it. Online marketplaces are a good example of an environment with low control over the input and where automatic IE can be decisive in a product exposure. We are particularly interested in similar scenarios.

Although there is already a lot of research in this area, there is still a lot of work left to do. Approaches that automatically perform IE are not yet perfected, and there is no consensus on what method is, in general terms, better than other.

There is also an intrinsic problem with IE, which is the subjective matter of image quality assessment by humans. Since there are so many image attributes, the relevance of each one in the overall quality of an image is deeply idiosyncratic thus differing from person to person, which makes rating an image quality a difficult task.

1.2 Objective

The main goal of this work was the development an IE system capable of enhancing the visual aesthetic quality of real-estate images. To accomplish this, we relied on different techniques, arranged in an end-to-end layout, to formulate a system that aims to properly process images with different characteristics.

This work is being done while in contact with *Indestia*, a company from Corunha, Spain, in the context of the project *"Indest - Indicador de composicion estética"*, financed by *Xunta de Galicia*. They helped by providing useful tools and data during the project.

1.3 Planning

Having in mind the objectives described above, we stipulated a road-map to help us guide throughout the planned tasks. This road-map is a high level plan of the tasks we needed to accomplish in order to achieve our goal. The plan was as follows:

1. Design the initial architecture of the image enhancer system.
2. Implement and train a prototype version of it.
3. Test the system and analysis of the gathered results.
4. Iterate the architecture and implementation based on the results.
5. Perform statistical work and comparisons with other image enhancement system.
6. Write the final report.

For more details, see diagram in Figure 1.1, where a detailed schedule of how the planned tasks elapsed is provided. The main task structure remained and the work went as planned, despite few delays.

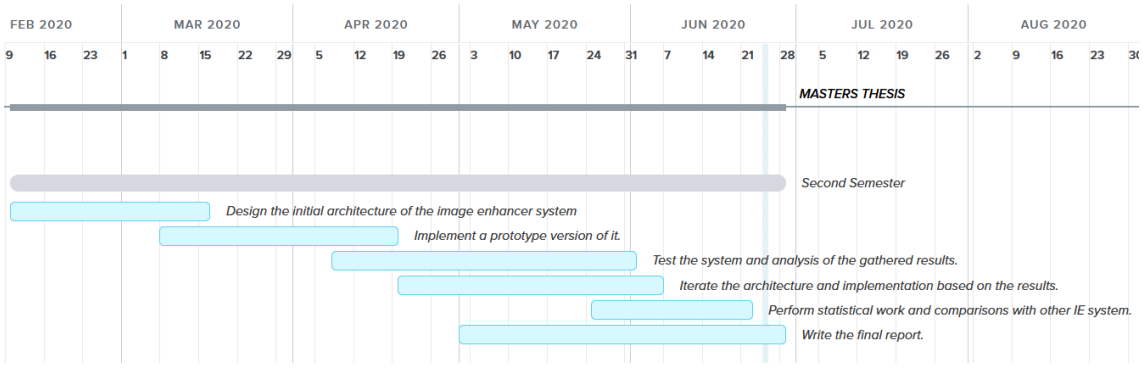


Figure 1.1: Gantt diagram for the second semester.

1.4 Document Structure

In this section we introduced the image enhancement problem, our objectives for this work and how we planned our work. The remainder of this document will have the following structure:

Chapter 2 presents a state-of-the-art section where the reader will be presented with a taxonomy for the IE research, as well as an overview of the existing techniques;

Chapter 3 discusses the idealized approach to the solution of our problems in order to achieve our objectives, that takes form as an end-to-end pipeline;

Chapter 4 contains all the practical work that was developed, as well as achieved results and performed analysis;

Chapter 5 collects all the main conclusions of this work and provides possible ways of expanding it;

This page is intentionally left blank.

Chapter 2

Context and State-of-the-art

Digital images are, now more than ever, a big influence in our daily lives, considering than almost every online activity relies, one way or another, on this form of media. Digital images can be constructed exclusively via software or by scanning an analog signal via image sensors. When the conversion from analog-to-digital occurs, a discretization also occurs, and with it, loss of information. This is due to the fact that every analog information is continuous, which is impossible to reproduce digitally [7]. As so, digital images can be represented as one or multiple matrix of discrete values.

Each image has a set of attributes like size, color space, contrast, brightness, saturation, distortions, artifacts, noise, format, etc, that define how we, and different computer software solutions, perceive it. All these features are not isolated but rather interact with each other. The image format and compression, for instance, is pivotal to every other aspect of the image, as different formats allow for distinct color and compression properties, that can in turn account for distinct visual characteristics. Many times this attributes are not well balanced between each other or optimized for the image context, and may cause a diverse range of deformations in the image quality.

During this chapter we will, first and foremost, explain why automatic Image Quality Assessment (IQA) is a difficult and important problem to solve. Then, we will perform an overview of the existing techniques in the Image Enhancement (IE) literature, while dividing them based on a presented taxonomy structure.

2.1 Image Quality Assessment

Although our work has its focus on IE, it is important to understand how the quality of an image can be measured.

L. He et al. defines image quality in three levels: fidelity, perception and aesthetics [8]. Fidelity is how well the image preserved the original information; Perception is how well the image is perceived according to every part of the Human Visual System (HVS). Lastly, the aesthetic of the image, which is the most subjective level, because it varies from person to person. It is also the most difficult to measure objectively because *"aesthetics is too nonrepresentational to be characterized using mathematical models"* [8].

IQA can be either subjective or objective. Subjective analyses is based on the how humans identify an image as good or bad. Mean opinion score (MOS) has became one of the most widely used subjective methods [9] and can be used to qualify the quality of any stimulus.

However, it relies on humans to qualify each stimulus individually, and such resources are not always available and are slow and expensive [10], so objective and machine automated methods take place. These methods can be divided into 3 categories: full-reference methods where the quality is determined by comparing with a ground truth image; reduced-reference methods where the reference is not fully available, only its features; and no-reference methods, where there is no reference available, as in most real-life scenarios [8]. While there are multiple developed full-reference and reduced-reference methods, no-reference methods continue to present a challenge in objective IQA [10]. Initial research focused on evaluating images based only on one particular feature or having a specific type of distortion in mind, as it is done in [11, 12], yet trying to develop a global no-reference method is still a difficult task. Almost every attempt to overcome this challenge, tries to do so by training a model using data acquired from a MOS method and then predicting the output with the said model [8]. These can be included in two different categories, two-step models and global models. [8]. Two-step models classifies each distortion individually and then combines it to achieve a final result [13, 14], whereas global models use every image feature without distinction [15, 16].

2.2 Image Enhancement

IE is a sub-field of computer vision and image processing [3] that aims to develop ways of improving the perception of an image specific feature or general quality, by a person or a computer, in a specific context [17].

Although, by definition, IE can be performed manually, we will focus on automatic IE achieved algorithmically or via machine learning. Automatic IE bears interesting challenges, specifically when trying to manipulate multiple aspects of the image at the same time, because individual features are not independent from each other. Furthermore, images may have areas that are too bright or dark to be possible to extract information from them, difficulting the task even more [18, 19].

Bellow in Figure 2.1 we propose a high-level taxonomy divided in three main layers, for IE techniques. The first layer from the top, *Environment*, divides the context where a technique is inserted in two categories, *Static* and *Adaptive*. *Static* means that, after the deployment of such technique, there are no changes to itself or surrounding components that affect said technique performance. *Adaptive*, on the other hand, implies the existence of some change to the technique or surrounding components, in order to improve its performance over time. The second layer, *Behavior*, divides the way a technique reacts to different inputs in two categories, *versatile* and *non-versatile*. *Versatile* means that the technique is capable of properly handling input images with different characteristics, whereas *non-versatile* techniques are specialized on a single type of image. The third layer, *Functions*, divides the technical foundation of each technique in 4 categories, *Classic*, *Evolutionary*, *Machine learning* and *Hybrid*. *Classic* refers to techniques based on mathematical and statistical evaluation and manipulation of the input image in order to improve it. *Evolutionary* and *Machine learning* refer to techniques based on evolutionary computation methods, which are inspired by biological evolution mechanisms, and machine learning methods, respectively. The *Hybrid* category encompasses techniques that take advantage of different technical foundations that belong in different categories. Finally, all these layers converge on the type of improvement trying to be achieved. Common approaches seek to improve images aesthetically by manipulating its brightness and contrast, but other types of manipulations are also possible, such as image super-resolution, automatic selective crop and artifact removal. Next, we will discuss what these means, as

well as present some research examples for each one.

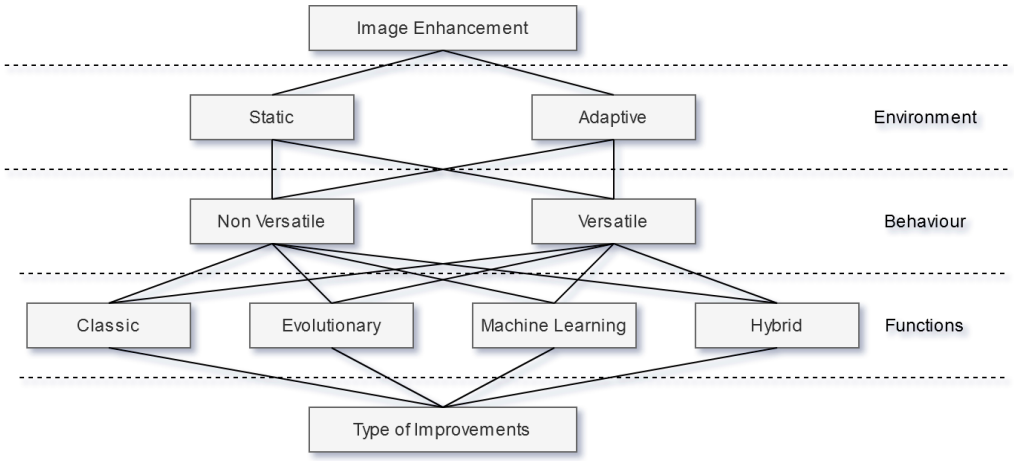


Figure 2.1: Proposed taxonomy for image enhancement techniques.

IE can be achieved by manipulating an image in two different domains [17, 18, 20]: spatial domain, and frequency domain. Spatial domain techniques manipulate the image directly on the pixels value and are usually easier to implement and understand, but less robust. Frequency domain techniques operate on the frequency space of images, and modifies the transform coefficients of the image, such as Fourier transform, discrete wavelet transform, and discrete cosine transform [18, 20, 21]. In the frequency domain, changes in position correspond to changes in the spatial values frequency, and each value represents the amount of intensity variance over a specific distance of the pixel values in the space domain. Some examples of spacial domain techniques are histogram processing algorithms, contrast stretching, and image threshold functions. Frequency domain methods are especially useful in feature extraction, noise reduction and edge detection / enhancement [22].

Image super-resolution aims to generate a high-resolution image from a single low-resolution image, which is useful to increase human interpretation of small details. Two recent examples are [23] and [24] that utilize machine learning techniques to enhance the resolution. The first example by *Chao Dong* et al. uses Convolutional Neural Network (CNN) to perform end-to-end mapping of the input low-resolution image to the output high-resolution image, in the three color channels simultaneously. CNN are a type of Artificial Neural Network (ANN) that is often applied to image processing. An ANN is an attempt to computationally simulate the network of neurons in the human brain, using a layered based architecture with multiple small processing components called "neurons". CNN are adequate to image processing because of its core operation, the convolutional operation, which allows to simplify the input information via feature extraction with specific filters that are applied sequentially throughout the whole image, detecting progressively complex shapes as we go to deeper layers. This also adds relevance to pixel context and neighborhoods and prevents features from only being detected if they are in a specific predetermined location. The second example by *Christian Ledig* et al. suggest *SRGAN*, which is another machine learning approach for the super-resolution problem. *SRGAN* focus on high frequency details when applying 4x upscaling factor. This is done using a Generative Adversarial Networks (GAN) with a novel loss function that takes in consideration the perceptual similarity instead of similarities in the pixel space. A GAN is a machine learning model architecture proposed by *I. Goodfellow* et al. in 2014 [25]. This architecture has two key components, the generator G and the discriminator D, where both G and D can be any function or network. The G purpose is to generate new data, and the D purpose is to

guess if the data is fake or not. This *min-max* game then leads to a result that will activate a loss function for either component, leading to alterations in the network that will then cause the generated data to be closer to the real one.

Automatic selective crop tries to improve images by cropping them in a way that leave out irrelevant and redundant features, and where the most detailed and important features remain. While not being the main focus of their recently proposed method, *Y. Deng* et al. shows in [26] a weakly supervised GAN learning feature based cropping without any ground truth pair.

Artifact removal seeks to mitigate the impact of visual distortions in the image, that appear especially in high frequency regions, when compressing an image with a destructive algorithm. *L. Galteri* et al. [27] developed a *"feed-forward fully convolutional residual network model trained using a generative adversarial framework"*. This means that a residual network is built to generate the new artifact free image. The generator is trained with the help of a discriminator, following the previously explained GAN framework.

As we are interested in enhancing the perceived aesthetics for humans, we should focus on aspects of the image that most appeal the human eye. The HVS is a heavily researched system, so we have some understandings of how it works [28]. Perceptual IE is a subcategory of IE that includes models that take in consideration the *"Principles of HVS"* to better enhance images [29]. Few of the most important principles that have played important roles in existing methods are: *contrast sensitivity function* that maps how the human eye reacts to different levels of contrast in different situations, *multi-scale and multi-orientation decomposition*, that explains how the human eye adjusts objects at various scales, orientations and distances; and visual masking, that refers to the phenomenon that occurs when an image appears to have lower contrast or brightness when surrounded by other stronger stimulus, called the mask [29]. We will not dwell deeper into this subject as it is out of our scope.

We have already showed that there are multiple types of IE techniques with different purposes and characteristics. Some are simpler static filters that are applied to the spacial domain, and others seek to adapt to the image context, preventing heterogeneous results across multiple images. Based on this research we will explore IE methods highly focused on improving image aesthetics, while dividing them in the three technical foundation categories presented previously in the taxonomy. We will explain the aim of each example as well as how they achieve the desired objective and what kind of results they obtain. We will start by presenting the classical ones.

W. Wencheng et al. recently proposed an IE pipeline that aims to improve the overall brightness and contrast of low-illumination images [30]. This pipeline starts by converting the image to a hue, saturation, value (HSV) space. This space conversion is done so that the brightness manipulation is performed without distorting the relationships between the colors, which is difficult to assure in the RGB space. The V component is then used to extract the illumination information, which is then used to adjust the parameters of the enhancement function. Two images are obtained through the enhancement process that are then fused utilizing a formulated strategy that extracts significant information from each one. Finally, the image is converted again to the RGB space. To extract the illumination information, a multi-scale Gaussian function is used. The enhancement function takes inspiration on the *Weber-Fechner* law that suggests that the brightness perceived by the human eye follows approximately a logarithmic function and so different illumination values would require different adjustments. For the image fusion technique a simple weighted sum is used where the weight of each image is determined using the principal component analysis (PCA) technique to extract feature values from each image.

The result showed significant improvement to brightness and contrast when compared to other classical approaches, while preserving the details. However, it is said that this method is computationally costly, so it is not viable to apply in video footage.

C. Y. Wong et al. proposed another pipeline approach that tries to bridge the problem where approaches that are only based on intensity enhancement may produce artifacts in "over-enhanced" regions and lack enrichment on color based features [31]. This pipeline is made up of the following parts: *color channel stretching, color space conversion, histogram equalization, equalization compression, and restoration of color saturation.* We will quickly go through each one of these. The color stretching is a pre-processing step that "stretches" each channel histogram to its maximum range, usually *[0-255]*. The space conversion is a pre-processing and a post-processing step, where the image is converted to the hue, value, intensity (HSI) space so that the intensity channel I is the one being manipulated. In post-processing, the image is again converted to the RGB space. Histogram equalization step follows a convectional histogram equalization implementation that aims to achieve $P_i = \frac{n_i}{N}$ where P_i is the probability of a pixel having intensity i, n_i is the number of pixels that have intensity i and N is the total number of pixels in the image. Equalization compression step tries to reduce the number of artifacts created by the equalization step by using a hyperbolic profile to adjust the equalization. Finally, a maximization operation is performed between the input and output saturation values. An experimental phase is conducted with a variety of natural scene images showing that this simple and efficient pipeline is a suitable choice to enhance color images.

H. Talebi et al. proposed in [32] a novel way of improving an image detail and contrast by expanding on Laplacian operators of edge-aware filter kernels in order to develop a robust method capable of enhancing the details of an image without compromising its overall quality by boosting noise and artifacts, which is a major concern when working with edge-aware filters. A structural mask of each image is created and is then used to blend the different image components that are calculated by progressively decomposing the image details. The results are promising as it is capable of correctly handling images with different characteristics and detail level.

Closing the classical techniques, we will succinctly go through the work of *S. Zhuo* et al. in [33], where a noise reduction pipeline is proposed. This pipeline starts with a hybrid camera system that takes a near infrared flash photo (N) at the same time as the normal photo (V) is taken. A noise reduction method is softly applied on V removing noise in lower frequency regions. After this, taking advantage of the fact that the near infrared image has the same visual structure as the normal one but without noise contamination, N is used to guide the high frequency *denoising* in order to maintain the majority of details. However, different materials reflect IR light differently, so some edges might be less evident in N. Yet, those edges are still visible in the smoothed version on V. In order to utilize the information in V, the RGB image is converted to *YIQ* color space and the intensity monochromatic channel Y is used in a variation of the first smoothing algorithm that has in consideration information in both images. At this point, the *denoising* is done and so the intensity channel is merged with the rest of the color components, and the image is converted back again to RGB color space. To further extract information provided by the near infrared image, a detail transfer function is also implemented and added to the pipeline. Furthermore, shadows and specularities that appear exclusively on the near infrared image are taken in consideration during operations done with N, so that the performance is not affected by such artifacts. Experiment results show highly effective noise removal and prove that using near infrared light for IE may by promising.

S. B. Kang et al. presented an interesting take on IE by introducing a framework that

explores the value of the personalization factor in automatic IE [34], where both classical techniques and machine learning models are used, making it an example of an hybrid approach to the problem. During an initial training phase, a database is built by presenting the user to a set of image examples in a simple interface that allows him to select the enhancement he likes the most in a set of enhanced versions of the original image. This database stores the original image features and the enhancement parameters that were choosen by the user. When a new input is received, a pre-processing phase takes place, where white balance and contrast stretch are performed, to make slight improvements to bad photos or photos that the system is not trained to handle. The system then searches for the most similar images in the database, and the parameters prompted by the user are used for the enhancement. The adjustments used in this work were *"white balancing via changes in temperature and tint, and contrast manipulation via changes in power and S-curves"*. To achieve a good result, the authors had to answer 3 important questions regarding the framework. How can we measure the similarity between a new image and the ones in the database? What is a good set of training images? And how to let any user navigate the possible enhancements set in the training phase? A new method for measuring the distance between images was tailor made for the problem, by measuring 38 different individual feature distances that would approximate images that require similar enhancement. To select the images that are shown to the users during the training, the top 25 most representative of the global set of 5000 images were selected using sensor placement problem [35] approach. Finally, to select which 8 enhanced images were presented to the user at any given time out of the 243 possible enhancements, machine learning was used. To answer the last question, a novel graphical interface that suits the problem was built. The results showed that in some scenarios preference assessment in IE can help overcome the subjective quality problem and therefore improve personal quality assessment.

We will now perform a more detailed analysis in a few examples of machine learning approaches. *Y. Deng* et al. introduced an adversarial learning model called *EnhanceGAN* in [26]. This model was already referenced earlier when providing an example of automatic cropping, which is also a feature of this model, but in this paragraph we will focus on its other enhancement properties. This model tries to make up for a common problem on fully-supervised approaches. They require a ground truth version pair for each dataset entry, making the dataset building process very costly. *EnhanceGAN* only requires weak supervision, that being binary quality labels on the dataset entries. This model is based on the GAN framework explained earlier. However, the generator G does not generate images itself, but rather learns an operation of transformation. This means that the work is extensible to other types of operators besides the ones presented. The ones presented are: a piecewise color enhancer that manipulates brightness and contrast in the luminance channel of the CIELab color space and chrominance in the other two channels; a deep filtering-based enhancer and the image cropping referenced early. The discriminator is a specific built network that tries to infer if the images quality is good or bad. The results produced by this model were subject to a user study that showed that they are on par with professional editing. It was also noticed that the outputs tended to approximate the image color pallet to color harmony schemes [36].

Y. Jiang et al. recently introduced another GAN based model dubbed *EnlightenGAN* [37]. The presented model has its focus on low light enhancement where it is even more difficult to construct a fully labeled dataset due to the difficulty of simultaneously taking a low- and normal-light picture of the same scenario. To tackle this problem, an unsupervised GAN that can be trained without image pairs, is proposed. The *EnlightenGAN* incorporates a self feature preserving loss to guide the training, since there is no strong external validation from labels. This helps by incentivizing the generator to maintain the feature distance

between the input and the output as short as possible. A self-regularized attention map was also added to the system to improve the enhancement on areas that need it the most. Another key aspect of this architecture is the global-local discriminator that tries to distinguish the *reals* from the *fakes* with attention to the global lightning and local low-light regions simultaneously, which is difficult with *"vanilla discriminators"*. The authors claim that the results of extensive experiments in objective and subjective image quality via user study, show that this method outperforms other recent methods in a variety of metrics.

C. Chen et al. took an interesting approach on extremely low light and low signal-to-noise IE [38]. Instead of manipulating the resulting image using classical techniques as it is done in the work of *W. Wencheng* et al. mentioned above, the proposed model is an end-to-end CNN architecture that receives as input the raw sensor data of the camera, replacing the traditional pipeline which rarely performs well under the short-exposure scenario presented. Although cross-sensor capabilities were out of the work scope, preliminary research show that it may also produce good results. To train the model, a dataset was also created and made available to the public. This dataset contains 5094 low-light images, each with a corresponding long-exposure version. The enhanced version can be the same for multiple low-light images. However, it is worth mentioning that the dataset only contains static scenarios. The conducted experiments on the model expose promising results in the noise suppression and color transformation tasks.

M. Afifi et al. recently presented in [39] a novel way of preforming white-balance within a camera image processing pipeline. An in-camera pipeline typically preforms the white-balancing procedure at the beginning of said pipeline, based on automatic selection by the camera or by manual user selection. It is then followed by a number of nonlinear procedures that alter the final image, making post-processing corrections to the white-balance very difficult. This work tries to mitigate that problem by proposing a pipeline that renders multiple *"tiny versions"* of the original image, as they are addressed by the author, each with a different white-balance setting. These renders present little processing overhead to the pipeline and allow the computation of mapping functions that can map the final image to any of these *"tiny versions"* color temperature, appearing as if it was rendered with those parameters by the camera pipeline. Furthermore it is possible to "blend" mapping functions allowing to map the original photo to any desired color temperature. The presented results showcase well the capability of this white-balance pipeline. *M. Afifi* et al. also recently presented a few others researches with reference to image white-balance and illumination in [40], [41], [42] and [43].

Lastly, *K. Zhang* et al. proposes a very deep architecture to perform image denoising [44]. As opposed to the presented work by *S. Zhuo* et al., where a near-IR camera setup was proposed to tackle the denoising problem, this work presents a novel very deep CNN architecture that manipulates the final image. It is developed while aiming to substantially decrease the run time and improve the denoising task when compared to traditional discriminative models. Experiments with this model showed that it is capable of removing Gaussian noise with an unknown noise level, which is again not possible in traditional discriminative models that are trained for specific levels. Moreover, the experimental results show that this architecture can handle other tasks like removing compression artifacts and perform super-resolution.

There are also some works that integrate evolutionary computation in the IE problem. *L. Rundo* et al. recently proposed a novel evolutionary method based on genetic algorithms to improve medical imaging systems [45]. *C. Munteanu* also proposed an IE method that relies on evolutionary techniques to improve gray-scale images by evolving the shape of the

contrast curve [46]. Other example is [47] by *S. S. Rajput* et al. in that applied differential evolution to the problem of face image super resolution to estimate the reconstruction weight vector for an input image. Results on the *FEI Face Database* dataset show that this method out-preformed the two popular methods to that problem.

Chapter 3

Approach

As we said in section 1.2, the objective of this work, is the development of an adaptive Image Enhancement (IE) system, and its application on real-estate pictures, in the context of the project *"Indest - Indicador de composicion estética"*. We had access to datasets addressed in section 4.2 and to an objective image qualification software for real estate images addressed in section 4.1. In this chapter, we will discuss how we approached the development of a solution to this problem.

Looking at the information shown in the previous chapter 2, we understand that evolutionary and machine learning based approaches can in fact produce models that better adapt to different kinds of inputs. Our research also shows that global enhancement, meaning the simultaneous enhancement of multiple features of an image, is a difficult goal to achieve, as the latent space of an IE solution is greatly vast. Common approaches to automatic IE, focus on individual and simpler tasks such as color balancing, in order to prevent over-enhancement where an image becomes too far-off from the original one, loosing important context sensitive information.

3.1 Proposed Pipeline Architecture

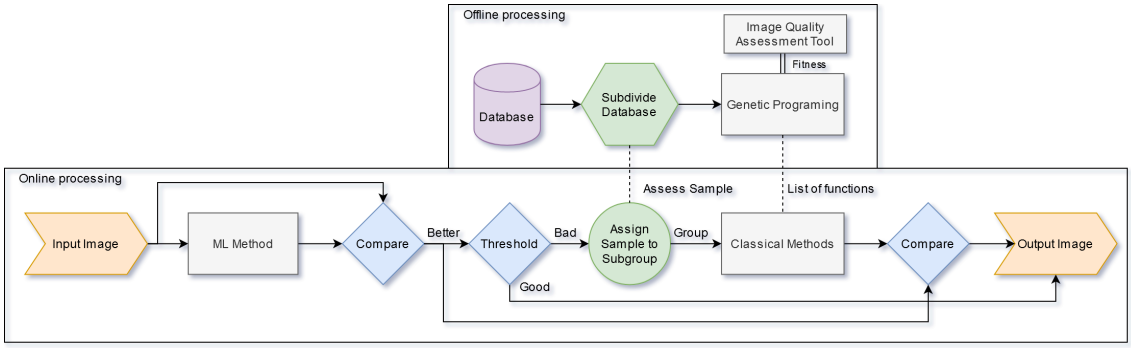


Figure 3.1: Schematic of the proposed end-to-end IE pipeline architecture. The pipeline is divided in two main sections, *online* and *offline*, containing the components computed in real-time over the input image, and the components independent from the input sample, respectively.

Having this in mind, we explored possible architectures that avail from all three methods, classical, machine learning and evolutionary, while maintaining the main focus on an end-to-end layout. To undertake the adaptive aspect of our desired solution, *i.e.* the capability of performing differently according to the input, we idealized a pipeline composed by both an *online* and *offline* section. An overview of the proposed pipeline architecture is demonstrated in Figure 3.1, and further detail of both *online* and *offline* modules is given in the next sections.

In order to have a better understanding of our results, we performed statistical work to infer how does our solution stand against one other solution that tries to achieve the same objective. We also conducted a user survey on Image Quality Assessment (IQA) to further evaluate the obtained results. Important to note that this architecture is very modular as it allows for the replacement of almost any component with a similar one. This facilitates any change to the pipeline in order to improve it or tailor it to a distinct context. The final solution will be supplied to *Indestia*, the company from Spain we are working with.

3.1.1 Offline Pipeline Section

The offline section is responsible for all the processing that is not possible to be executed in an online manner without heavily compromising the pipeline's time efficiency. Thus, the offline operations can be continuously performed and updated, while being fed to the online section and consequently altering the pipeline performance.

In our scenario, any evolutionary computation had to be done offline considering the large time consumption inherent to such techniques. We planed to take advantage of such components to evolve the conditional set of implemented classical filters to be applied to each input sample, utilizing the score of IQA tools as guide for the evolution. Throughout the development, we gave special attention to Genetic Programing (GP) techniques as our evolutionary component, since these techniques are designed to evolve program trees, which is similar to our required solution. The implemented classical functions were simple functions that altered the image in a specific way. We implemented functions that manipulated image color balance, contrast, saturation, brightness, noise and sharpness.

GP is usually utilized to optimize specific problems, meaning in the IE context, the improvement of a single image at a time. Nonetheless we seek a somewhat general solution that is capable of showing improvement on a diverse set of images. To do this, in each evaluation, we evaluated each individual on multiple different images from the database. Furthermore, we addressed the possibility of dividing the database in groups containing images with similar characteristics, and evolve a solution over each one of the above-mentioned groups. During the online processing, the input is assigned to one of the groups, and the corresponding set of functions is applied. More details on this will be presented in Section 4.6.

3.1.2 Online Pipeline Section

The online section, on the other hand, is responsible to perform all the operations that are applied to the specific input image. It is allocated to quicker tasks such as making use of trained machine learning models and performing classical manipulations to the input image. We also propose the introduction of a *comparison phase* that guarantees the best output possible out of the multiple iterations computed during the pipeline processing. This introduces some overhead in the pipeline. However we further propose the introduction of

a *threshold phase* that filters images whose quality can presumably be even more enhanced, from those considered to have such quality that attempting to improve them even further, may degrade their quality. More details on this components will be presented in Section 4.8.1.

Regarding the machine learning component, we were inclined to explore adversarial and convolutional networks as they have continuously indicated to perform well on image manipulation tasks. We focused on a recent architecture that utilize adversarial learning to perform image-to-image translation [48]. This component would help to subtly improve the overall image quality, and so it is placed at the beginning of the pipeline as a way to provide an *head start* for the following methods. More details on this will be presented in Section 4.7.

Pseudo-Code in Algorithm 1 sums up the pipeline's online decision process, where *IQA* represents a IQA function that returns a score based on the input image's quality; *ML* represents a Machine Learning component in the pipeline; and *EC* represents the set of classical functions applied to an image that is returned by the evolutionary component.

Algorithm 1: Overview of the pipeline's online decision process

```
Input: Input Image
Output: Output Enhanced Image
I = Input Image
if IQA(I) > IQA(ML(I)) then
    if IQA(I) > Quality Threshold then
        return I
    end
    if IQA(I) > IQA(EC(I)) then
        return I
    else
        return EC(I)
    end
else
    I = ML(I)
    if IQA(I) > Quality Threshold then
        return I
    end
    if IQA(I) > IQA(EC(I)) then
        return I
    else
        return EC(I)
    end
end
```

This page is intentionally left blank.

Chapter 4

Experiments in Image Enhancement

In this chapter, the performed experimental work is presented along with its results. It will have a *section-based* structure, where each section tackles a distinct topic, based on the approach presented in Chapter 3. We will begin by presenting the used metrics as well as the datasets that were available during the experiments.

4.1 Image Quality Assessment Tools

We propose the development of a system that outputs a more suitable version of an input image, in the online real-estate marketing use case. Since we are dealing with image quality measurements, which most of the times derives from a subjective appreciation, it is very important that we have a deterministic and automated way of qualifying an image quality. To tackle this problem, we made use of 3 distinct *no-reference* Image Quality Assessment (IQA) tools, and two well known *full-reference* ones. As presented in Section 2.1, a *no-reference* IQA method is one that is capable of scoring an image quality without a reference image. *Full-reference* methods, on the other hand, require a reference image to be able to score an image, and thus are useful for comparison purposes.

4.1.1 *PhotoILike*

PhotoILike is an IQA tool provided by *Indestia*, the external company from Corunha, Spain. This tool is a closed source, third-party, black-box software that receives as input a single image and returns a value from 1 to 10, where 1 means the worst quality and 10 the best quality. Note that the calculated score is not solely based on the images aesthetic, but rather on multiple features considered relevant for real-estate marketing. For instance, the baseline score of a pool picture, is much higher than the baseline of a bathroom one.

4.1.2 Neural Image Assessment

Neural Image Assessment [49] (*NIMA* for short), is a *no-reference* IQA tool based on a deep Convolutional Neural Network (CNN), proposed by *H. Talebi* and *P. Milanfar*. The paper highlights how the same architecture, trained with different datasets, leads to state-of-the-art performance in predicting both technical and aesthetic scores. As the paper states, technical judgment has in consideration features like "*noise, blur, compression artifacts, etc.*". One the other hand, the aesthetic evaluation "*quantifies semantic level*

characteristics associated with emotions and beauty in images". The technical prediction model was trained using the *TID2013* dataset [50], where 25 reference images and 3000 distorted images with different level and types of distortions are made available along the a *Mean Opinion Score* for each image. The aesthetic prediction model was trained using the *AVA* dataset [51], which contains over 250,000 images with human labeled meta-data, including semantic labels for each "*photographic style*" and large numbers of aesthetic scores for each image. Both provided models predict the final score as an average of a distribution of scores between 1 and 10 where 1 means the worst score and 10 the best. Both models were used during the experiments, and require each input image to have a resolution of 224 by 224 pixels.

4.1.3 Blind Image Spatial Quality Evaluator

Blind Image Spatial Quality Evaluator [52] (*BRISQUE* for short), is a *no-reference* IQA tool, proposed by *A. Mittal* et. al. As opposed to the previous methods, *BRISQUE* is based on a set of classical feature extraction procedures that computes a collection of 36 features per image. Those features are then fed to a *support vector machine* model trained using the TID2008 dataset [53]. This tool originally outputs a value between 0 and 100 where 0 represents the best quality and 100 the worst. However, in order for the outputs to be in concordance with the previously presented methods, the output was mapped to a 1 to 10 range where 1 means is the worst score and 10 the best. *BRISQUE* is used as metric in other image enhancement works like the ones in [54, 55, 56, 57, 58].

4.1.4 Full-Reference Methods

Two *full-reference* methods were used during this work as a way to perform comparisons between images, *mean squared error* (MSE) and Structural similarity (SSIM) [59].

MSE is a simple *full-reference* method that computes the average squared difference between two images. The formula is presented bellow.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{4.1}$$

SSIM, instead of comparing images in a pixel-by-pixel manner like other metrics, aims to approximate to the human perception by using groups of pixels and using *luminance*, *contrast* and *structure* to guide the comparison. It returns a number between 0 and 1, where 1 means the images are 100% similar. It is defined by $SSIM(x,y) = l(x,y)\times c(x,y)\times s(x,y)$ where l, c and s are *luminance*, *contrast* and *structure* respectively, and those can be in turn defined by:

$$\begin{cases} l(x,y) = \dfrac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x,y) = \dfrac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x,y) = \dfrac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{cases}$$

Where μ is the mean of the values of each image, σ is the standard deviation of the values in each image and σ_{xy} is the covariance between the two images. C constants are very small constants added to avoid a division by 0. In multi-channel images, the SSIM is calculated for each channel and then averaged to obtain the final score.

4.2 Datasets

To assess the quality of this system and to train the machine learning components, large quantities of images are required. To tackle this problem, we made use of two distinct sets of data.

4.2.1 MIT-Adobe FiveK Dataset

The MIT-Adobe FiveK Dataset [60] is a collection of 5,000 photos taken with SLR cameras by multiple photographers, where each image has 5 paired versions, making a total of 30,000 image files. Each of the five versions was retouched by a photography student in an art school, using *Adobe Lightroom* as the dedicated photo adjustment software. The retouchers task was to make the original image as visually pleasant as possible. We will make use of this dataset in Section 4.5.1.

4.2.2 Provided Datasets

We were given by *Indestia*, a dataset of 432 real estate photographs separated in three categories, *"good"*, *"medium"* and *"bad"*. In addition, a set of enhanced versions of each one of the 432 images was also made available. These enhanced images resulted from applying a third-party automatic Image Enhancement (IE) software, named *One-Click*, to each original image. This tool, based on our observation of the dataset, tries to improve the quality of an image with brightness and saturation adjustments. Figure 4.2 shows two examples of image improved by *One-Click*. The provided enhanced images served as ground truth reference for our experiments as well as a comparison target for our results. Besides this, we were also given an extra dataset of 12,090 real estate photographs and landscapes taken with a wide variety of cameras, not just SLR as the previously presented FiveK Dataset, resulting in a wider range of image quality and resolutions. Not only that, but some samples presented black or white bars to the side or above and bellow the actual image, logos watermarks, and some were close up pictures of, for instance, blank walls. Some examples are showcased in Figure 4.1 However this *aberrations* may very well be part of a real-life scenario, so we take those in consideration during our tests and results. Each dataset entry had a paired enhanced version and also had a label score given by the same IQA tool. All the images were provided in "*.jpg*" format.

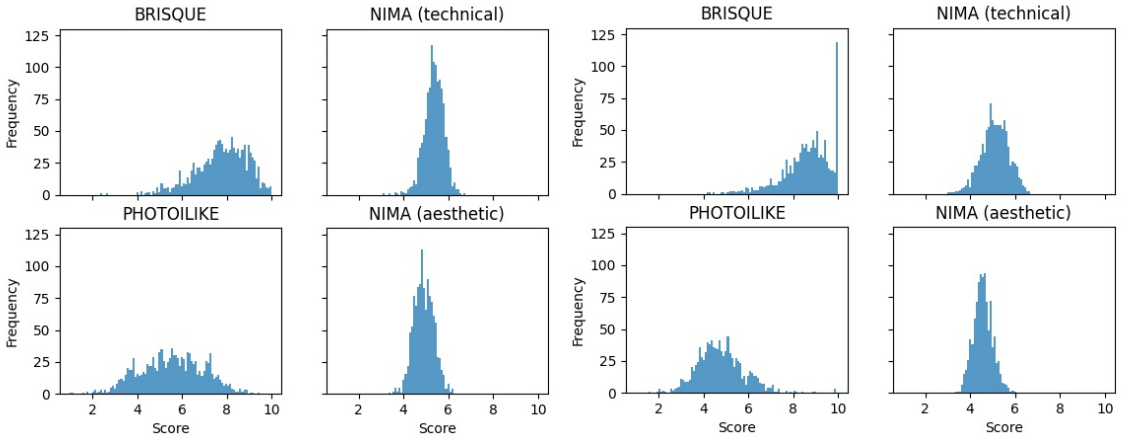
Figure 4.1: Example set of images considered *aberrations*.

4.2.3 Test Sub-Dataset

In order to examine our results in an unbiased way, we separated in an early stage, 10% of the 12,090 images from the rest of the dataset. All the performed experiments were

Figure 4.2: Example set of images enhanced by *One-Click*.

conceived on those 1,209 images. As a way of having a baseline for our measurements, we examined the original images of this test set, using all the IQA metrics presented in section 4.1. The same tests were also performed on a subset of 1,000 images from the *FiveK* Dataset.



(a) (b)

Figure 4.3: (a) Graphical representation of the test dataset scores, computed by all four *no-reference* IQA tools used during the experiments. | (b) Graphical representation of the scores of 1,000 samples from the *FiveK* dataset, computed by all four *no-reference* IQA tools used during the experiments.

Figure 4.3 represents two sets of four histograms. One set for each dataset, and one histogram for each metric. Note that the *full-reference* methods are not displayed as they are purely for image comparison sake. It is noticeable that, the *BRISQUE* tool is much more propitious of giving the maximum possible score than any other metric. Another relevant thing to note is that, as the *FiveK* dataset contains multiple different images such as faces, animals and close up objects, rather than only real estate pictures, it performs poorly in the *PhotoILike* metric, when compared to the test dataset.

4.3 Classical Approaches and Preliminary Work

A set of 7 classical and previously documented methods were re-implemented and used during this work. In this section we will go through these different methods, explaining how they work and what type of improvement they offer. The preliminary results collected during the first semester are presented as well. This work was also a deliverable to *Indestia*, the company we are working with.

4.3.1 Implemented Classical Functions

The 7 implemented methods focused on five main aspects of IE: contrast adjustment, brightness adjustment, color balance, noise removal and edge enhancement, also referred as *sharpening*. We will now present and explain each one individually.

Contrast Stretching

Contrast in image processing is the range of intensity values available to an image. The contrast stretching is a point operation method that, as the name implies, tries to improve the image contrast by linearly increasing the difference between the maximum intensity value and the minimum intensity value in an image, therefore increasing the contrast level. It is necessary to specify the new lower and upper limit of the image range, which is usually as large as the image supports. However, it is important to keep in mind that the standard implementation of this technique is very susceptible to outliers, as the transformation is made linearly and a single pixel can set the input minimum or maximum value. The linear formula to implement this technique is presented bellow, where P refers to the pixel value and *in* and *out* refers to the input and output, respectively.

$$P_{out} = (P_{in} - Min_{in}) * \frac{Max_{out} - Min_{out}}{Max_{in} - Min_{in}} - Min_{in} \tag{4.2}$$

Since we are working with colored images, this was done individually to all three color dimensions, red, green and blue, which were then merged back to form a full RGB image. Studies such as [61, 62, 63] discuss and use this technique in IE.

Histogram Equalization

Histogram Equalization is another method that tries to improve the quality of an image by manipulating the contrast. It does this by spreading out the most common intensity values by the less common ones, increasing the global contrast of an image. Bellow is the histogram equalization formula, where k denotes every intensity level, I denotes the maximum intensity value available to the image, n the number of pixels with a certain intensity value, and MN the total number of pixels in the image.

$$out_k = (I - 1) \sum_{i=0}^{k} \frac{n_i}{MN} \tag{4.3}$$

Notice that, unlike the aforementioned contrast stretching, this method is not linear and so does not keep the overall histogram "shape". This means that in some cases, the image

quality could be hindered in the process, specifically in colored images, where the relations between color channels change. This technique is highly used and there are multiple interactions of it. [64, 65] use it and discuss its results.

Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE) is yet another contrast enhancement method. It's an iteration of the Adaptive Histogram Equalization (AHE) technique, that in turn, is an improved version of the regular histogram equalization.

AHE differs from the histogram equalization because instead of manipulating the whole image histogram, it divides the image in smaller tiles and applies the equalization in each tile's histogram. The resulting tiles are put back together using interpolation techniques. This is useful for images that have both underexposed and overexposed regions.

CLAHE improves upon the AHE by clipping the maximum intensity values of each region and redistributing the clipped values uniformly throughout the histogram before applying the equalization. This helps to reduce the noise amplification noticed when using the AHE, when a tile shows intensity levels fairly similar throughout the entire image. Works presented in [66, 67, 68] explore some use cases for this method.

Gamma Correction

Gamma Correction tries to accommodate the fact that the Human Visual System (HVS) perceives brightness in a non-linear way, whereas digital cameras and monitors tend to capture and display it linearly. This is done by scaling each pixel brightness from *[0 - 255]* to *[0 - 1]* and applying an expression to map the original values using the following power-law expression: $O = I^{\frac{1}{G}}$, where I denotes the input, G the gamma value and O the output. A couple of usage examples for this method are presented in [69, 70].

Non-local Means Denoising

Non-local Means Denoising [71], as the name implies, tries to reduce the existing noise in an image. It replaces the value of each pixel in each channel to the average of similar pixels. Because of computational limitation, the search it restricted to a window of adjustable size.

Unsharp Masking

Unsharp Masking is an IE technique that sharpens the edges of an image. It does that by subtracting a blurred version of the original image from the original image to create a "mask". This mask is then applied to the original image, enhancing edges and details. Two works that are based on this approach are presented in [26, 72].

Simplest Color Balance

Simplest Color Balance was proposed by *N. Limare* et. al in [73]. The algorithm is presented as the simplest color balancing procedure possible, and assumes that the highest R, G, B value corresponds to white and the lowest corresponds to black. The algorithm

tries to remove incorrect color cast by scaling each channel histogram to the complete *0-255* range via affine transform. Since it is common for images to have outliers pixels, a small percentage of pixels in both extremes of each histogram are saturated to either black or white. The saturation level is an adjustable parameter that correlates with the quantile percentage of saturated pixels. Important to understand that this method was not added at the beginning of the project so it *does not* take part in the preliminary experiments.

4.3.2 Preliminary experiments and results

At first, we implemented all methods in a *C++* script running on *GPU* inside a *Docker* container. Then, during a first approach, each filter was applied individually to the *bad*, *medium* and *good* dataset introduced in section 4.2.2. Finally, we measured the similarity between the resulting images and the respective ground truth reference using the full-reference SSIM metric. A visualization of this image was also calculated using the method *subtract* from *OpenCV*. It is very important to regard the fact that all the preliminary experiments were made using the default parameterization for every classical filter.

Table 4.1 presents the experiment results, by showing the average approximation value to the reference of each method in each subset. We calculated the approximate value as follows: (i) we calculate the similarity between the original and the reference image; (ii) calculate the similarity between the new and the reference image (iii) subtract the values of (i) to (ii). A positive number indicates the new image is closer to the reference image than the original, whereas a negative number indicates that the new image is less similar to the reference image than the original.

	Bad	**Medium**	**Good**	**Total Average**
Contrast Stretching	0.10	0.03	0.04	0.06
Histogram Equalizer	-13.93	-15.95	-17.81	-15.90
CLAHE	-1.77	-2.47	-3.02	-2.42
Gamma Correction	-2.61	-2.39	-2.18	-2.39
Non-Local Means Denoising	0.33	-2.46	-2.14	-1.42
Unsharp Masking	-2.90	-0.16	-0.38	-1.15

Table 4.1: Average approximation to the reference image, per method and per subgroup *Bad, Medium, Good*, using SSIM metric. The approximation values are calculated by averaging the subtraction of the SSIM score between the original and the reference image, for each subset. The value was then multiplied by 100 to ease the readability.

Looking at numbers in the previous table, we notice that neither one of the 6 methods presented high results, with just one (contrast stretching) presenting positive average results. However, it is important to note that a low SSIM score when comparing the resulting image with the ground truth reference, does not mean a resulting bad image. An enhanced image can present improvements to the original in multiple aspects, which means there are various ways of improving the visual appearance of an image, *i.e.* there are multiple enhancement solutions to each image. The diagram in Figure 4.4 helps to visualize this. We also provide examples where the new image is subjectively better visually than the ground truth reference despite the low values that indicate that it is under-performing according with the established metric.

It is essential to remember that the implemented methods rely on classical techniques. These can indeed be very effective and efficient but they often lack versatility. We also came to similar conclusions with our own experiments. Despite the fact that the CLAHE

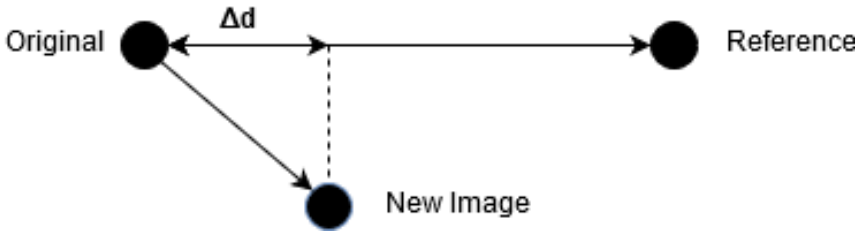


Figure 4.4: Visualization of the approximation to the reference where Δd represents the approximation that the new image got in relation to the original.

performed poorly in the SSIM metrics, even moving away the resulting image from the reference, this prooved to be the method that showed the single biggest approximation to the reference both in the *"bad"* and *medium* subset, and one of the best in the *"good"* subset. This particular result solidifies the idea that similarity metrics have their caveats which enticed us to search for other ways to evaluate the images.

(a) (b) (c)

Figure 4.5: (a) Original Image from the *"bad"* sub-set | (b) Reference Image | (c) New image created using CL, CS and NL with an approximation to the reference of -13.83 using the SSIM metric multiplied by 100

(a) (b) (c)

Figure 4.6: (a) Original Image from the *"bad"* sub-set | (b) Reference Image | (c) New image created using CL and CS with an approximation to the reference of -11.24 using the SSIM metric multiplied by 100

Lastly, it is worth mentioning the fact that all these tests were performed with only one method, but multiple cumulative methods can sometimes provide better visual results. The set of images in Figure 4.8 show an example of this statement.

(a) (b) (c)

Figure 4.7: (a) Original Image from the *"medium"* sub-set | (b) Reference Image | (c) New image created using CL, CS and UM with an approximation to the reference of -2.50 using the SSIM metric multiplied by 100

(a) (b) (c) (d)

Figure 4.8: (a) Original Image | (b) New image created using only CL | (c) New image created using only CS | (d) New image created using both CL and CS

4.4 Preliminary exploration in Machine Learning

We then started exploring approaches that made use of machine learning, specifically Generative Adversarial Networks (GAN) based, and performed tests and research in order to increase our insight of the problem. We used the pre-trained model of the *EnlightenGAN* [37] referenced earlier, in the *bad*, *medium* and *good* datasets. At this time, *PhotoILike* was the only *no-reference* IQA tool we had available to us, so it was used as metric for our outputs, to help us understand how state-of-the-art architectures behave with our dataset and metric.

	Bad	**Medium**	**Good**	**Total Average**
***EnlightenGAN Improvement* - μ**	0.63	0.12	-0.17	0.20
***EnlightenGAN Improvement* - σ**	0.48	0.45	0.44	0.46

Table 4.2: Average improvement and standard deviation of the EnlightenGAN model per subgroup *Bad, Medium, Good*, using *PhotoILike*. The values follow the same scale as the IQA tool, so bigger standard deviation values mean more divergent scores.

We can quickly infer that the improvements made by this method are substantially bigger than any classical approach and that the *"bad"* subset was also the one where the improvement was the greatest. However, we can also note that the standard deviation of the values is relatively big and, in the *"good"* dataset, the average quality of the image, according to the *PhotoILike* tool, decreased. This can be due to specific parameterization in the pre-trained model but also shows that even for more adaptive and advanced methods, images that are originally good may be a challenge to the enhancement problem.

4.5 Image Dataset Segmentation

As mentioned in the previous Chapter 3, we wanted to explore the possibility of fragmenting the offline database in order to restrict each offline evolution to a set of images with similar visual characteristics. The input image is assigned to one the the division sets, and the solution for that set is then applied. In this section we will go through our attempts to unravel this problem along with the obtained results.

4.5.1 Image Clustering

The first possibility we explored was clustering the database based on image features thought to be relevant for our problem. It is important to observe that image clustering is a significant and very challenging problem in computer vision. Adding to this, in our scenario we were dealing with unlabeled samples, meaning that our clustering is unsupervised, difficulting not only the clustering task itself but also the results analysis.

Manual Feature Selection

As of the beginning of searching a solution to this problem, we developed a set of 5 functions that focused on extracting 5 distinct features. Those features are: noise, contrast, saturation, brightness and sharpness. To extract the noise, we used the work proposed in [74] to quickly estimate the images Gaussian noise. For contrast, we calculated the *RMS* contrast [75], meaning standard deviation of pixel intensities. For saturation, we averaged the pixels intensity in the S channel of the HSV color system. For brightness, we used the *HSP* color system [76], as it grants a brightness value closer to the real human perception, when compared to the luminance (L) channel of the *HSL* or the value (V) channel from the *HSV*. We then averaged the perceived brightness (P) channel to obtain a final value. Finally, for sharpness, we applied a Laplacian filter, calculated the variance of the output and used that as a sharpness score, as it is purposed in [77]. An array of all these 5 features was normalized and was used to feed a clustering algorithm.

Feature Extraction via Auto-Encoder

In addition to the manual feature extraction referred in the previous section, we considered the idea of using *auto-encoders* as an automatic feature extractor. *Auto-encoders*, originally proposed in [78], are a particular neural network architecture used to perform data encoding in an unsupervised environment [79]. The training goals of these types of networks is to approximate as much as possible its output to its input, while filtering data with a *bottleneck layer*. Figure 4.9 displays a simple schematic structure of an auto-encoder with 3 fully connected hidden layers. In that example, the most internal layer z, contains the encoded information that can be used as the feature set.

We implemented a variation of a conventional *auto-encoder* using *Keras* for *Python* [80], where instead of fully connected layers, we utilized convolutional layers in order to preserve image spacial-dependent information. This convolutional *auto-encoder* architecture was firstly proposed in [81].

We tested two similar networks, where one was deeper and compressed the data further than the other. The building blocks for the network, were sets of convolutional and *Max-Pooling* layers for the encoder, and convolutional and *UpSampling* layers for the decoder.

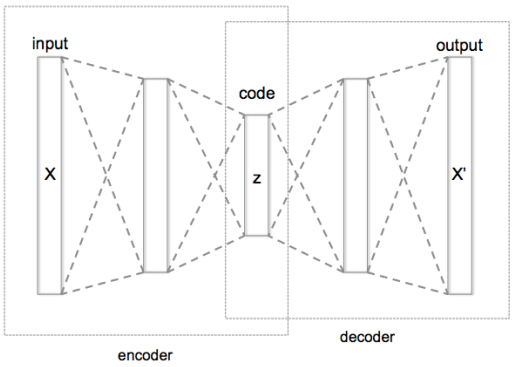


Figure 4.9: Schematic structure of an auto-encoder with 3 fully connected hidden layers. The z layer is the most internal layer that contains the encoded information. [1]

All convolutional layers use the same 3 by 3 kernel with a *reLU* activation function, padding *"same"* and stride of 1, due to the fact that these are commonly used for convolution operations. *MaxPooling* and *UpSampeling* used the default *Keras* parameterization besides the padding where *"same"* was also defined. Both networks use *ADAM* optimizer, a *binary cross-entropy* loss function to guide the training. The input layer has a size of 128 by 128 by 3 where the third dimension represents the *RGB* channels. One encoder had a sequence of 64, 32, 16, 8 and 3 convolutional filters and the other just 32, 16, 8 and 3, making the size of the encoded layer of 4 by 4 by 3 (48 features) and 8 by 8 by 3 (192 features) respectively.

Clustering Techniques

We made use of two clustering techniques, the classical *K-Means* clustering and *Mean-Shift* clustering. *K-Means*, originally referenced in [82], aims to divide n data-points into k predefined groups or clusters. It does that by selecting k points randomly, defining those points as the cluster centers, or *centroids*, and assigning each point to the closest centroid. As such, the final solution is highly dependent on the initial set of selected points. To minimize this, the algorithm repeats this process multiple times, and selects the solution with less within-cluster variances, i.e. minimizes the squared errors of the distance metric for each cluster. However, *K-means* has some noticeable weaknesses. First and foremost, the fact that the number of clusters has to be manually defined is not ideal in an unsupervised problem such as ours. Then, the susceptibility to initial conditions also proves to be prejudicial. Also, its arithmetic distance measurement makes it very vulnerable to outliers that deviate the cluster and, in its original variation, the one that was used in our tests, *K-means* assumes clusters of similar shape and density.

Mean-Shift clustering, proposed in [83], has a distinct characteristic from *K-means*, as it does not require specifying the number of clusters in advance, and it is instead calculated by the algorithm with respect to the input. It can be classified as a mode-seeking algorithm, as it locates the *maxima* (modes) of a density function based on a kernel function. The algorithm iteratively moves each data point towards the highest density of neighbors based on the density function, which will eventually lead to the center of the cluster. Despite this, this approach has a few problems such as being relatively computationally expensive, and not scaling well with data dimensionality.

Results

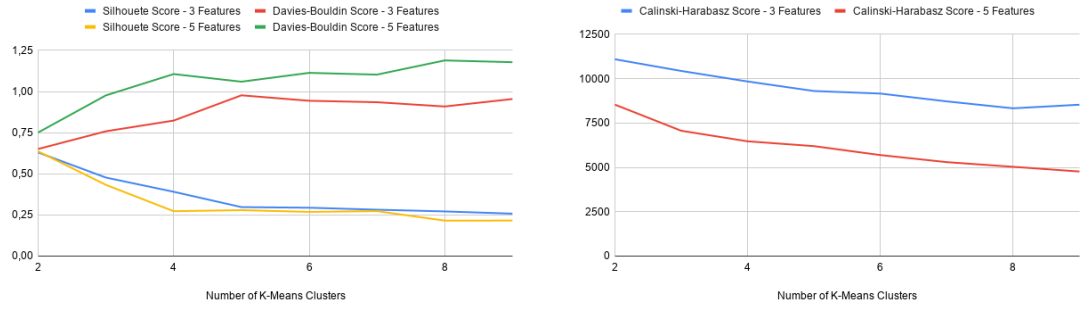
As a main metric for our unsupervised clustering problem we used the well-known *Silhouette Coefficient* presented in [84]. The *Silhouette Coefficient* is computed using the mean of the *intra-clusters* distances and the mean of the *nearest-cluster* distances, for each point. The final score is the result of an arithmetic average of all the sample scores. The best coefficient score is 1, the worst is -1, and values near 0 indicate overlapping clusters. The formula for calculating the coefficient for each sample can be defined as

$$(b - a)/max(a, b)$$

where a is the mean of the *intra-clusers* distances and b is the mean of the *nearest-cluster* distances. We also employed two other additional metrics to qualify the cluster model: *Calinski-Harabasz Index* proposed in [85] and *Davies-Bouldin Index* proposed in [86], both able to score unsupervised clustering models. *Calinski-Harabasz Index* is the average ratio of the sum of *between-clusters* dispersion and of the sum of *intra-clusters* dispersion for all clusters, dispersion being the sum of distances squared. Better clusters correspond to higher scores. *Davies-Bouldin Index* qualifies the separation between clusters based on centered distance and cluster size. Values closer to zero signify better cluster separations. All the metrics implementation are from the *Scikit-Learn* for *Python* [87].

The experiments started by extracting all the 5 features from 10,000 train samples from the provided dataset. Then, we used both clustering techniques to try to cluster those images based on their features. We experimented with all 5 features and with only the 3 considered most relevant and with less correlation between them.

As *K-means* requires a number of clusters as parameter, and we had no way of knowing which was the best, we repeated the same test using 2 to 9 clusters as it seemed an adequate range of clusters for our problem. All the *K-means* parameters besides the number of clusters were kept as default. The same is true for the *Mean-Shift* algorithm. Note that to make sure the *kernel* for the distribution function is fit to our data, *Scikit-Learn* provides the function *sklearn.cluster.estimate_bandwidth*, that automatically estimates the *kernel* bandwidth.



(a) (b)

Figure 4.10: (a) Graphical representation of the evolution of the *Silhouette* and *Davies-Bouldin* metrics across multiple K-means clusters. Higher *Silhouette* score is better, lower *Davies-Bouldin* is better | (b) Graphical representation of the evolution of the *Calinski-Harabasz* metric across multiple K-means clusters. Higher *Calinski-Harabasz* score is better.

Looking at the results in Figure 4.10 and Table 4.3, we can understand that, according to all three metrics, the less features used the better are the results. Additionally, we can

	Silhouette	***Davies-Bouldin***	***Calinski-Harabasz***
Mean Shift - 3 Features	0,77	0,24	935,67
Mean Shift - 5 Features	0,58	0,46	570,44

Table 4.3: Clustering using *Mean Shift* algorithm evaluated by *Silhouette*, *Davies-Bouldin* and *Calinski-Harabasz* metrics. Higher *Silhouette* score is better, lower *Davies-Bouldin* is better, higher *Calinski-Harabasz* score is better.

see that, using *K-means* with only two clusters demonstrated the best results, along with using *Mean Shift* with 3 features. Regardless, we noticed that in both *Mean Shift* models, the cluster prediction revealed only two clusters where one completely "dominated" the other, having more that 93% of the samples in both cases. *K-means* with 2 clusters also showed a similar issue yet less attenuated, where one cluster possessed around 84% of the total samples.

Regarding the experiments with the *auto-encoder*, the bottleneck of the networks was too tight for the network to produce good results, and at the same time produced features in a highly dimensional space, making it very hard to cluster [88]. A comparison between the input image and its decoded version is in Figure 4.11. It is evident that the networks did not perform well, and so, we did not executed clustering experiments on the extracted features. Even so, it is interesting to observe that, naturally, the size of an *auto-encoder* bottleneck layer plays a large role in its output quality

(a) (b) (c) (d)

Figure 4.11: (a) Input example | (b) Decoded Output from the network with a bottleneck of 8 by 8 by 3 | (c) Input example | (d) Decoded Output from the network with a bottleneck of 4 by 4 by 3

We considered neither results to be satisfactory. This can be due to the intrinsic difficulty of the image clustering problem itself [89], but was surely enlarged by extremely heterogeneous and unlabeled dataset used. Much could be done to improve the results, such as improve the clustering techniques, improve the encoding networks and research different sets of features, although, the idea of performing image clustering was abandoned, as further research in this topic was considered out of the work scope.

4.5.2 Dividing Distribution Scores

Following the observation of the results presented in 4.5.1, we pondered the possibility of a simpler solution. Observing the distributions of the test set in Figure 4.3, we considered the possibility of dividing the dataset based on the IQA tools score. The sections are divided by the *33* and *66* 100-quantiles . The k_{th} q-quantile of a variable X is defined in Equation 4.4, where P means probability. We conducted tests where we divided the test dataset based on *NIMA*'s score distribution. In theory, this approach should work better as the number of quantiles increase.

$$P(X < k_{th}) \leq \frac{k_{th}}{q} \wedge P(X \leq k_{th}) \geq \frac{k_{th}}{q} \wedge P(X \geq k_{th}) \geq \frac{k_{th}}{q} \tag{4.4}$$

4.6 Evolving Classical Filters Sequence

Results from the preliminary work demonstrated that classical filters may struggle with versatility but can perform well if under the right conditions. Besides that, it is made clear that applying different filters sequentially can produce unique results and that slight adjustments in the order of said filters may cause great changes to the output. Contemplating these conclusions, we tried to develop a way to automatically compute a set of classical functions that better suits different kinds of input images. In this section, we will describe the experiments regarding such objective, along with the achieved results.

4.6.1 Genetic Programming

Genetic Programing (GP) is a technique developed to evolve programs, that can be viewed as a subset of Evolutionary Algorithms (EA), since both share the same core principles. These principles are inspired by biological evolution mechanisms, such as *crossover*, a process involving swapping information between two individuals in order to create a new one, and *mutation*, where a random part of an individual is replaced by another random part. As in nature, *mutation* probability of a mutation occurring is much lower than the *crossover* probability. [90] These operators are then applied on a population of individuals, producing new offsprings that integrate the next generation [90].

Usually, in EA, each individual represents a candidate solution to a problem defined by the fitness function. Based on this function score, each individual can have a different chance of "reproducing" and having some information passed to the next generation. Because of this, choosing an appropriate fitness function is one of the most crucial and difficult tasks when using EA. Even so, there are many other parameters in a EA that affect the decision process of choosing the right individuals to breed, how to breed them and how often it can happen, meaning the probability of crossover and mutation. This architecture makes these algorithms very useful in optimization problems.

In GP, the individuals represent functions or programs rather than a specific candidate solution to a problem. Each individual is generally represented as a tree with depth 3, constituted by primitive nodes and terminal nodes. Figure 4.12 represents a simple program tree, where numbers and variables represent the terminal set, and the operators represent the primitive set. Tree depth is the distance from the root node to the terminal or leaf nodes.

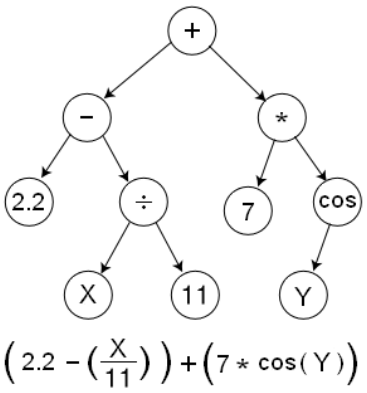


Figure 4.12: Visualization of a simple function represented as a tree structure. [2]

All the implementation was done using *DEAP* [91] for *Python* as the base evolutionary

engine.

Primitive Set and Conditionals

Making use of GP in a new problem, requires the definition of the primitives and terminals that are going to be available to the population during the evolution. In our scenario, we wanted to evolve a sequence of filter functions that generally received at least an image and a numeric value as input. This is not true for the *HE*, *CS* and *CLAHE* filter that received zero, zero and two parameters respectively.

Knowing this, we defined a primitive set containing all the seven classical functions previously implemented and a terminal set containing the input image, and an *ephemeral constant* ranging from -1 to 1. An ephemeral constant is simply a constant that assumes a random value within a pre-defined range at creation and keeps that value unless it suffers from a mutation. The defined range was then mapped by each function in order to adapt it to the desired magnitude. Each function parameter range was manually defined so that the function provided acceptable results.

Additionally, another group of primitives was added in order to allow the output program to generate conditional results. This means the introduction of an "*if-then-else*" function that, depending on the boolean value of a condition, returns the output of the "*then* tree" or the "*else* tree", allowing the same program solution to behave differently according to the input characteristics. To make this possible a set of "conditional functions" had to be introduced, so we adapted and added to the primitive set, the 5 functions presented in Section 4.5.1, that extracted each one a relevant feature from an image. All these functions were modified to expect an image as input along with an ephemeral constant, that serves as a threshold for that condition. Figure 4.13 shows a graphical example of a possible individual.

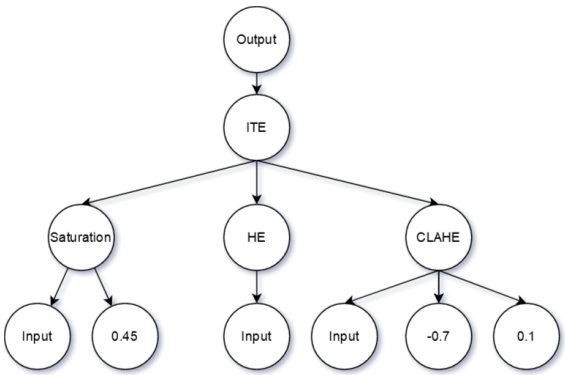


Figure 4.13: Graphical example of a possible individual. The numbers represent the ephemeral constants, the *ITE* node represents the *if-then-else* primitive and the "Saturation" node represents the conditional function.

Fitness Function and Bloat control

As we stated, defining the fitness function is one of the most crucial steps of building a GP solution. In our case, we wanted a fitness function that evaluated each individual based on its output visual quality, and so, we logically thought of using the IQA tools presented in section 4.1. As the *PhotoILike* tool had in consideration other aspects beyond the input visual quality it would not make a good fitness function for our problem. Thus,

we selected the *NIMA* tool as it produces results closer to *state-of-the-art* when comparing with *BRISQUE*.

We performed multiple tests using both the technical and the aesthetic model, individually and together. Additionally, as a way to make sure that the output does not deviate too much from the original image, meaning degradation of the contextual information, we also used SSIM to regulate the evolution. Other attempt to undertake this issue, was to cap the tree depth at a relatively low depth, preventing the solution to apply excessive manipulation to the input. This can also be understood as *bloat* control procedure. Bloat is a common problem in GP where the candidate solutions increase in size without increasing in fitness performance, causing parts of the solution to be redundant. A common and simple approach to this problem is to impose a tree depth cap and prevent any solution to over-grow.

It is of high importance to mention that the fitness function played a central role in our experiments as most of the other parameters were set across experiences.

4.6.2 Performed Experiments and Results

In this subsection we will go through all the performed experiments regarding the GP component.

We began by evaluating the original dataset with the 4 *no-reference* IQA methods. After that, with the purpose of having a criterion for all the future experiments, we measured the improvement achieved by *One-Click*, the external IE software, on the test dataset using all available metrics. In addition to this we also conducted the same experience on a list of classical functions, manually arranged by us based on expertise and personal tastes, with the default parameterization. This list was as follows: *Contrast Balance, CLAHE, Unsharp Masking, Non-local means denoising and Contrast Stretching*. This two experiments will be an essential "cornerstone" for the rest of the result analysis. The results are presented in table 4.4 and 4.5, and in Figure 4.15. In benefit of readability, *NIMA* models are abbreviated to respective initials and *PhotoILike* is abbreviated to *PHIL*. While interpreting the results, we will focus more or SSIM rather than *MSE* since SSIM gives a much more valid metric of similarity, nonetheless *MSE* will still be presented, rounded to units. All the results are presented in Table 4.4 and show the metrics improvement over the original images scores. Negative improvement means degradation of quality according to the respective metric. An output example of both methods is presented in Figure 4.14.

(a) (b) (c)

Figure 4.14: (a) Original image as input | (b) *One-Click* output | (c) Classical List output

Afterwards, we started the GP experiments. We set some parameters in order to properly restrict the work scope. We used values considered default for crossover and mutation probability of *75%* and *5%*. In EA, mutation helps local exploration of the solutions

	NIMA A.	***NIMA T.***	***BRISQUE***	***PHIL***
Original - μ	4,91	5,36	7,77	5,56
Original - σ	0,45	0,43	1,15	1,41

Table 4.4: Average (μ) and standard deviation (σ) of the original images from the test dataset using 4 *no-reference* metrics. All the metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality.

	NIMA A.	***NIMA T.***	***BRISQUE***	***PHIL***	***SSIM***	***MSE***
One-Click - μ	0,16	-0,02	-1,19	0,39	0,80	5250
One-Click - σ	0,25	0,23	0,69	0,69	0,07	4175
Classical List - μ	0,43	-0,06	-0,08	0,25	0,66	3805
Classical List - σ	0,31	0,26	1,86	0,64	0,09	2407

Table 4.5: Average (μ) and standard deviation (σ) of the improvement made by *One-Click* and a list of manually selected classical functions, on the test dataset. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 the highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

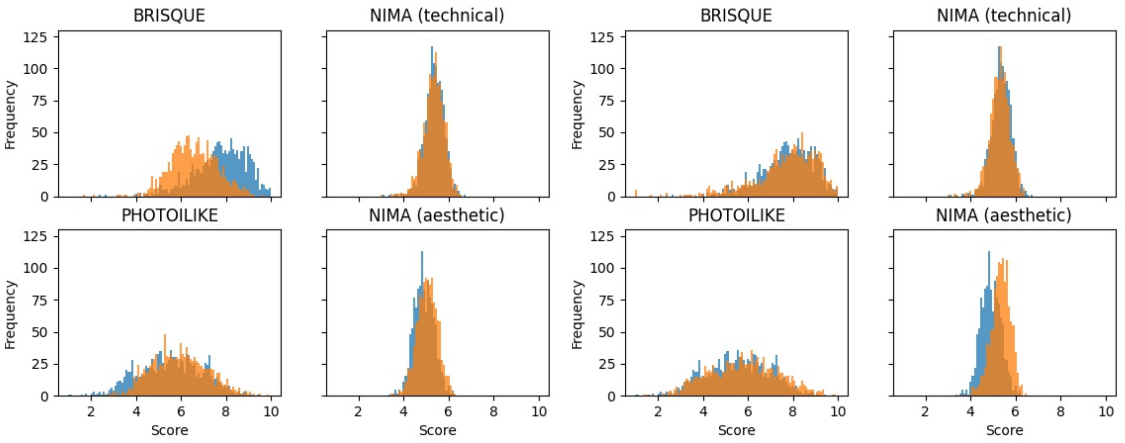
space, whereas crossover promotes the global exploration of the solutions space. Besides this we also set the mutation and crossover operators, the selection method and the tree generation mode, across all experiments.

For the crossover operator, *one-point crossover* was used. This operator randomly selects a point in two individuals, and swaps the sub-trees from that point, creating two new individuals. For mutation, *uniform-mutation* was used. this operator simply selects a random node from the tree, and replaces its sub-tree with a tree with a depth between 0 and 2. Mutation can also happen to ephemeral constants in which case the constant will assume a new value from the same predefined range *[-1,1]*. The selection method used was *tournament-selection* with tournament size of 3 [90].

This new population of individuals is subject to the crossover sequentially, *i.e.* the individual x_i is mated with the individual x_{i+1} and the two offsprings replace the parents. Each parent is only mated once. After the crossover process is over, the mutation operation occurs over the offspring population. This final population is transferred to the next generation.

The used tree generation mode was *Ramped half-and-half*, where 50% of the times all leaf nodes of the generated tree have the same depth between a pre-defined minimum and a maximum (*Full*) and the other 50% of the time the leaf nodes can have a different depth between the same minimum and a maximum (*Grow*). All the used operators are relatively simple, therefore it is possible to further expand this research by studying the impact of each operator in the overall algorithm performance. Each test was made with an initial population size of 80 and 150 or 200 generations depending on the test. This limit was imposed due to time constrains. All the GP configuration used is presented in sum in Table 4.6. The operators and probabilities used are considered standard [90].

As mentioned in Chapter 3, it is not the objective of this experiments to produce a solution that improves upon a specific image. Instead, we endeavor a solution as generalist as possible. To achieve this, it is necessary to do a fundamental change to the typical GP technique. Each solution will be evaluated on a set of 10 randomly selected images, an the average fitness of all images, will be considered the fitness of the individual. In addition to



(a) (b)

Figure 4.15: (a) Graphical comparison between the original test dataset (blue) and the results from the *One-Click* (orange), computed by all four *no-reference* IQA tools used during the experiments. | (b) Graphical comparison between the original test dataset (blue) and the results from the list of classical functions manually selected (orange), computed by all four *no-reference* IQA tools used during the experiments.

Cross-over	One-point
Mutation	Uniform. Adds a tree with depth between 0 and 2
Selection	Tournament Size 3
Tree Generation	Ramped half-and-half
Population Size	80
Number of Generations	150 / 200
Cross-over probability	75%
Mutation probability	5%

Table 4.6: Summary of the GP configuration used during the experiments.

that, to further prevent overfitting to a specific group of images, a new set of 10 images is selected in each generation. For this reason, it is expected a significant variation in fitness from one generation to another. In all performed experiments, the individual with the overall highest fitness was selected as the test subject.

Initially, a couple of experiments were done using each *NIMA* model individually as fitness, and a maximum tree depth of 10, during 150 generations. The same experiments were then reiterated with a maximum tree depth of 5. Table 4.7 and Figures 4.16 and 4.17 show the results of these experiments.

Looking at the results, a few initial conclusions stand out. There was a very significant improvement to the aesthetic component of the images, considering the initial distribution of the scores (Figure 4.16 and 4.17), when using the aesthetic model as fitness. However, even when evolving the solutions with the technical model as fitness, the final score in the same metric, was negative.

Additionally, it is worth mentioning that the SSIM lowers when the maximum depth is higher and when the aesthetic model is used in fitness evaluation. This makes sense, as increasing the depth of the trees allows the solution to apply more filters, altering more the original aspect of the input. Moreover, the technical component as well as the SSIM

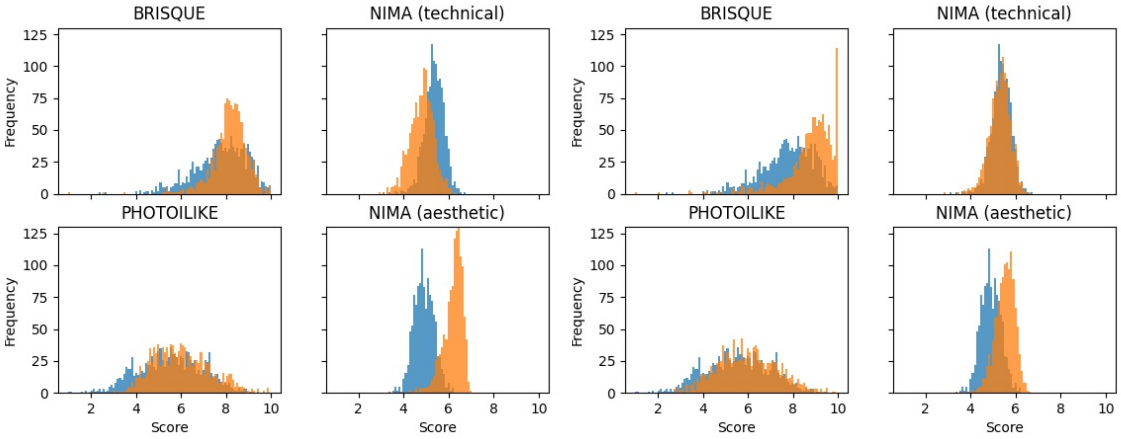
	Fit.	*Depth*	*NIMA A.*	*NIMA T.*	*BRISQUE*	*PHIL*	*SSIM*	*MSE*
μ	***N.A.***	10	1,30	-0,55	0,40	0,47	0,47	13213
σ	***N.A.***	10	0,45	0,43	1,10	0,96	0,09	4262
μ	***N.T.***	10	0,62	-0,06	0,99	0,43	0,70	6045
σ	***N.T.***	10	0,34	0,26	1,42	0,77	0,08	3828
μ	***N.A.***	5	1,15	-0,46	0,08	0,35	0,63	8175
σ	***N.A.***	5	0,41	0,41	0,91	0,90	0,07	2408
μ	***N.T.***	5	0,42	-0,03	0,52	0,24	0,75	4638
σ	***N.T.***	5	0,28	0,22	1,50	0,70	0,10	3967

Table 4.7: Average (μ) and standard deviation (σ) of the improvement and similarity using four GP configurations on the test dataset. *"Fit."* indicates the used fitness function where *N.A.* and *N.T.* mean *NIMA Aesthetic* and *NIMA Technical* respectively. *"Depth"* indicates the maximum depth of the tree. The highlighted cells are the ones where the score is measured by the model used as fitness. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

metric demonstrate to have an inversely proportional relation to the aesthetic score.

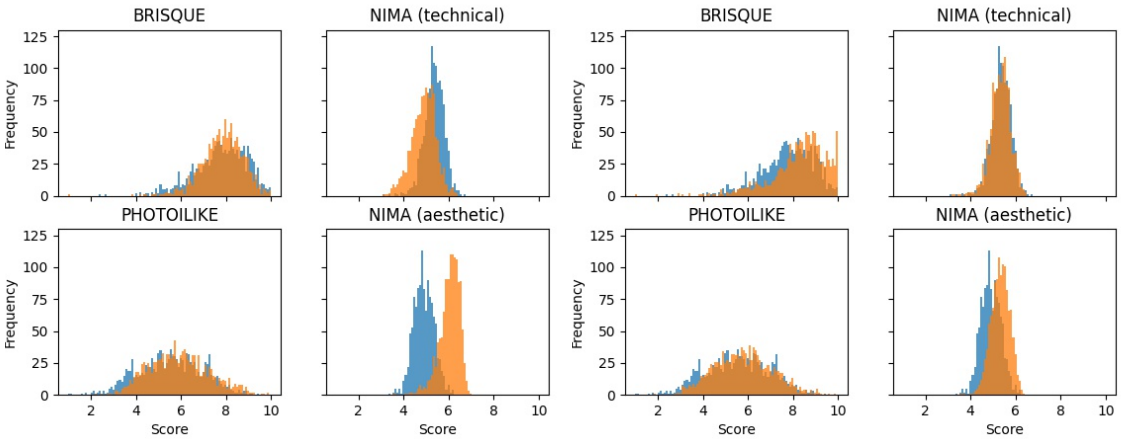
Figure 4.19 and 4.20 show output examples for these experiments. Using the *NIMA* aesthetic model can in fact produce very interesting results, that almost resemble paintings instead of pictures. The model's evaluation tends to associate higher scores with over-edited and saturated images, as both the 5 depth and 10 depth experiments seem to be converging to a similar style, even if the five depth results are more lenient. Other conclusion we can extrapolate from these results is the fact that, improvements of the same magnitude as those of this test will also mean extreme adulteration of the original image, even if they are considered good for the aesthetic model and *PhotoILike*, showing that this tool is also conducive to high aesthetic scores. We must evaluate the following results with this in mind.

Regarding the technical model, it appears to prefer sharpening and contrast increasing operations, which can also result in the noise and artifacts enhancement, contradicting the desired results, making it a very *"fragile"* feature. This *"fragility"* is perceivable in the evolution of the fitness functions (Figure 4.18 *(b)* and *(d)*) that unlike the aesthetic mode, struggles to maintain a high score. These graphs also demonstrate what was previously stated about the fitness oscillation due to the batch of images used in fitness assessment changing each generation.



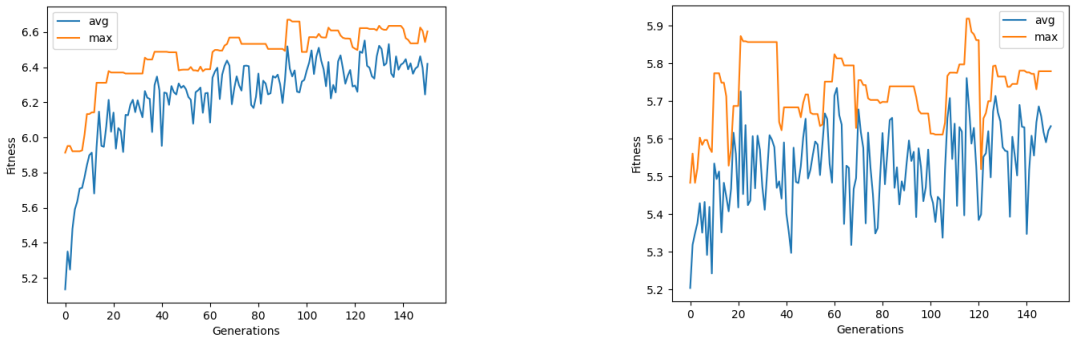
(a) (b)

Figure 4.16: (a) Graphical comparison between the original test dataset (blue) and the results from using *NIMA* aesthetic as fitness and tree depth of 10, computed by all four *no-reference* IQA tools | (b) Graphical comparison between the original test dataset (blue) and the results from using *NIMA* technical as fitness and tree depth of 10, computed by all four *no-reference* IQA tools



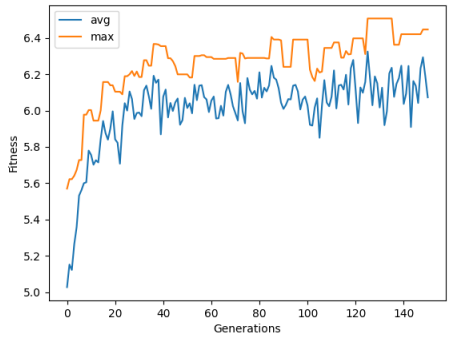
(a) (b)

Figure 4.17: (a) Graphical comparison between the original test dataset (blue) and the results from using *NIMA* aesthetic as fitness and tree depth of 5, computed by all four *no-reference* IQA tools | (b) Graphical comparison between the original test dataset (blue) and the results from using *NIMA* technical as fitness and tree depth of 5, computed by all four *no-reference* IQA tools

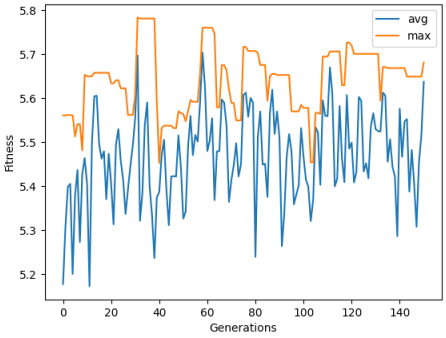


(a)

(b)



(c)



(d)

Figure 4.18: (a) Fitness evolution during 150 generations using the aesthetic model, with a depth of 10 | (b) Fitness evolution during 150 generations using the technical model, with a depth of 10 | (c) Fitness evolution during 150 generations using the aesthetic model, with a depth of 5 | (d) Fitness evolution during 150 generations using the technical model, with a depth of 5.

As a side note, it is interesting to observe that a plain white image, performs really good on the technical model, with a score of *5.85* whereas the aesthetic model classifies it as a *3.77* image. All of these results go according to the *NIMA*'s models definition by the authors addressed in section 4.1.2. An additional examination reveals that, solutions from GP where the depth was capped at 5, did not make use of conditional functions. This is understandable as adding a conditional function would "use" one depth level, which is 20% of the available tree depth. Likewise, aesthetic evolution did not make use of conditional functions as it promotes over-enhancement.

In spite of the interesting and visually intriguing results produced by the aesthetic model, they altered the original image too much, making it not suitable for our use case. As such, a followup experiment was performed in which the fitness function was the arithmetic average of both models scores, and the maximum depth was set to 5 to make sure the output solution would not provoke exaggerated enhancement. The results are presented in Table 4.8 and a comparison with the previous tests is made in Figure 4.21. Figure 4.22 shows an example of an output from this test.

The experiment results proved to be positive, as we achieved a respectable aesthetic score, while preserving much of the images similarity to the original (SSIM), and with much less reduction of the technical score. Even though, looking at the results, it was considered that there was still an over-enhancement on the images, enough to make small artifacts in

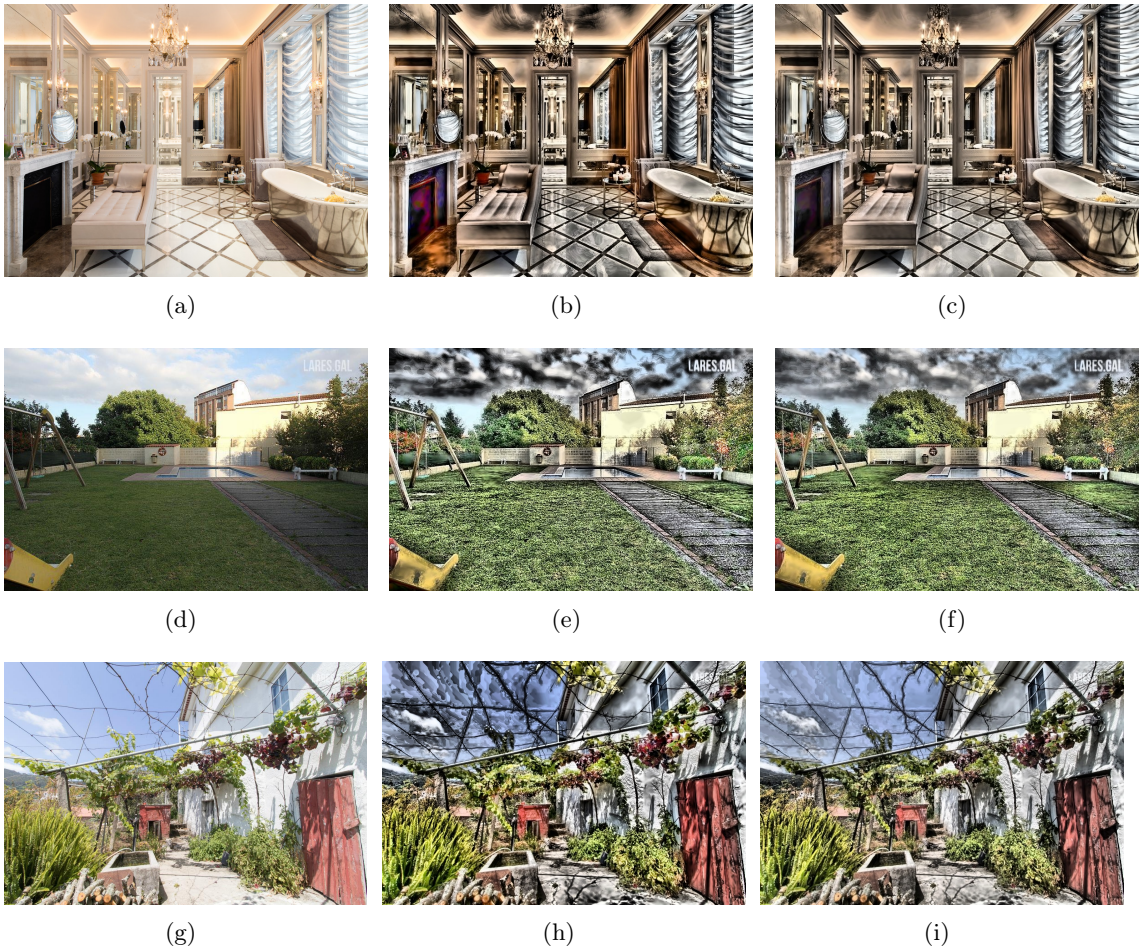


Figure 4.19: From left to right: Original image | GP output with depth 10 and using *NIMA* aesthetic as fitness. | GP output with depth 5 and using *NIMA* aesthetic as fitness.

the original image, stand out a lot. As previously mentioned, we are working with "*.jpg*" images, a lossy compression method, that often leads to image artifacts in high frequency regions. Both scores from SSIM and *NIMA*'s technical model denounce this problem. An example of this problem is available in Figure 4.23.

We further extended our research by idealizing and experimenting with other fitness function. We proposed to guide the evolution not only based on each image aesthetic / technical quality, but also on the similarity to the original image. Several tests were made using $NIMA * SSIM$, where *NIMA* corresponds to one of the models, as fitness function. From here on out, all the tests were performed with 200 generations each and a maximum tree depth of 10, in view of the fact that the fitness function can now regulate the appearance of over-enhancement, thus giving "space" to the conditional functions. Adding to this, we also put in practice the division of the image database, discussed in Section 4.5.2. A separation based on the *NIMA*'s aesthetic distribution percentiles was made and so, two division points at 33% and 66% divided our dataset in three equality populated groups. Hereupon, three GP evolution processes take place, one for each group.

The results in Table 4.9 immediately demonstrate that the new fitness function was successful in regulating image over-enhancement but it was strict, causing an undesired stagnation in the image improvement, as it can be observe by the SSIM and *MSE* values. These experiences resulted in very slight manipulations to the images, mostly being very minute

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.20: From left to right: Original image. | GP output with depth 10 and using *NIMA* technical as fitness. | GP output with depth 5 and using *NIMA* technical as fitness.

sharpening and gamma adjustment operations. One of the experiences, the one using the *NIMA* technical model, even classified as the best individual the one who did nothing to the image, returning the input as output. It is worth mentioning that this was the first time where a test presented positive improvement across all metrics, even if very short. This endorses the claim that greatly improving an image across all metrics is an arduous task.

To confront this issue we ran the same experiments with an alteration to the fitness function. This time, *NIMA*'s score was squared, removing relevance from the similarity score and increasing the evolution more propitious to altering the image. This iteration of the fitness function can be described as $NIMA^2 \times SSIM$. Table 4.10 showcases the obtained results. A comparison between the results of the previous fitness $NIMA \times SSIM$ and this one, is also available in Figure 4.24

The new experiments outcome proved to be much more balanced between the similarity metrics and the aesthetic and *PhotoILike* score. As expected, as the aesthetic score increased, the technical score suffered a slight decrease. With regard to the dataset division, both sets of experiences brought evidence of moderately balancing the results of the aesthetic and SSIM score, meaning increasing the aesthetic score when it was too low, and decreasing it when it was too high, in benefit of the technical and similarity appraisal.

Looking at the numerous images results and respective metrics performance, we also concluded that SSIM scores in the range of *0.6* to *0.8* usually indicates, in our scenario, a

	Fit.	*Depth*	*NIMA A.*	*NIMA T.*	*BRISQUE*	*PHIL*	*SSIM*	*MSE*
μ	$\frac{A+T}{2}$	5	0,91	-0,19	0,44	0,39	0,73	6401
σ	$\frac{A+T}{2}$	5	0,36	0,31	1,12	0,80	0,09	2178

Table 4.8: Average (μ) and standard deviation (σ) of the improvement and similarity using one GP configurations on the test dataset. *"Fit."* indicates the used fitness function which represents the average score between the aesthetic and technical model. *"Depth"* indicates the maximum depth of the tree. The highlighted cells are the ones where the score is measured by the model used as fitness. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

	Fit.	*Div.*	*NIMA A.*	*NIMA T.*	*BRISQUE*	*PHIL*	*SSIM*	*MSE*
μ	$N.A. \times SSIM$	No	0,04	0,01	0,20	0,11	0,99	16
σ	$N.A. \times SSIM$	No	0,08	0,08	0,50	0,60	0,01	16
μ	$N.T. \times SSIM$	No	0	0	0	0	1	0
σ	$N.T. \times SSIM$	No	0	0	0	0	0	0
μ	$N.A. \times SSIM$	Yes	0,21	-0,06	-0,11	0,22	0,96	332
σ	$N.A. \times SSIM$	Yes	0,22	0,15	0,84	0,62	0,03	276
μ	$N.T. \times SSIM$	Yes	0,04	0,01	0,39	0,11	0,99	17
σ	$N.T. \times SSIM$	Yes	0,08	0,09	0,53	0,59	0,01	18

Table 4.9: Average (μ) and standard deviation (σ) of the improvement and similarity using four GP configurations on the test dataset. *"Fit."* indicates the used fitness function where *N.A.* and *N.T.* mean *NIMA Aesthetic* and *NIMA Technical* respectively. *"Div"* indicates it dataset division was used. The highlighted cells are the ones where the score is measured by the model used in the fitness function. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

great equilibrium between not changing an image and overly modifying it. The previous conclusions on the use of conditional functions remain in part true, as the GP solutions from finesses containing the aesthetic model continue not using the conditionals. The new solutions from experiments with the technical model also demonstrated a disinterest in conditional functions but without making use of all the depth available. Instead, the technical model tended to propose shorter and less impactful solutions to all images.

We did not perform any more exploratory tests on this topic, despite that, research can still be further extended in various strands. Expanding or altering the set of functions available to the GP algorithm, fine tuning evolution parameters, more experimentation with the fitness measurement and changing the evolution dataset are evident ways of enlarging the scope of this work.

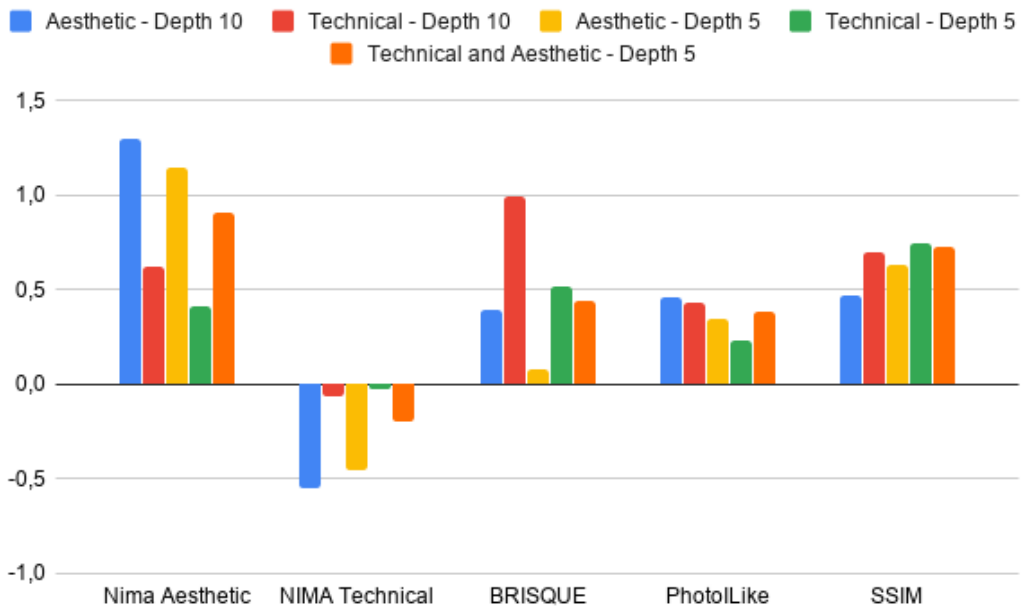


Figure 4.21: Graphical comparison of the improvement obtained by the four *no-reference* metrics, and similarity to the original using SSIM, across the five experiments presented. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

(a) (b)

Figure 4.22: (a) Original image. | (b) GP output with depth 5 and using the arithmetic average of the *NIMA* technical model and *NIMA* aesthetic as fitness.

(a) (b)

Figure 4.23: (a) Original image containing artifacts. | (b) Highlighted artifacts due to image over-enhancement.

	Fit.	*Div.*	*NIMA A.*	*NIMA T.*	*BRISQUE*	*PHIL*	*SSIM*	*MSE*
μ	$N.A.^2 \times SSIM$	No	0,75	-0,28	-0,48	0,30	0,83	2707
σ	$N.A.^2 \times SSIM$	No	0,35	0,31	0,97	0,77	0,05	959
μ	$N.T.^2 \times SSIM$	No	0,073	0,01	0,39	0,13	0,98	38
σ	$N.T.^2 \times SSIM$	No	0,08	0,09	0,59	0,60	0,02	241
μ	$N.A.^2 \times SSIM$	Yes	0,57	-0,21	-0,52	0,29	0,88	1451
σ	$N.A.^2 \times SSIM$	Yes	0,38	0,29	1,06	0,71	0,05	818
μ	$N.T.^2 \times SSIM$	Yes	0,08	0,02	0,45	0,13	0,98	45
σ	$N.T.^2 \times SSIM$	Yes	0,09	0,09	0,69	0,60	0,01	63

Table 4.10: Average (μ) and standard deviation (σ) of the improvement and similarity using four GP configurations on the test dataset. *"Fit."* indicates the used fitness function where *N.A.* and *N.T.* mean *NIMA Aesthetic* and *NIMA Technical* respectively. *"Div"* indicates that dataset division was used. The highlighted cells are the ones where the score is measured by the model used in the fitness function. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

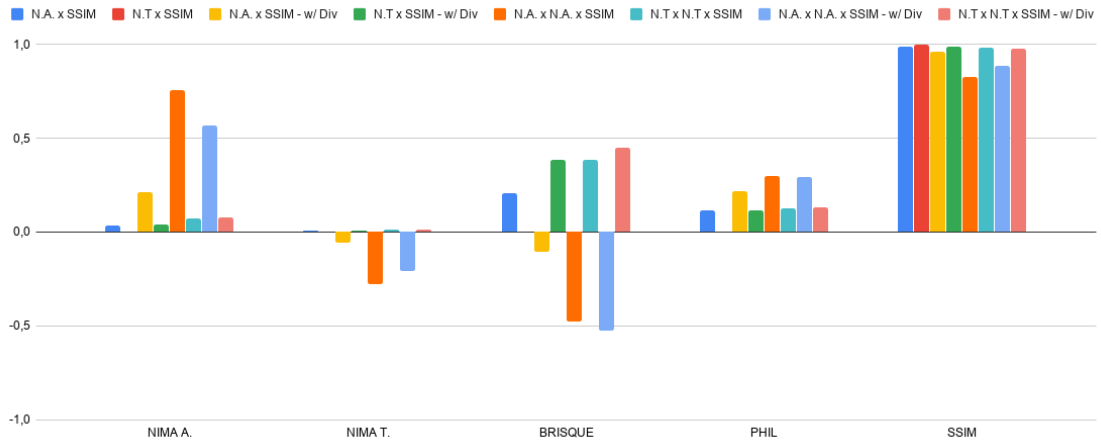


Figure 4.24: Graphical comparison of the improvement obtained by the four *no-reference* metrics, and similarity to the original using SSIM, across the eight experiments presented. *N.A* and *N.T* refer to *NIMA* aesthetic and technical model respectively, and *Div* signifies the use of dataset division. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

4.7 Machine Learning in Image Enhancement

We sought to explore machine learning methods and incorporate them in our pipeline, where a trained machine learning model will be used in the beginning of the pipeline online component. In this section we will present the work done to achieve this end, as well as the results obtained.

As we saw in Chapter 2 and briefly discussed in Chapter 3, GANs are most of the time the "go to" architecture for current state-of-the-art image processing. Therefore, we explored the possibility of applying this same architecture to our problem. GANs are a machine learning layout proposed by *I. Goodfellow* et al. in 2014 [25], that takes advantage of the concept of adversarial learning to create generative models. Its overall architecture is explained briefly in Chapter 2. During this work we focused on experimenting with an iteration of this architecture proposed recently by *P. Isola* et al. in [48] referred as *Pix2Pix GAN* and is introduced as a *"general-purpose solution to image-to-image translation problems"*. One main change it provided over the typical GAN is that, instead of learning to map noise input to an output that goes accordingly to the training dataset, it learns to map an input image to a variance of that input image based on the training dataset. This is called a conditional GAN or *cGAN*. Furthermore, *pix2pix* suggests the training of the generatior via not only by adversarial loss, but also by direct comparison with the target image using *L1* loss. Both losses in conjunction encourage the generator to produce images that are in the doming of the target set but also that are plausible translations for the input original image. Because of this, *pix2pix* GAN seemed like a good solution to be applied to the IE field. We explored this hypothesis by implementing and testing a *pix2pix GAN* according to the original paper specifications.

Pix2pix GAN is considered a supervised machine learning model, and as such it requires labeled datasets during training. In our scenario, a label is the enhanced version of an input image, that is considered as target. We made use of two of datasets presented in Section 4.2, those being the provided dataset and the *FiveK* dataset. For the provided dataset we used the first set of 436 images and had to use the also provided paired versions, enhanced by *One-Click*, as the ground truth. For *FiveK* dataset we used the first thousand images and here, as opposed to the provided dataset, we had 5 distinct possible targets to choose from. To make that decision we ran our six available IQA tool on all the five experts targets and selected the one that performed best according to them. Figure 4.11 shows the improvement score for each expert. We selected the expert E as it was the one that presented better scores overall.

4.7.1 Results

We trained the implemented GAN using both mentioned datasets with the respective targets, and ran the model on our test set to document each model performance. Note that, following the original implementation, each image was resized to 256 by 256. Table 4.12 shows the results for both trained models.

We selected the model that showed the best performance in our metrics and performed all the subsequent tests using it. The first test we ran using this model was a simple one in which we compared the performance of the manually selected set of classical functions introduced in Section 4.6, over the original images and over the GAN output. The improvements of the GAN were extremely soft and demonstrated to be cumulative with the values in Table 4.5. Table 4.13 shows these results and Figure 4.25 showcases a couple of examples of the GAN output.

		NIMA A.	NIMA T.	BRISQUE	PHIL	SSIM	MSE
μ	**Expert A**	0,06	-0,08	-0,11	-0,17	0,72	4795
σ	**Expert A**	0,21	0,24	0,40	0,59	0,14	6006
μ	**Expert B**	0,05	-0,08	-0,05	-0,15	0,78	2631
σ	**Expert B**	0,18	0,22	0,28	0,46	0,13	2943
μ	**Expert C**	0,08	-0,03	0,01	0,03	0,76	3068
σ	**Expert C**	0,21	0,23	0,29	0,51	0,13	3426
μ	**Expert D**	0,09	-0,07	-0,04	0,01	0,75	3774
σ	**Expert D**	0,21	0,23	0,34	0,51	0,13	3831
μ	**Expert E**	0,09	-0,05	0,013	0,10	0,74	3688
σ	**Expert E**	0,21	0,24	0,36	0,54	0,13	35

Table 4.11: Average (μ) and standard deviation (σ) of the improvement and similarity for each *FiveK* expert and IQA tool. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

		NIMA A.	NIMA T.	BRISQUE	PHIL	SSIM	MSE
μ	**Model A**	0,08	-0,04	-0,22	0,15	0,92	677
σ	**Model A**	0,16	0,16	0,67	0,30	0,05	317
μ	**Model B**	-0,14	-0,14	-0,37	-0,05	0,90	941
σ	**Model B**	0,17	0,18	0,92	0,66	0,05	538

Table 4.12: Average (μ) and standard deviation (σ) of the improvement and similarity for each trained GAN model, where A and B represent the model trained with the provided dataset and the *FiveK* dataset, respectively. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

	NIMA A.	NIMA T.	BRISQUE	PHIL	SSIM	MSE
μ	0,51	-0,10	0,13	0,38	0,61	4577
σ	0,36	0,27	1,52	0,69	0,08	2368

Table 4.13: Average (μ) and standard deviation (σ) of the improvement of the GAN model A plus a set of classical functions. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

(a) (b) (c) (d)

Figure 4.25: From left to right: Original image; Output using solution from A; Output using solution from B; Output using solution from C;

4.8 Image Enhancement Pipeline

As already mentioned in Chapter 3, it is proposed the creation of an IE pipeline, containing two image processing phases, those being a *pix2pix* GAN and a set of classical filters evolved using GP, and also two decision components, that aim to alter the pipeline's output, based on the input characteristics. In this section, we will go through both of those decision components, explaining their motivation and details. Then we will present the final results along with the details and results of a performed user survey about the topic.

4.8.1 Decision Components

Another conclusion was extrapolated from the results of all the previous tests. After extracting supplementary metrics from them, those being: "original score where improvement was positive" and "original score where improvement was negative", both with respective averages and standard deviation. These values showed us an intuitive, yet interesting conclusion. In almost every single experiment, the average original score where there was improvement, was lower than where there was not an improvement. This demonstrates that, according the available metrics, its much easier the improve upon a bad image than it is for a good one. This is a very rational conclusion, since better images have overall "less room for improvement", making the enhancement operation harder and most importantly, degrading the images quality. The standard deviation shows very similar values in both measurements, averaging values of *0.45* for both *NIMA* models and *1.3* for *PhotoILike*. This is expected as *PhotoILike* has in consideration features beyond visual aspect.

Having this in mind we propose the introduction of a *threshold filter* that filters images whose quality presumably can not be improved without degrading the original quality, from images that could benefit from improvement attempts. To do this experiment, we decided to use the metric that presented the least standard deviation in the original images score where there was no improvement, which consistently proved to be the technical model. Using this metric also makes sense because, based on the previous results, it responds negatively to over-enhancement.

This experiment proves that a threshold in automatic image enhancement can be beneficial. We are also aware that a more dynamic solution under the same ideology is possible, however it would require extra complexity, making it out the scope of this work.

An extra step was added to the pipeline, where, in an intuitively manner, a comparison between each iteration of the input image was made according to a desired metric. This way, we guarantee the best output possible out of the chosen metric. In our case we selected *PhotoILike* as the IQA tool to make the decision, as it is the closest to our use case scenario.

4.8.2 Final results

With the purpose of evaluating our pipeline's overall performance, the GAN model that presented the best performance was selected. Furthermore, a selection process for the best GP configuration from the experiments, was carried out.

We concluded that, in order to assess the best solution from the GP algorithm, we could not rely on individual scores from the IQA tools. Instead, and following a similar line of thought as of when the iterations of the fitness function, we created a function that aims

to obtain a final score based on the balance of multiple IQA scores. The proposed function can be defined as

$$\frac{(N.A. \times SSIM) + (N.T. \times SSIM) + (PHIL \times SSIM)}{3}$$

where *N.A.* and *N.T.* mean the average aesthetic and technical score respectively, *PHIL* the average *PhotoILike* score and SSIM the average SSIM score.

Using this process we selected the three best configurations. They were:

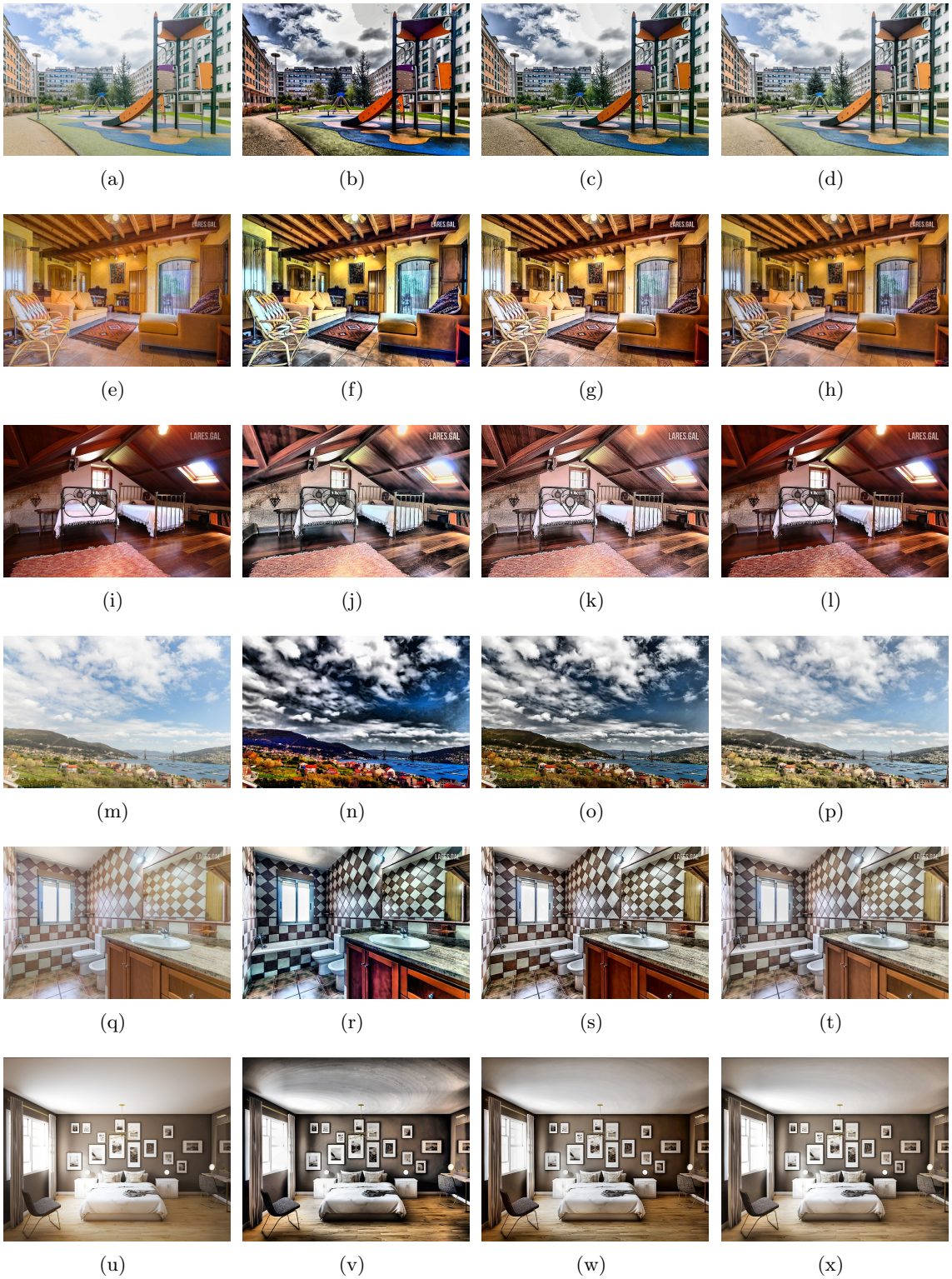
- **A -** Technical model as fitness function with a tree depth of 10.
- **B -** Average of both technical and aesthetic model as fitness function and a tree depth of 5.
- **C -** Aesthetic model squared times the SSIM as fitness function and a tree depth of 10.

For each selected method we repeated the original experiment 10 times in order to obtain statistical validation of the stochastic GP method. Then the best solution from each group of 10 experiments was selected using the same formula. Table 4.14 contains the results of the 3 chosen solutions. It is noteworthy the fact that the same GP configuration converged to very similar solutions regarding the used filters and generally only changing the parameterization and the order of occurrence. We are aware that the number of experiments should ideally be higher, however we had to compromise due to time constrains.

	GP	***NIMA A.***	***NIMA T.***	***BRISQUE***	***PHIL***	***SSIM***	***MSE***
μ	***A***	0,97	-0,13	0,63	0,44	0,58	87
σ	***A***	0,40	0,35	1,43	0,85	0,08	4120
μ	***B***	0,88	-0,21	0,20	0,51	0,69	5574
σ	***B***	0,41	0,30	1,51	0,79	0,11	3738
μ	***C***	0,58	-0,17	-0,53	0,37	0,88	1349
σ	***C***	0,30	0,24	1,03	0,69	0,04	591

Table 4.14: Average (μ) and standard deviation (σ) of the improvement and similarity using the best 3 GP configurations, on the test dataset. *"GP"* indicates the configuration used labeled according to the list presented in Section 4.8.2. Highlighted are the cells that presented the best average score in each metric. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

The extracted metrics scores are positive, however there is a clear dissimilarity between the three solutions. Solution *A* appears to be much more destructive followed by *B* and finally *C*. We can observe this by examining the aesthetic and technical scores along with the SSIM, which for configuration *A* is a bit lower than our determined range. Observing the output images, we can confirm this statement. Figure 4.26 presents two image examples for each solution. These results also corroborate the statement that the *PhotoILike* metric is propitious to emphasize the score based on aesthetic enhancement.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

(u) (v) (w) (x)

Figure 4.26: From left to right: Original image; Output using solution from A; Output using solution from B; Output using solution from C;

Employing the best 3 solutions from the GP, we performed some tests using both pipeline decision components. We used the compare component and the threshold component both separately and in conjunction, and recorded the metrics score along with the number of time the output images were equal to the original, only processed by the *GAN*, only processed by the classical functions or processed by both.

In order to get a value to use as a threshold, we randomly divided the test set in two equality populated image sets of 604 images. We applied to one of the subsets, the classical methods proposed by each pipeline, obtaining threshold value using, as explained in 4.8.1, the average original score for images where improvement was negative, using the *NIMA* technical model.

Table 4.15, 4.16 and 4.17 present the results for each GP solution *A*, *B* and *C*, respectively. For simplicity of presentation we only show the average score of each metric. For comparison purposes, we highlighted in green every cell that scored better than *One-Click*, whose results are presented in Table 4.5. The performance of the *PhotoILike* metric is especially important to us, as this tool was made exclusively for our use case scenario. Table 4.18 shows the distribution of outputs for every scenario. Figure 4.27 shows a visual comparison between the output of each pipeline with the original dataset. As it is observable, using both decision components altered the pipeline output making the *PhotoILike* and aesthetic values decrease but improving the technical model and SSIM scores. This means a more balanced output that better prevents unnecessary enhancement.

Components	***NIMA A.***	***NIMA T.***	***BRISQUE***	***PHIL***	***SSIM***	***MSE***
Nothing	1,06	-0,24	0,62	0,53	0,54	9611
Both	0,44	-0,11	0,08	0,47	0,79	3912
Compare	0,72	-0,17	0,30	0,70	0,68	6221
Threshold	0,62	-0,15	0,18	0,35	0,70	5869

Table 4.15: The values represent the average improvement of each pipeline configuration, using GP solution *A*. All the highlighted cells represent values in which the performance was better than *One-Click*. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

Components	***NIMA A.***	***NIMA T.***	***BRISQUE***	***PHIL***	***SSIM***	***MSE***
Nothing	0,92	-0,31	0,50	0,50	0,65	6119
Both	0,40	-0,15	0,03	0,46	0,83	26
Compare	0,67	-0,23	0,21	0,69	0,74	4145
Threshold	0,54	-0,19	0,10	0,33	0,76	3806

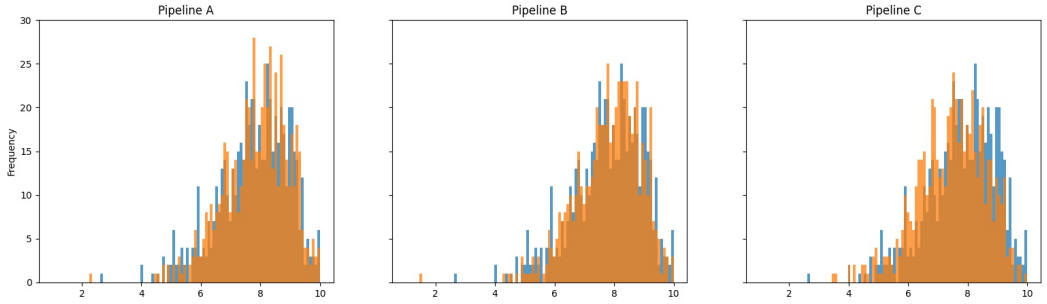
Table 4.16: The values represent the average improvement of each pipeline configuration, using GP solution *B*. All the highlted cells represent values in which the performance was better than *One-Click*. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

Components	*NIMA A.*	*NIMA T.*	*BRISQUE*	*PHIL*	*SSIM*	*MSE*
Nothing	0,69	-0,27	-0,36	0,43	0,82	23
Both	0,31	-0,14	-0,32	0,41	0,90	1135
Compare	0,49	-0,20	-0,37	0,60	0,87	16
Threshold	0,41	-0,17	-0,34	0,28	0,86	16

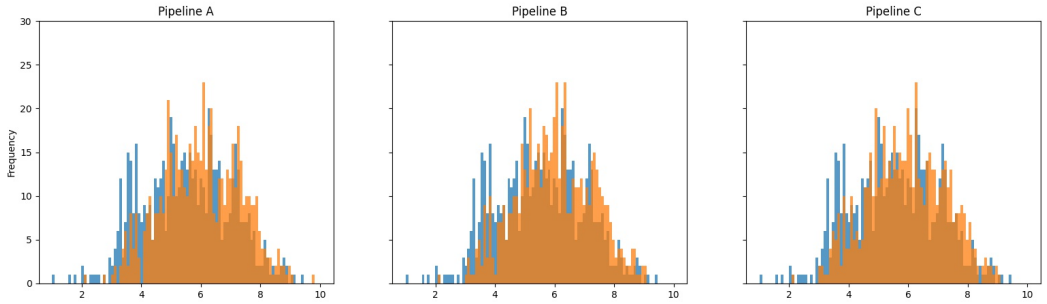
Table 4.17: The values represent the average improvement of each pipeline configuration, using GP solution *C*. All the highlighted cells represent values in which the performance was better than *One-Click*. All the *no-reference* metrics values range from 1 to 10 where 1 means the lowest quality and 10 highest quality, and the improvement is calculated by subtracting the original score from the the resulting one.

	A			***B***			***C***		
	Both	Comp.	Thresh.	Both	Comp.	Thresh.	Both	Comp.	Thresh.
Input (%)	18,71	9,77	0,00	19,21	9,27	0,00	19,54	10,26	0,00
ML (%)	43,05	23,18	43,38	42,55	20,53	44,87	42,72	21,69	45,70
Classic (%)	11,42	20,36	0,00	10,93	20,86	0,00	10,60	19,87	0,00
Both (%)	26,82	46,69	56,62	27,32	49,34	55,13	27,15	48,18	54,30

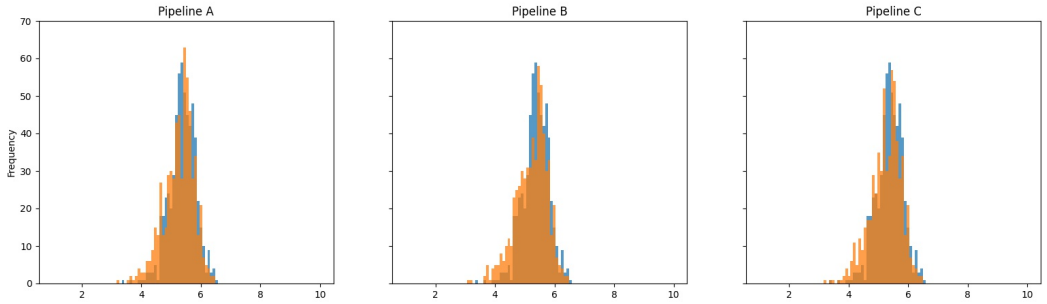
Table 4.18: Distribution of the pipeline outputs in percentage, for each pipeline configuration, and for each GP solution, *A*, *B* and *C*. *Comp.* means that only comparison phase was used, *Tresh.* means only the treshold was used, *Both* meas both were used.



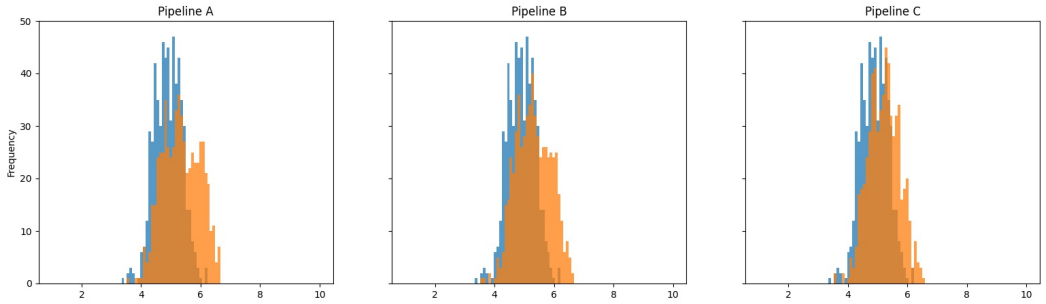
(a)



(b)



(c)



(d)

Figure 4.27: Graphical comparison between the original test sub-dataset (blue) and the results from the pipeline using both decision components (orange), computed by all four *no-reference* IQA tools. From top to bottum, the used metric is *BRISQUE*, *PhotoILike*, *NIMA* Technical and *NIMA* Aesthetic.

4.8.3 User Study and Result Analysis

With the purpose of performing additional validation of our results, we developed a online based user study so that we could get a real world appreciation of our obtained results and compare it the the IQA results. To carry out this study, we randomly selected twelve images from each GP solution. The twelve images were selected in groups of four, where each group was considered bad, medium or good, by the *PhotoILike* IQA tool, meaning they were in the bottom, middle or top 33.(3)% of the distribution. These selection method was used so that the sample of images used for the survey would be as representative of the real set of images as possible. These selected images were processed by a pipeline using both compare and threshold modules, where the output was either from the classical filters or from the *GAN* followed by the classical filters. So, the survey was made out of 36 questions where the respondents were simply asked to select out of 3 versions of one image, the one that appeared to them as the most appealing photo in a online real-estate marketing context. The versions of each images were: the original, the outputted by *One-Click* and the outputted by our pipeline. We ran a few test questions so that we could have a rough idea of how much time the choosing process would take, and determined it was around 10 seconds. That would make the questionnaire time length of around 6 minutes, plus the reading of the introductory text, which according to the study in [92], is an acceptable duration for web survey. We distributed the online survey by everyone who was willing to participate and collected 60 responses. A copy of the performed user test is available at `https://tinyurl.com/iqa-user-test`. Figure 4.28 shows an example of three sets of images used in the survey, all from the middle set 33.(3)% of the distribution.

Looking at the *PhotoILike* results in the previous Section 4.8.2, we can expect the pipeline output to often dominate the user answers. Even looking for the remaining metrics results, only the technical model consistently performed worse than *One-Click*, even if it also presented a negative improvement in said metric. Other interesting analysis to be made is to look at the correlation between the multiple metrics. In Table 4.19 a correlation matrix is presented, using values from all the experiments available. Note that, as already observed, *PhotoILike* and *NIMA* aesthetic model has high correction, whereas SSIM has a high correlation to the *NIMA* technical model. Also note that the aesthetic and technical model provide very disassociate scores.

	NIMA A.	***NIMA T.***	***BRISQUE***	***PHIL***	***SSIM***
NIMA A.	1,0000	-0,9292	0,0820	0,8055	-0,8536
NIMA T.	-0,9292	1,0000	0,0972	-0,6649	0,7243
BRISQUE	0,0820	0,0972	1,0000	-0,1007	-0,0679
PHIL	0,8055	-0,6649	-0,1007	1,0000	-0,8580
SSIM	-0,8536	0,7243	-0,0679	-0,8580	1,0000

Table 4.19: Correlation matrix of all IQA tools. Higher values indicate higher proportionality correlation and lower values indicate higher inverse proportionality correlation. Zero indicates there is no correlation between variables.

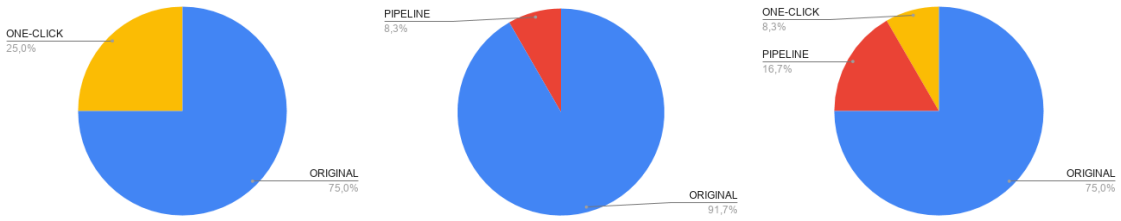
In order to better understand the user test outcome, we performed a few analyses. Firstly, we calculated the *mode* of each question, meaning the option that was selected most of the times. Then we did the same thing, but excluding options where the original image was selected. This gave us a better perception of what option was the second and third most selected when the original was the first, allowing us to compare *One-Click* and our solution with mode detail. We then divided the questions in groups of three groups of twelve, one containing each GP solution and performed the same data analysis for each one individually. The results are graphically represented in Figures 4.29, 4.30 and 4.31

(a) (b) (c)

(d) (e) (f)

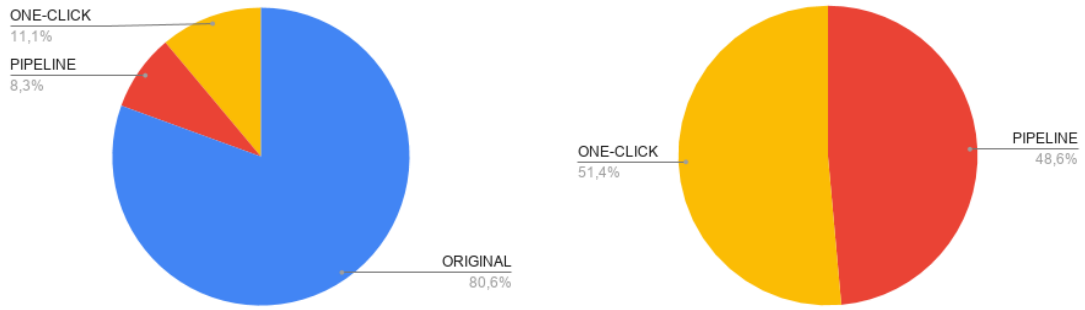
(g) (h) (i)

Figure 4.28: From left to right: Original image | Our output (A, B and C from top to bottom) | *One-Click* output.



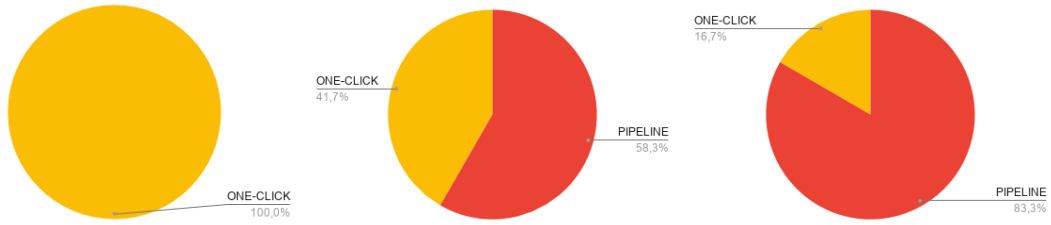
(a) (b) (c)

Figure 4.30: Mode distribution using each 12 questions groups, divided by GP configuration. From left to right: configuration A, B and C



(a) (b)

Figure 4.29: (a) Mode distribution using all 36 questions. | (b) Mode distribution using all 36 questions, but excluding answers where the original image was selected.



(a) (b) (c)

Figure 4.31: Mode distribution using 12 questions groups, divided by GP configuration. From left to right: configuration A, B and C

These results show something very important for our use case and pipeline. The fact that a great majority of the users selected the original image in most questions indicates that there is indeed the need of a threshold in an automatic IE pipeline. However our approach was not enough, since many images that passed the threshold were fewer times choosed by the used, when comparing to the original version, and so it could be further improved. A deeper analysis shows that similarity and content fidelity plays and important role in the user choice. In Figure 4.30 and 4.31 we can see that, pipeline using GP A performed very poorly when compared to B and C, which denounced a bigger similarity with the original image, even when A was the one with the best *PhotoILike* and *NIMA* aesthetic score. We can consider the results from GP C to be positive, as it performed considerably better than *One-Click* according to the users.

The user test, as we suggested previously, presented unexpected results when comparing to our metrics predictions. Having as objective the understanding of why this happened, we "simulated" 5 more answers to our questionnaire, one for each *no-reference metric* and one for SSIM, where it could only choose between *One-Click* and our output, and always choosing the one with the highest similarity to the original. With this information, we also calculated the number of times each metric agreed with the average user choice. We divided the findings in the 3 previous considered groups of 12 images. Important to note that in this analysis, SSIM also excludes the original image as an option and compares its answer with the one that was most voted between *One-Click* and ours. Figure 4.32 and Table 4.20 present the results of both experiments, respectively.

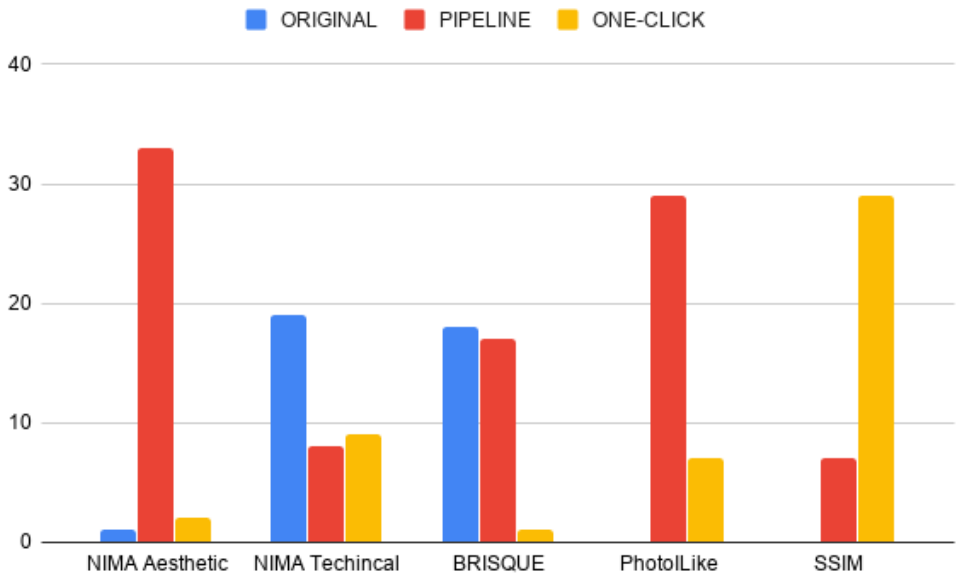


Figure 4.32: Graphical representation of each metrics response to the 36 questions. Note that SSIM only shows 2 bins as it could only chose between *One-Click* and ours. This is not true for *PhotoILike*.

	Group A	***Group B***	***Group C***	***TOTAL***
NIMA Aesthetic	0	0	3	3
NIMA Techincal	6	6	6	18
BRISQUE	4	3	4	11
PhotoILike	0	1	3	4
SSIM	12	6	4	22

Table 4.20: Number of times each metric *"agreed"* with the average user vote. Note that SSIM excludes the original image as an option and compares its answer with the one that was most voted between *One-Click* and ours.

Based on our observations of the presented results, we can extrapolate a few conclusions. Firstly, it is obvious that *NIMA* aesthetic model and *PhotoILike*, a tool said to evaluate real-estate images based on their marketing appeal, were the most divergent from the user survey results, as both gave much higher scores to our solution. This can be due to an exploit in the classifier model, that makes it inclined towards giving a good classification to over-enhanced images, which generally does not reflect the user opinion. Secondly, the *NIMA* models, especially the aesthetic, are logically much more favorable to our solution, as we aimed to optimize it for them. Moreover, looking at Table 4.20, we can deduce that SSIM has a great impact on peoples decisions, as it was the one that better predicted the results. However, this accuracy decreases as we go from group A to group B and from group B to group C. This is where our solution produced much more similar results to the original, with SSIM scores closer to the ones of *One-Click*, while still increasing the aesthetic slightly, but not excessively. This also goes according to the scores obtained when evaluating images from the *FiveK* dataset improved by human experts, as they also promoted a high similarity to the original and a very slight increase in aesthetic score. Additional exploration of the survey data also revealed that, in our few data samples, images in the *good* category, meaning that they already have a considerably good quality, were more often preferred to be enhanced with *One-Click*, which makes sense as *One-Click*

alters less the image and has less chance of deteriorating the original quality. On the other hand, *"bad"* images were more often preferred to be enhanced with a *stronger* enhancement of the aesthetic quality. All of this reinforces the claim that, at least in our case study, it is much more reliable to search for a balance between similarity metrics and aesthetic metrics. Interestingly enough, a relatively straightforward IQA tool like *BRISQUE*, that did not show particularly big correlation with another metric, also did a good prediction for users response. We did not collected nor study any demographic data as it was out of scope, however, this study could have gone more in depth in that direction.

Chapter 5

Conclusions and Future Work

In this thesis, we defined image enhancement as an image processing procedure in which an image becomes better suited for a task, and presented useful applications for such techniques. We discussed different types of possible image improvements as well as different architectures and methods that aim to achieve them. We ran through classical, evolutionary and machine learning based approaches and briefly analyzed how they worked. We also gave context around the complex problem of image quality assessment and explained that it is a difficult task due to the problem intrinsic subjectiveness.

As seen in the state-of-the-art section, there are different classes of methods that measure image quality. We perform tests using state-of-the-art *no-reference* methods, and one provided black-box *no-reference* Image Quality Assessment (IQA) that aims to qualify an input image based on its appeal for online real-estate marketing, which was envisaged as our case study. We also made use and one *full-reference* method, Structural similarity (SSIM), as a measure of over-enhancement.

We performed a broad analysis and proposed the research, development and study of an unsupervised end-to-end Image Enhancement (IE) pipeline that takes advantage of multiple techniques in order to output the best achieved result. This was done while having our *no-reference* metrics as guidance for every training and evolution. Despite that, seeking further results validation, we conducted a user test using our top 3 solutions based on the IQA metrics scores.

The results were unexpected as they deviated from most of our metrics. We found that, in our use case scenario, the user choice when selecting between an original image and two possible enhancements of that image, is greatly influenced by the similarity to the original version, much more than it is the aesthetic quality of the images. This means that preservation of the original contextual information is a highly demanded feature, and should always be regarded when aiming for a similar objective as ours. This is why most commercial applications of automatic IE, focus purely on simpler and less destructive operations, such as color balancing and brightness adjustment, as it is easier and safer.

Additionally, we argue the importance of a threshold in IE solutions, as the results repeatedly showed that every image has an inherent maximum quality that attempting to improve any further, will most of the times cause quality degradation. This came along with the conclusion that images whose original quality is lower, are generally easier to improve upon, than those that demonstrate higher quality right away.

The provided IQA model produced one of the most far off predictions, even when specifically tailored for this use. This can be due to model exploits that consistently benefited

high distortions in favor of unrealistic increases in the aesthetic aspect the image. However, these exploits can also be utilized and manipulated to create surreal versions of an input image. A proof of concept of this possibility is achieved in this work when we tried to enhance an image purely based on the aesthetic model score, creating very interesting variations of the original images, even if not suited for our particular scenario.

The executed experimentation confirms the research done, as both indicate that enhancing multiple features of an image simultaneously with considerable improvements is a difficult problem. Also, the user test results diverge from our metrics predictions, suggesting that the problem of automatic IE is closely tied with the equally difficult problem of automatic IQA.

Based on the user test, our best Genetic Programing (GP) solution showed the highest similarity to the original image and had a choice rate against *One-Click* of 83.3%; but even then, the original image had an overall choice rate of 75%. Nonetheless, our work constitutes a step forward in the automatic image enhancement field in real-world applications as it explores new possibilities and endorses a few interesting conclusions, namely the necessity to heed similarity when approaching the image enhancement problem in a similar context.

However, our research can be further explored in a number of meaningful and interesting ways. IE is a extensive field, and so it was virtually impossible to expand upon every detail of this work. The core of our solution relies on seven different elementary classical functions. This set of functions can be changed in order to manipulate the obtained results. The parameters range for each function is also manually defined, and so, it could also be changed to allow the generated solutions to have more or less impact in the image. All this implies the possibility of the creation of a similar system where enhancement intensity is a controllable parameter. Furthermore, improving image clustering may offer a significant advantage to this architecture.

Bibliography

[1] Chervinskii, "Schematic picture of an autoencoder architecture," Dec 2015.

[2] B. English Wikipedia, "Genetic program tree," Sep 2007.

[3] S. E. Umbaugh, *Computer Vision and Image Processing: A Practical Approach Using Cviptools with Cdrom*. USA: Prentice Hall PTR, 1st ed., 1997.

[4] S. Rana, "A review of medical image enhancement techniques for image processing," *International Journal of Current Engineering and Technology*, vol. 5, pp. 1282–1286, 01 2011.

[5] G. Anbarjafari, H. Tasmaz, R. Kiefer, and C. Ozcinar, "Satellite image enhancement: Systematic approach for denoising and resolution enhancement," *DYNA Ingenieria e Industria*, vol. 90, pp. 326–329, 01 2016.

[6] P. Schuch, S. Schulz, and C. Busch, "Survey on the impact of fingerprint image enhancement," *IET Biometrics*, vol. 7, pp. 102–115(13), March 2018.

[7] W. Robert, "Analog-to-digital converter survey and analysis," *IEEE*, 1999.

[8] L. He, F. Gao, W. Hou, and L. Hao, "Objective image quality assessment: a survey," *International Journal of Computer Mathematics*, vol. 91, no. 11, pp. 2374–2388, 2014.

[9] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, pp. 213–227, Mar 2016.

[10] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. IV–3313–IV–3316, May 2002.

[11] Zhou Wang, A. C. Bovik, and B. L. Evan, "Blind measurement of blocking artifacts in images," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 3, pp. 981–984 vol.3, Sep. 2000.

[12] R. Ferzli and L. J. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb)," *IEEE Transactions on Image Processing*, vol. 18, pp. 717–728, April 2009.

[13] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal Processing Letters*, vol. 17, pp. 513–516, May 2010.

[14] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, pp. 3350–3364, Dec 2011.

[15] H. Tang, N. Joshi, and A. Kapoor, "Learning a blind measure of perceptual image quality," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 305–312, June 2011.

[16] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the dct domain," *IEEE Transactions on Image Processing*, vol. 21, pp. 3339–3352, Aug 2012.

[17] R. Maini and H. Aggarwal, "A comprehensive review of image enhancement techniques," 2010.

[18] G. Singh and A. Mittal, "Various Image Enhancement Techniques- A Critical Review," *International Journal of Innovation and Scientific Research*, vol. 10, no. 2, pp. 267–274, 2014.

[19] Z. Shi, M. Zhu, B. Guo, M. Zhao, and C. Zhang, "Nighttime low illumination image enhancement with single image using bright/dark channel prior," *EURASIP Journal on Image and Video Processing*, vol. 2018, 12 2018.

[20] A. M. Dhanalakshmi and R. Baskar, "Image enhancement technique- a review," *International Journal of Pharmacy and Technology*, vol. 8, no. 4, pp. 20927–20936, 2016.

[21] P. Suganya, S. Gayathri, and N. Mohanapriya, "Survey on Image Enhancement Techniques," *International Journal of Computer Applications Technology and Research*, no. September 2013, pp. 623–627, 2013.

[22] I. A. Reshi, "New Techniques Used for Image Enhancement," *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, vol. 7, no. 6, pp. 18–22, 2017.

[23] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[24] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and Others, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *Cvpr*, vol. 2, no. 3, p. 4, 2017.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[26] Y. Deng, C. C. Loy, and X. Tang, "Aesthetic-driven image enhancement by adversarial learning," *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pp. 870–878, 2018.

[27] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Transactions on Multimedia*, vol. 21, pp. 2131–2145, Aug 2019.

[28] T. Pouli, E. Reinhand, and D. Cunningham, "The human visual system," in *Encyclopedia of Image Processing* (P. A. Laplante, ed.), CRC Press, 2018.

[29] L. K. C. Deepti Ghadiyaram, Todd Goodall and A. C. Bovik, "Perceptual image enhancement," in *Encyclopedia of Image Processing* (P. A. Laplante, ed.), CRC Press, 2018.

[30] W. Wang, Z. Chen, X. Yuan, and X. Wu, "Adaptive image enhancement method for correcting low-illumination images," *Information Sciences*, vol. 496, pp. 25–41, 2019.

[31] C. Y. Wong, G. Jiang, M. A. Rahman, S. Liu, S. C. F. Lin, N. Kwok, H. Shi, Y. H. Yu, and T. Wu, "Histogram equalization and optimal profile compression based approach for colour image enhancement," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 802–813, 2016.

[32] H. Talebi and P. Milanfar, "Fast multi-layer laplacian enhancement," *IEEE Transactions on Computational Imaging*, 06 2016.

[33] S. Zhuo, X. Zhang, X. Miao, and T. Sim, "Enhancing low light images using near infrared flash images," *Proceedings - International Conference on Image Processing, ICIP*, pp. 2537–2540, 2010.

[34] S. B. Kang, A. Kapoor, and D. Lischinski, "Personalization of image enhancement," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1799–1806, 2010.

[35] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[36] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," *ACM Trans. Graph.*, vol. 25, p. 624–630, July 2006.

[37] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "Enlightengan: Deep light enhancement without paired supervision," *arXiv preprint arXiv:1906.06972*, 2019.

[38] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to See in the Dark," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3291–3300, 2018.

[39] M. Afifi, A. Punnappurath, A. Abdelhamed, H. C. Karaimer, A. Abuolaim, , and M. S. Brown, "Color temperature tuning: Allowing accurate post-capture white-balance editing," in *Color Imaging Conference (CIC)*, Society for Imaging Science and Technology, 2019.

[40] M. Afifi, B. Price, S. Cohen, and M. S. Brown, "When color constancy goes wrong: Correcting improperly white-balanced images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1535–1544, 2019.

[41] M. Afifi and M. S. Brown, "What else can fool deep learning? addressing color constancy errors on deep neural network performance," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[42] M. Afifi and M. S. Brown, "Deep white-balance editing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[43] M. Afifi and M. S. Brown, "Sensor-independent illumination estimation for dnn models," in *British Machine Vision Conference (BMVC)*, 2019.

[44] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[45] L. Rundo, A. Tangherloni, M. Nobile, C. Militello, D. Besozzi, G. Mauri, and P. Cazzaniga, "Medga: A novel evolutionary method for image enhancement in medical imaging systems," *Expert Systems with Applications*, vol. 119, 11 2018.

[46] C. Munteanu and A. Rosa, "Evolutionary image enhancement with user behaviour modeling," *ACM SIGAPP Applied Computing Review*, vol. 9, 12 2000.

[47] S. Rajput, K. V. Arya, and V. Bohat, *Face Image Super-Resolution Using Differential Evolutionary Algorithm*, pp. 635–644. Springer, 09 2019.

[48] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[49] H. T. Esfandarani and P. Milanfar, "NIMA: neural image assessment," *CoRR*, vol. abs/1709.05424, 2017.

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

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

[52] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[53] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "Tid2008 - a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, pp. 30–45, 01 2009.

[54] J. Lim, M. Heo, C. Lee, and C.-S. Kim, "Contrast enhancement of noisy low-light images based on structure-texture-noise decomposition," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 107 – 121, 2017.

[55] Y. Peng and P. C. Cosman, "Underwater image restoration based on image blurriness and light absorption," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1579–1594, 2017.

[56] Y. Kinoshita and H. Kiya, "Convolutional neural networks considering local and global features for image enhancement," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2110–2114, 2019.

[57] S. Wong, Y. Yu, N. A. Ho, and R. Paramesran, "Comparative analysis of underwater image enhancement methods in different color spaces," in *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 034–038, 2014.

[58] G. Wang, L. Li, Q. Li, K. Gu, Z. Lu, and J. Qian, "Perceptual evaluation of single-image super-resolution reconstruction," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3145–3149, 2017.

[59] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Student and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, 2004.

[60] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input / output image pairs," in *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[61] S. S. Al-amri, N. V. Kalyankar, and S. D. Khamitkar, "Linear and Non-linear Contrast Enhancement Image," *Journal of Computer Science*, vol. 10, no. 2, pp. 139–143, 2010.

[62] S. Bazeille, I. Quidu, L. Jaulin, and J.-P. Malkasse, "Automatic underwater image pre-processing," *Proceedings of CMM'06*, 10 2006.

[63] Yu Wang, Qian Chen, and Baeomin Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 68–75, Feb 1999.

[64] Y. Xie, L. Ning, M. Wang, and C. Li, "Image enhancement based on histogram equalization," *Journal of Physics: Conference Series*, vol. 1314, p. 012161, oct 2019.

[65] Komal Vij and Yaduvir Singh, "Enhancement of Images Using Histogram Processing Techniques," *Int. J. Comp. Tech. Appl.,*, vol. Vol 2, no. 2, pp. 309–313, 2011.

[66] Y. Chang, C. Jung, P. Ke, H. Song, and J. Hwang, "Automatic Contrast-Limited Adaptive Histogram Equalization with Dual Gamma Correction," *IEEE Access*, vol. 6, pp. 11782–11792, 2018.

[67] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 38, no. 1, pp. 35–44, 2004.

[68] M. S. Hitam, E. A. Awalludin, W. N. Jawahir Hj Wan Yussof, and Z. Bachok, "Mixture contrast limited adaptive histogram equalization for underwater image enhancement," in *2013 International Conference on Computer Applications Technology (ICCAT)*, pp. 1–5, Jan 2013.

[69] S. C. Huang, F. C. Cheng, and Y. S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1032–1041, 2013.

[70] H. Farid, "Blind inverse gamma correction," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001.

[71] A. Buades, B. Coll, and J.-M. Morel, "Non-Local Means Denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.

[72] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Transactions on Image Processing*, vol. 9, pp. 505–510, March 2000.

[73] N. Limare, J.-L. Lisani, J.-M. Morel, A.-B. Petro, and C. Sbert, "Simplest color balance," *Image Processing On Line*, vol. 1, 10 2011.

[74] J. Immerkær, "Fast noise variance estimation," *Comput. Vis. Image Underst.*, vol. 64, p. 300–302, Sept. 1996.

[75] E. Peli, "Contrast in complex images," *J. Opt. Soc. Am. A*, vol. 7, pp. 2032–2040, Oct 1990.

[76] D. Rex Finley, "Hsp color model - alternative to hsv (hsb) and hsl," 2006.

[77] J. L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia, "Diatom autofocusing in brightfield microscopy: a comparative study," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 3, pp. 314–317 vol.3, 2000.

[78] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, p. 318–362. Cambridge, MA, USA: MIT Press, 1986.

[79] P. Baldi, "Autoencoders, unsupervised learning and deep architectures," in *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, p. 37–50, JMLR.org, 2011.

[80] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[81] M. Maggipinto, C. Masiero, A. Beghi, and G. A. Susto, "A convolutional autoencoder approach for feature extraction in virtual metrology," *Procedia Manufacturing*, vol. 17, pp. 126 – 133, 2018. 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USAGlobal Integration of Intelligent Manufacturing and Smart Industry for Good of Humanity.

[82] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[83] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[84] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[85] T. Caliński and H. JA, "A dendrite method for cluster analysis," *Communications in Statistics - Theory and Methods*, vol. 3, pp. 1–27, 01 1974.

[86] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[87] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[88] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," *Univ. Minnesota Supercomp. Inst. Res. Rep.*, vol. 213, 01 2003.

[89] S. Thilagamani and S. Moorthi, "A survey on image segmentation through clustering," *International Journal of Research and Reviews in Information Sciences*, vol. 1, 01 2011.

[90] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin, *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[91] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[92] M. Revilla and C. Ochoa, "Forum: Ideal and maximum length for a web survey," *International Journal of Market Research*, vol. 59, pp. 557–566, 2017.