NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

GUILHERME FIGUEIREDO RIBEIRO
BSc in Computer Science

AN INTERACTIVE PLATFORM FOR REPRESENTING, INTERLINKING AND ANALYZING CONTENT ON ART

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
September, 2022

DEPARTMENT OF
COMPUTER SCIENCE

AN INTERACTIVE PLATFORM FOR REPRESENTING, INTERLINKING AND ANALYZING CONTENT ON ART

GUILHERME FIGUEIREDO RIBEIRO

BSc in Computer Science

Adviser: Dr. Erik Heimann
Managing Director, IT-OBJECTS

Co-adviser: Pedro Abílio Duarte de Medeiros
Associate Professor, NOVA University Lisbon

Examination Committee

Chairs: Teresa Romão
Associate Professor, FCT-NOVA

Matthias Knorr
Assistant Professor, FCT-NOVA

Dr. Erik Heimann
Managing Director, IT-OBJECTS

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
September, 2022

An interactive Platform for Representing, Interlinking and Analyzing Content on Art

This document was created using the (pdf/Xe/Lua)LaTeX processor, based on the NOVAthesis template (v6.6.5), developed at the Dep. Informática of FCT-NOVA by João M. Lourenço. [34]

Acknowledgements

First and foremost, I want to thank both my advisers, Dr. Erik Heimann and Professor Pedro Medeiros, for their assistance and support during this thesis writing process. I would like to thank the NOVA School of Science and Technology, the Informatics Department, and all of its professionals for all of the education they've provided me over the past five years, as well as for providing me the tools I will gladly carry with me for the rest of my life. I am also grateful to all of the IT-Objects team, particularly Patrick da Silva, for their assistance and support, especially for allowing me to conduct my master dissertation thesis with them and for trusting in me and believing in my work.

I would like to thank my girlfriend Marta for all of her help and support over the course of our seven-year relationship. She has always been a huge source of inspiration for me to follow my objectives and goals. Always challenged me to think outside the box and step beyond my comfort zone, which helped define me into the person I am today. My family has also been a great help, mainly my parents, and I owe them a heartfelt thanks for giving me all I needed during my academic career and as a person, providing me the opportunity to continue my academic journey.

I also owe a big thank you to my friends, Diogo Barreiros, Henrique Ribeiro, Rodrigo Gomes, Ronaldo Corte, Rúben Vaz, and João Ginginha, for all the moments we have shared, all the help and support along this academic journey. Last but not least, I want to express my gratitude to my school friends, with whom I have shared my biggest academic and personal successes and who have supported me through thick and thin ever since my first years of education. You have consistently motivated and encouraged me to push my limits and go beyond with your support. I've shared a lot of memorable moments with you that I'll never forget.

“Sic Parvis Magna”

Abstract

Unfortunately, many art objects such as paintings and sculptures are not yet cataloged in digital form. Often not even in printed catalogs or books. This circumstance provides a considerable obstacle for obtaining information from different collections within museums or private collections all over the world. Because a significant number of works of art are also never cataloged, either because the owners lack the knowledge to do so or because there isn't a simple tool to help the cataloging process, by providing a large amount of information in an accurate way, a significant number of these works of art also remain unknown to the world.

With the help of machine learning algorithms and tools that can interpret images contents, a platform is to be created that offers a simple data acquisition from art pieces. In addition to simpler technical attributes such as style and type of art, more detailed information such as probable artist, materials, and decade of creation could already be suggested to the user during the data acquisition. This is possible thanks to the algorithms that can analyze images of the object in question. Alongside these suggestions, there would be the possibility to interlink data by multiple features in order to get similar art pieces to the submitted one. From the user utilization of the platform, the growing data set shall be linked with each other, visualized in an interactive platform, and finally serve to improve the analysis of new art objects.

The creation of this platform made it feasible to offer a wide amount of information about any work of art. Four separate image classification models that could categorize art pieces based on their author, style, decade, and type were developed. To improve the prediction power of these, a number of strategies were studied and applied. Further details, including a set of relevant tags, the most dominant colors, and the materials utilized in the pieces of art, were also provided using Google Cloud Vision API. The platform's whole infrastructure was built with these features and best practices in mind.

Keywords: Art, Platform, Machine Learning, Algorithm, Catalog, Interlinking, Analysis, Predict

Resumo

Infelizmente, muitas peças de arte, como pinturas e esculturas, ainda não estão catalogadas em formato digital. Muitas vezes nem mesmo em catálogos ou livros. Esta circunstância constitui um obstáculo considerável para a aquisição de informação de diferentes coleções de museus ou coleções particulares em todo o mundo. Igualmente, uma grande quantidade de peças de arte não são catalogadas tanto por falta de conhecimentos por parte do proprietário ou por falta de um método simples de as tornar públicas.

Com a ajuda de algoritmos e ferramentas de aprendizagem computacional que conseguem interpretar o conteúdo de imagens, uma plataforma que oferece uma ferramenta de aquisição de dados de peças de arte será criada. Além de atributos técnicos mais simples como o estilo e tipo de arte, informações mais detalhadas como o artista provável, materiais e a década de criação poderiam ser sugeridas ao utilizador durante a aquisição dos dados. Isto é possível graças aos algoritmos que conseguem analisar imagens do objeto em questão. Além destas sugestões, haveria a possibilidade de interligar os dados por diferentes atributos de forma a obter peças de arte semelhantes à que fora submetida. A partir da utilização da plataforma pelo usuário, o crescente conjunto de dados deve ser interligado entre si, visualizado numa plataforma interativa, e por fim ser utilizado de forma a melhorar a análise de novos objetos de arte.

Com o desenvolvimento da plataforma, é possível oferecer uma grande variedade de informação. Foram desenvolvidos quatro diferentes algoritmos que conseguem categorizar uma peça de arte de acordo com o seu artista, estilo, década e tipo. Para melhorar a performance destes algorimos, um conjunto de técnicas foram estudadas e aplicadas. Mais informação, como um conjunto de palavras relevantes, as cores mais dominates, e os materiais usados nas peças de arte, poderam também ser sugeridas usando a Google Cloud Vision API. Toda a infraestrutura da plataforma foi desenvolvida tendo em conta as funcionalidades necessárias e as melhores práticas.

Palavras-chave: Arte, Plataforma, Aprendizagem Computacional, Algoritmo, Catalogar, Interligar, Análise, Prever

Contents

List of Figures

List of Tables

Acronyms

AF	Activation Functions 13, 14, 16
AI	Artificial Intelligence 19
API	Application Programming Interface 21, 71, 72, 73
AV	Activation Value 14
CNN	Convolutional Neural Network x, xiii, 15, 16, 19, 24, 25, 35, 36, 38, 39, 40, 45, 47, 49, 84
DCNN	Deep Convolutional Neural Network 19
DNN	Deep Neural Network 14, 34
GPU	Graphical Processing Unit 34, 35
HL	Hidden Layers 14
IC	Image Classification 1, 3, 4, 10, 19, 35, 37, 38, 44, 53, 64, 72, 75, 84, 85, 86
IR	Image Recognition 1, 35, 38
K-NN	K-Nearest Neighbors 8, 10, 11, 84
L1D	L1 distance or Manhattan distance 10
L2D	L2 distance or Euclidean distance 10
LC	Linear Classifier 11, 13, 14, 84
ML	Machine Learning 3, 4, 5, 13, 18, 20, 22, 29, 34, 41, 63, 69, 85
NN	Neural Network 13, 14, 15, 19, 34, 35, 36, 47, 54, 55, 84

RGB	Red, Green and Blue 11
SQL	Structured Query Language 72
URL	Uniform Resource Locator 72

1 Introduction

1.1 Context and description

There are millions of works of art in the world [20]. A challenging task would be to gather all these. Also, the fact that new works of art are continuously being produced or discovered makes this predicament more obvious. One other fact that could be identified is that there is a large amount of art pieces that are unknown to the world [70] [36]. Whether because sometimes the owners don't have the required knowledge to catalog them or because there is no tool to help this process by providing a wide variety of information in an accurate way. Because of this, the main objective of this thesis is to create a platform that, among other features, can facilitate the process of cataloging works of art. When developing a platform that allows users to interact with artworks in various ways, it is crucial to depend on data. To have a good interaction with both the searching and interlinking of works of art, it is important to have a significant quantity of data. With the help of a computer program, this process could be automated, with a wide variety of attributes being accurately suggested to the users. For this, it would be required to create a tool that not only can interpreter the contents of images of pieces of art, but also recommend some information regarding of them.

Image Recognition (IR) is the task of getting an image and, with the help of a computer program, automatically analyzing and detecting its contents [24]. This task is done by gathering relevant information from the input and eliminating irrelevant variables. Afterward, this technique can be used to create an Image Classification (IC) tool. Once the contents of the image can be efficiently detected, the image can be classified. In other words, a IC is able to combine images in groups, also known as classes, depending on their contents. There are a lot of algorithms and techniques that can perform these tasks, but they are not very frequently used in art with the identification of multiple attributes. Since most of the time, only individual features are extracted from art pieces [68] [33], the main focus of this thesis is to develop a method that is able to extract multiple features from those pieces of art.

When solving an IR problem, the main characteristics of the images must be identified.

With that in mind, it is possible to understand what set of features can be identified across similar images in order to group them under a class. This is also known as classification. This step is useful to distinguish images between distinct classes since different types of images will have a different and unique set of characteristics. In the context of this art feature's extraction, it was necessary to, given a set of images that belong to a set of classes, identify the unique features for each labeled class. With this knowledge, the computer program can identify for a single piece of art, the previously known set of characteristics and, with that, classify it. For example, in an art movement recognition problem, the computer program will search for the most common features that are similar across the whole art movement. Cubism is a style that is known for its geometric shapes and brighter colors [64]. For the human eye, it is simple to distinguish a painting from this art style when compared to Luminism, which can be described for its naturalistic landscapes, often seascapes or river views, emphasizing the treatment of light [35]. The ability to develop an algorithm that can find these differences, and, with them, correctly classify different art pieces, is the key to understanding the difference between art movements. In the context of art pieces, this method can be applied to other parameters, such as author or creation year.

1.2 Motivation

Nowadays online usage is constantly increasing and reaching every day more people around the world [40]. For that reason, there are a lot of different people navigating the internet every day. Sometimes, these people, can not go to an art gallery or museum to see art pieces, opting to have some kind of virtual experience. From a different perspective, some people may own works of art that they would like to expose to the public but lack the resources to do so. Whether it's because they lack the knowledge to identify works of art or because there isn't a simple, straightforward way to do it. Because of this, there is a barrier between the users and the traditional cataloging process. In order to address this, there must be an easy-to-use interface that allows users to engage with artwork in addition to cataloging it.

The human eye is capable of quickly and accurately understand items from among tens of thousands of possible options within a fraction of a second. This is valid from the basic interpretation of simple features in an image, for example, objects, to other more complex contents [18]. However, even being able to detect them, the human requires knowledge on the subject to be able to comprehend it. This effect is visible in art, where a person must have some knowledge in it to be able to understand and classify it. This same task is quite different when performed by a computer. This happens because the computer can not interpreter images so easily from scratch. It must be previously feed with knowledge. This can be acquired by a training process. This process provides the tools to correctly identify contents in different situations. In art, this can be even more challenging since, sometimes, it is a difficult task to identify the contents of an image, even with prior knowledge. For example, in two different paintings, one object can be represented in different ways, depending on the artist who created it or the art movement. Taking this into account, the main challenge is to create an algorithm that is robust enough to correctly classify art even under non optimal circumstances.

1.3 Objectives

The main goal of this work is to create a user-friendly platform where users can interact with art in different ways and according to their interests. The key features of this platform is the possibility for the user find art pieces as well as catalog them. The cataloging process is thought to be automated to simplify this process. With this, it will be possible to gather a large amount of information from a single image of the art piece. This tool is supposed to be an efficient way to catalog art pieces by any type of users, even those who do not have a vast knowledge in art. This platform will be developed to be used in a web interface since it is supported by a wide variety of devices, such as computers and smartphones. This platform is meant to be used in order to represent, interlink, and catalog art pieces that are spread around the world in a simple, intuitive, and fast way that is available to everyone.

1.4 Methodologies

The main challenge when developing the proposed platform, as stated in section 1.2, will be the image analysis. This feature will be supported by algorithms that will use Machine Learning (ML) techniques and tools in order to suggest the most accurate values for the intended features. To create these algorithms, there will be necessary to understand the fundamentals behind the capabilities of a computer to understand visual features that the humans can identify. For this, there are several methods that will be studied and discussed in this document. The main goal of this first phase is to understand what are the currently available methods to implement a methodology like this and to have knowledge on the best practices when developing an IC tool. The development of these algorithms following the best practices will not only provide the highest possible accuracies when predicting data but also provide a good performance when doing so. To achieve this, when developing the algorithms, the time performance matter will be taken into account. It is thought to be develop four different algorithms. Each one will be used to acquire data from four different aspects of art pieces. The author, style, type and a time-related information, such as the year, will be the features that will be extracted by these IC algorithms. To create these algorithms, a deep study will be conducted when developing one of the proposed feature extraction's model. This process will help to understand what is the process of developing and optimizing an IC algorithm. It will then be used the same process for the other proposed algorithms. Additional methods will be explored to understand if there is a possibility to acquire even more data from art images. The platform will also be developed in a way that, with its usage, the algorithms will be improved. In addition to the IC features available in the platform, the interface itself will be designed and implemented so that it is as simple and intuitive as possible for different types of users. It must be available and easily usable to everyone regardless of their knowledge or experience of web interfaces or art. The way the platform is to be developed will also be available to the majority of mobile devices since it will be available from a web browser. Since it is easier to carry a small device, such as a smartphone when interacting with pieces of art, this platform must be developed in a compatible environment to a wide range of these devices. The other reason is that nowadays, the majority pictures are taken from smartphones [58].

1.5 Contributions

For this platform, more features were thought to be implemented than the IC feature. It is thought to implement a search engine, where users can search for a piece of art by multiple parameters, such as name, style, author, and more. The possibility to represent pieces of art around the world was one of the key aspects of this platform. Therefore, there must be an easy way to find pieces of art. Alongside this feature, one other feature that is going to be developed is an interlinking data system. This means that a user will be presented with art pieces that are related to some other art piece that is being visualized or interacted with. The challenge is the need to create a system that understands the similarities between art pieces. For example, if a user submits a piece of art to be evaluated, alongside the suggested information provided by the algorithms, users will be able to visualize similar art pieces with the uploaded one.

1.6 Document Structure

This document is structured in different chapters. Here is a brief summary of what is the structure of it and the contents of each chapter.

- **Introduction** In this first chapter of this document, a complete description of the challenge that was identified was done. It starts with a brief description of the context behind the development of this thesis. Afterward, it describes what problem the development of this thesis is intended to solve.

- **Fundamental Concepts and State of the Art** The second chapter of this thesis is a study of multiple aspects in the context of the problem, mainly what are the main features to categorize art among different styles, authors, and others. Also, some common ML tools, techniques, and their fundamental concepts were studied. A comparison between these methods and a discussion is presented. At the conclusion of this phase, there is an understanding of the available methodologies. Doing this, it was possible to know which set of tools can be used in the thesis development.

- **Related Work** Some existing similar tools are presented in the third chapter. This first step aims to define how the work that was developed in this thesis will be distinguished from these. Also, results from this study were used as a base point and as a comparison for the development of the platform.

- **Prepossessing of the Data and Database Creation** In the fourth chapter of the document, the data needed and used for the development of this thesis was analysed. Different aspects of the data are explained in depth, such as analysis, transformations, organization, and all the optimizations that were done to it alongside the used methods. Since different ML models were created, different optimizations and transformations were also applied according to the context of each model. These differences are also be described in this chapter and a brief explanation of them is given. Still in this chapter, the data is arranged into a single database to make it simpler to utilize in various situations during the platform's development.

- **Development of Machine Learning Algorithms for Art Classification** This chapter of the document describes the whole approach when creating the machine learning algorithms for the proposed image classification problems. The data that was processed and optimized in the previous chapter is the base element for this process. All the developed ML algorithms, with the respective training and validation methods, architectures, and tools that were used are deeply described and discussed. Additional technologies are also investigated, put to the test, and used in this chapter to support the feature extraction process from works of art.

- **Infrastructure Design and Platform Implementation** This section of the document describes the architecture of the infrastructure created to develop this platform. This architecture is composed of several elements that were designed according to the platform's needs. One of these elements is the interface. Not only the design process of it is discussed in this section, but also the justification for several decisions during the development of it.

- **Experimentation and Analysis** In this chapter a set of different tests applied to multiple elements of the developed platform were conducted. These were created to understand the behaviour of different elements, such as the ML algorithms and the platform's interface. Their results are presented, analyzed, and discussed in this section.

- **Conclusions and Future Work** The final chapter of this document describes the conclusions that were taken from the development of this thesis. Here it is discussed if the developed work can be seen as successful according to what was initially thought. Also, in this section, a brief discussion of the acquired results and the development process is used to understand in what ways the developed work can be improved and complemented in the future.

2 Fundamental Concepts and State of the Art

In this chapter, an explanation of the requirements to create an art classification algorithm will be done. Mainly, it will be described the main features that can be quickly identified by the human eye in art pieces. This brief study is useful to understand what features must be taken into account when creating the algorithms. It will also provide an idea of which features are the most relevant according to the goal of the algorithm. For example, to classify images among their art movement, the most relevant features might be different from the ones of an algorithm that classifies images among their creation year. These features can be colors, objects, or other contents. Since this algorithm will be used in everyday life situations, the conditions in which the platform is going to be used must be discussed, this is, the type of images are expected to be analysed by the platform. With this knowledge, it is possible to prepare the algorithms for these conditions. Also in this chapter, there is a study of some of the most traditional methods used in image classification problems. These methods are described and discussed. This is useful to understand whether they are relevant to the development of the art recognition problem or not. Additionally, researching these practices that are already in use helps determine how to improve the proposed work and how it can be applied to art classification.

2.1 Art Classification

Art pieces can be classified as highly subjective. This subjectivity is not based on established facts, but rather on individual beliefs and sentiments [60]. For this reason, it can be interpreted in different ways by different people. A piece of art, for example a painting or a sculpture can represent an object or a situation that can be seen in different ways. On the other hand, some art pieces are very realistic and can represent reality as the human eye knows it. The main challenge with the proposed work of this thesis is to create an algorithm that can classify images of art pieces even with this subjectivity aspect. Therefore a study was carried to understand what are the most relevant features among art pieces, so that the classification algorithms can also understand those differences. As said in the introduction of this chapter, the platform is meant to be used in everyday life situations. The most common usage of it will be through a mobile device, such as a smartphone. Although a large number of smartphones nowadays are equipped with high-quality cameras [57], the platform must be also prepared to receive low-quality images. Also, it is important to create an algorithm that can identify images even in non optimal situations, such as a picture with conclusions from other objects or slight transformations. This exact situation can happen in public places, such as museums, where the platform is expected to be used.

In some cases, it is possible to highlight some features in art pieces that are relevant when classifying them. For example, assuming an artist has a very specific painting or sculpture style. It is expected to be an easy task when identifying art pieces of that artist since all its artworks are somehow visually similar. However, this situation can not be seen as a universal rule, since artists can change their art style several times in their lives, or even create art from different styles at the same time. Therefore, there must be other parameters that can be used to classify art authors. One other possibility is to identify their technique. Assuming artists, have unique brushstrokes when painting, or have a specific technique when sculpturing, these could be used to identify an art piece's author. However, to be able to do this, there must exist a large high-quality images data set, to be able to train the models to correctly identify these features. One other aspect that must be fulfilled is the need for high-quality submitted images by the users. Despite the smartphones camera resolution is being improved with time [57], this aspect is unlikely to be guaranteed by common mobile device cameras.

Although it is possible to identify some features of art pieces that can distinguish them from each other in several aspects, it is very hard to identify all these features or a set of features that is generic enough that can be correctly used across every situation. Therefore, the existence of an algorithm that can automatically identify those differences, simplifies this process. These features known by the algorithm are the elements that are used to classify images among different aspects, such as the art movement or the author.

2.2 Known machine learning classifiers

To have a better understanding of the methods that could be used to develop the platform's recognition tool, a study of some of the most popular tools available was done. In the context of this thesis, to create an image classification algorithm for some features in pieces of art, it is necessary to use a classifier. Nowadays, there are a lot of known classifiers that can be used in very different fields. Some of them have shown to be better with image classification, others have better performance in raw data, such as numerical analysis. Supervised learning and unsupervised learning are two subcategories of machine learning. Based on the utilized data and their accessibility, this differentiation is made. When the learner already has some knowledge of the data or when the output has been provided and labeled, the learning process is considered supervised. Unsupervised learning, on the other hand, does not equip the learner with any prior knowledge about the input or the output [63]. To understand which is the method that has better performance in solving the proposed work, some classifiers are presented, discussed, and tested in this section. Supervised learning techniques were studied for the image classification challenge that this thesis proposes. This was be done because not only the data that was used is labeled, but also because the goal is to develop an algorithm that can categorize images under a set of classes that are deterministic. Three of the most common supervised learning classifiers were put to the test in this section [54] [9] [44]. Those classifiers are the K-Nearest Neighbors, Linear Classification, and lastly Neural Networks. This last classifier was deeply analyzed since there are some variations of it.

2.2.1 K-Nearest Neighbors

Starting with a simple classifier, K-Nearest Neighbors (K-NN) is one of the most popular ones. This classifier is known for its prediction method and simplicity [2]. The fact of this is one of the simpler algorithms to implement is that this classifier, in practice, does not learn anything. This means that the classifier is not able to improve itself by learning from previous mistakes. When predicting, it relies directly on the data. This is also known as Lazy learner [38]. When classifying, it searches in the known training set for the K most similar neighbors. This K value, in this context, represents the number of images that are presented on the output of the classifier when classifying one image. This value must be carefully chosen. It should not be so small that the output ends up being very exclusive, having higher probabilities of failing when classifying. For example, if $K = 1$, the probability of the correct prediction class being present in the output label is smaller to an higher K value. On the other hand, if K has a very high value, the output classes of the result, can be very generalized, which can be translated in a low efficient prediction, since a large variety of classes can be outputted. Therefore, the choice of this K value must be a balance between these two ends [73].

Despite it being useful in some situations this classifier is not the best when applied in real-world image classification problems [55] [6]. This can be seen in a situation where a set of output images are visually similar between them but do not belong to the same class, therefore not making the right prediction.

To observe this classifier behaviour applied to an image classification problem, Fei-Fei Li et al. [73] applied it to the labeled data-set CIFAR-10 [13]. This data-set consists of a set of 60000 images labeled equally among 10 different classes. Each image is 32 by 32 pixels. Applying this classifier with a K value of 10, meaning that the output is based on the 10 most similar images, the results obtained by the author can be seen in Figure 2.1.

The image shows the results after applying the K-NN classifier in ten different images. In each row, it can be seen each test scenario. The first element of each row is the image that was used as input. At the right, there are other 10 images. These images correspond to the 10 most similar images, according to the K-NN algorithm. Analyzing the results, it

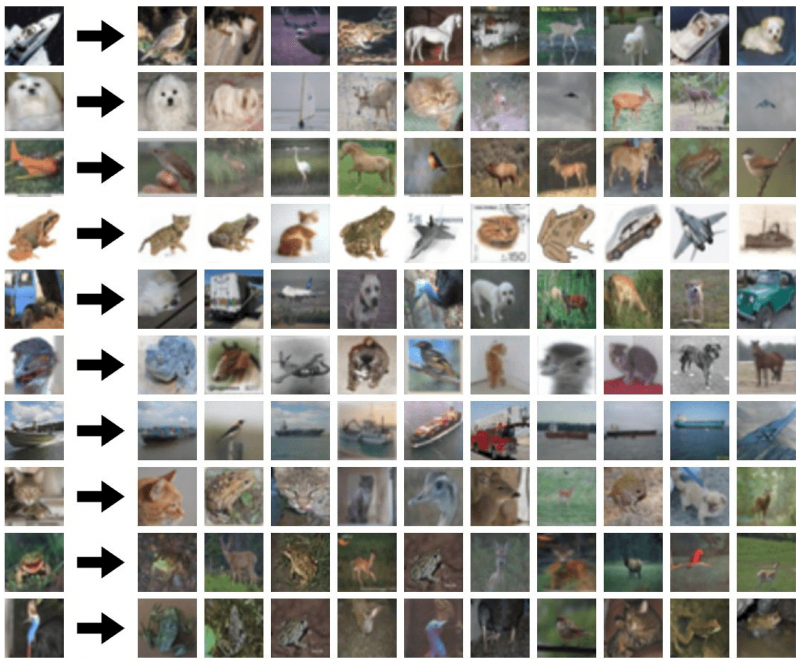


Figure 2.1: Nearest Neighbor Prediction algorithm applied to the CIFAR-10 images dataset [73]

can be seen that five of the submitted images got the wrong first prediction. This means that, among the whole data set, the most similar one belongs to a different class. In the case of the frog image, presented in the fourth row of the figure, it is possible to see that the nearest neighbor is a very similar image when compared to the input image. Despite that, the first returned image corresponds to a cat, therefore, the algorithm made an incorrect prediction. This is visible throughout the results on the other tested images. Not only the first prediction is wrong, but when a deeper analysis is made through the set of the output of each prediction, there are very distinct results. This means that there is not a consistent output prediction class. Using again the frog input image, the 10 outputted images have very distinct classes, such as images belonging to the cat, airplane, car, and more. Among all the 10 returned images, only 3 of them were correctly identified.

As mentioned in this section, this classifier relies only on the existing data to make predictions. This means that there is no training phase. However, in big data context the testing phase is slow, with a time complexity of $O(nd)$ [16]. In this expression, n is the size of the training set and the d is the dimensionality [65] [16]. This can be explained due to the fact that, in the testing phase, the algorithm compares the input image with every individual image in the training set looking for the K most similar ones [38]. This process consists in taking, sequentially, every pixel of the test image and comparing it with the corresponding pixel of every image in the training set. This operation is repeated for every training set image. When comparing both pixels of both images, a deterministic function is applied to return a value that represents the difference between both pixels. All these generated values are summed and return a value that represents how similar both images are. When this phase of the algorithm ends, the K images with the lowest values are presented by the order of that value [73]. When comparing images with a deterministic function, there are multiple options. Two of the most popular are the L1 distance or Manhattan distance (L1D) and L2 distance or Euclidean distance (L2D) [62].

L1D computes the sum of the absolute difference of pairs of pixels, represented as:

$$d1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

On the other hand, L2D computes the sum of the square root of the square of the difference of all compared pixels. Mathematically, it can be represented as:

$$d2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

A visualization of the difference when applying these distance methods can be seen in Figure 2.2. This figure shows three different areas that represent three different classes location in a two dimensional space. These areas were generated by applying the K-NN method, with a K value of 1. Any point that is located inside each colored area is classified as an element of the class represented by the respective color. For example, any new point that is located inside the green area is classified as being part of the class represented with the green color. This happens because the K closest points are also green, depending on the distance function that was chosen. These areas can be quite different from each other. When applying L1D, the areas are delimited by lines that tend to follow the direction of the coordinate axis. Contrarily, when using L2D, these decision boundaries have an independent direction. This second approach creates more natural boundaries. This happens because L1D depends on the choice of the coordinate axis system, while the L2D metric, is completely independent of the axis orientation [73].

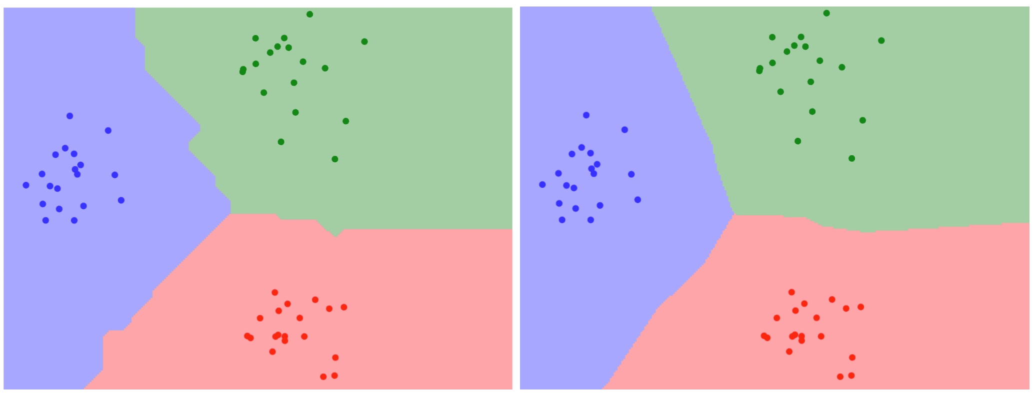


Figure 2.2: Comparison between K-Nearest Neighbor using Manhattan Distance (figure at left) and Euclidean Distance (figure at right) in a 2 dimensional space. [30]

This classification method is not very useful to solve the IC proposed problem of this thesis. There are two main reasons for this. Firstly, when applied to image classification, this classifier may result in a poor accuracy value, as seen in Figure 2.1. As it was presented, when testing images from a large data-set, the K-NN just computes an analytical comparison between pixels between the input image and every known image in the training set. This means that this method does not try to identify the contents of the images or even learn from mistakes. In the case of recognizing art pieces, this method should not be a good solution, since different art pieces may be visually similar among them, but not part of the same class. The other reason to discard this classifier is the time complexity of the testing process. Since for testing, the algorithm compares the given image to every single image on the training set, it would provide a very slow experience every time a user would request a prediction. Furthermore, with the growth of the data, this problem would be more significant, since the bigger the training set, the slower it would take to make a single prediction.

2.2.2 Linear Classifier

One other known classifier is Linear Classifier (LC). LC approaches the problem by automatically creating a linear function in order to make predictions of a given image. The prediction result is composed of a set of scores that represent how probable the test image belongs to a known class [73]. This classifier relies on a mathematical linear function with the following structure:

$$f(x,W) = Wx + b$$

This linear function is composed of an input image x and a set of parameters W, which are automatically generated. In contrast to the prior methodology, this classifier is based on an algorithmic improvement process. Training phase is the name of this stage. In each iteration of the training process, the values in each entry of the W matrix are adjusted by comparing the intended value and the obtained value. This process is also known as propagation and back-propagation [71]. With this, the classifier is able to learn from its own mistakes.

The linear function can be structured in different ways. The main goal of it is to create a function that combines the parameters in a way that is possible to create a good prediction solution for the intended context [73]. One of the most simple functions that can be used, as exemplified by Fei-Fei Li et al. [73], is to multiply both "x" and "W" parameters. This linear function can be represented in a diagram, as seen in Figure 2.3.

In the diagram, the input image that is classified is represented by a matrix of $ImageSize*3$ rows per $ImageSize*3$ columns. Each entry of this matrix is a numerical representation of one of three values for a single pixel of the image. Each pixel of the image is composed of 3 different values. Each one represents a value for each Red, Green and Blue (RGB) color channel. Since a pixel can be separated into values corresponding to the 3 different colors, RGB, an image can also be represented with 3 different arrays of values. This matrix is then converted into a single column with $ImageSize^2*3$ rows. W is the matrix with

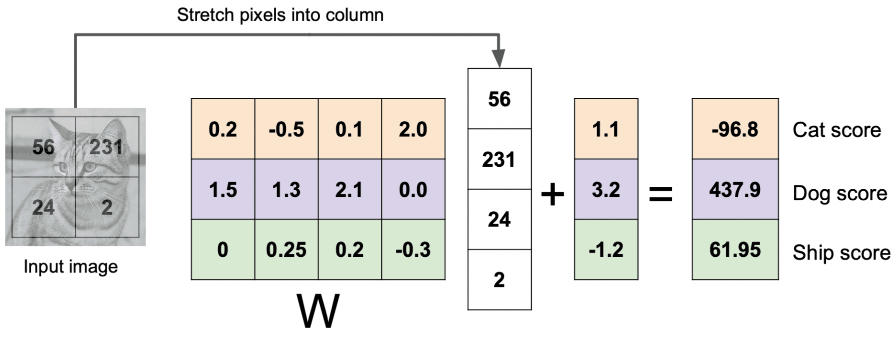


Figure 2.3: Diagram representation of the Linear Classification function and all its components [73]

the same number of rows as the number of classes per $ImageSize^2 * 3$ columns. In this example, this matrix is multiplied by the input image matrix. In each entry of every row of this W matrix, it is represented every generic pixel that was generated by the algorithm after the training process. Finally, the last column is an optional and data-independent parameter, which can be used to represent computational preferences. This function outputs a set of scores which represent, the scores of the respective classes. Every entry represents the probability of the input image being part of the class represented in that row.

Figure 2.4: From left to right: Three W matrix lines transformed into images that represent the “Cat”, “Dog” and “Ship” classes [73]

In the context of this problem, where the training data-set is a set of images, each line of the W matrix can be used to produce a visual representation of the knowledge of each trained class, where each entry of each row represents a pixel of that image. In some cases, where the class does not change too much across the data set, this image can be a good representation of the class. In real-world situations, when classes represent contents that have high variations in different situations, this generalization produces images that are not representative of the classes that are trying to be represented. This happens because the classifier is only allowed to create one single template image for each class. To visualize these templates, Fei-Fei Li et al. [73] produced some images after applying a LC to the CIFAR-10 data-set [13]. In Figure 2.4 the images are the representation of the average "Cat", "Dog", and "Ship" classes from the training set, respectively. Although the algorithm was able to produce these templates, they are not a good visual representation of the respective classes to the human eye. Since every represented class can have several shapes and colors, these images can't be seen as a good representation of the class, therefore, the ML algorithm is not able to use it effectively to correctly predict new images. Also, as can be seen in Figure 2.4 the ship class representation is very different from the other two generated images. Although this image is substantially different when compared to the other two images, when trying to predict new images, it might be confused with other images where there is a great blue color presence, such as "Airplane" or "Fish". This effect can be seen in the other two generated images. Both the cat and the dog generated images are visually very similar, which makes the prediction task harder and not reliable between these two classes. This translates into a low efficient classifier when predicting real-world images with high visual variations. For this reason, this method was not applied to the art classification problem proposed in this thesis.

2.2.3 Neural Networks

From previous analysis in subsection 2.2.2, the usage of LC has its limitations. The main problem is the impossibility to recognize classes with high variations among their images. Therefore, one solution to this limitation is to create an algorithm that can classify different images that belong to the same class. Instead of using only one LC, one solution is to stack multiple of these functions. The stacking of multiple of these methods is also known as Neural Network (NN). NN is the combination of multiple layers of linear functions that, combined, can produce better results when compared to the LC itself [73]. Just like the previous classifier, a NN can be represented with a mathematical expression.

$$f = W_2 max(0, W_1 x)$$

In this expression, the NN is composed of two layers. The first one, similarly to the previously analyzed algorithm, $W_1 x$ outputs a set of scores representative of how likely an image belongs to a class. However, this time, it has a higher number of returned values. This translates, visually, into having more templates for each class. While in the previous classifier the generated template for the class "Dog" was a brown blurred image, using this multi-layered architecture, there will a higher number of variations for a single class. The second layer uses a non-linear expression, in this case, represented by the $max()$ function, and applies it to the previously computed layer results. The weighted sum of the outputs from the preceding layer serves as the input to each node in such a network [31]. In each layer, the node's Activation Functions (AF) then transforms this set of values. This means that a certain pattern of values are transformed into a new set of values in the next layer by a function [31]. Afterward, these combined values are used to calculate the final scores for each intended class in the last layer of the NN.

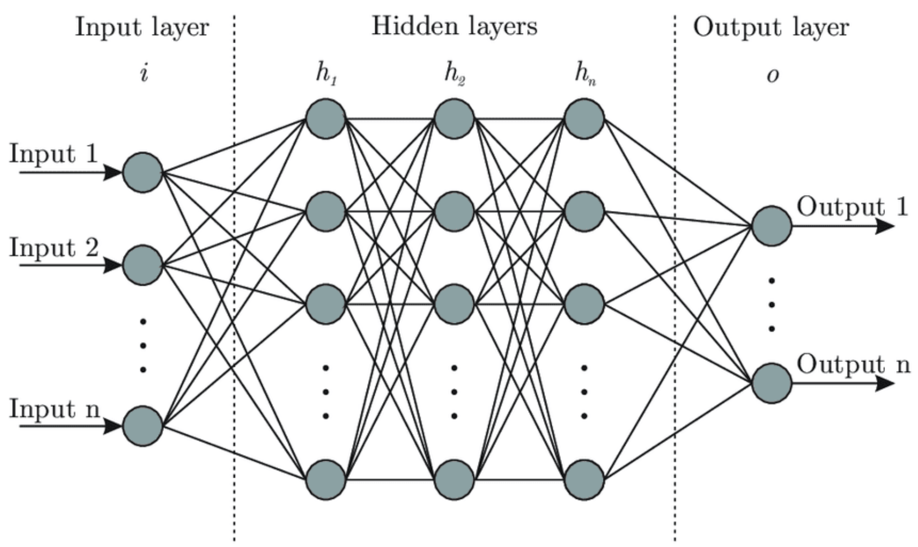


Figure 2.5: Different types of layers in a Deep Neural Network visual representation [8]

While the previous example only uses two layers, it is possible to combine multiple layers and create a multi-layered NN, also known as a Deep Neural Network (DNN). A DNN can be separated into a 3 stages structure [39]. The first one is the first layer, also known as the input layer. In the context of this problem the input layer is the set of all pixels of the input image. Among other information, such as computational capabilities, every neuron has a value, also known as Activation Value (AV). Every AV of every node is propagated to the next layer which is part of the second stage, a set of Hidden Layers (HL). These HL are the components of the DNN that really compute the solution for the proposed problem and can be seen as the brain of the DNN. Here, each combination of activation values from the previous layer triggers a new set of neurons in the current layer. The activated patterns depend on the weights of the model and the AF. These weights are automatically generated during the training phase of the algorithm. These weights are adjusted in a way that the model can produce the best results as possible when predicting new images. Similar to the LC classifier, the values are generated throw iterations on the model. The values are passed throw the layers and, in the last layer, compared with the values that were expected. The process is then inverted, passing again values throw the network so that the weights are readjusted. This output layer, the third and final stage of the network, has the same number of neurons as the number of intended classes. Each neuron has a value that represents the probability of the input image being classified as the respective class. Therefore, the highest value in this layer represents the most probable class of the input image.

The computational power of every neuron in a NN depends on a non-linear function called AF. These functions are applied to the individual neurons, enabling them in a limited range. Without it, this classifier would act as a Linear Regression Model with limited performance [56]. There are multiple AF, such as “Linear”, “Sigmoid”, or “ReLU” functions. The choice of these AF can be determinant to achieving better performance and fewer errors. Other techniques can be applied to improve the classification accuracy, such as the number of hidden layers in the network and the training methods [56].

The training process of these NN is composed of two phases. In the first phase, also known as the forward phase, the algorithm computes the input values and compares them to the desired results, calculating the error function. The second phase can be described as an adjustment of the weights of each neuron connection in order to reduce this error function. This process must be repeated until the error is low enough [14] in order to make as many correct predictions as possible.

2.2.4 Convolutional Neural Networks

CNN is a type of NN that has been showing great results in late years in a great variety of fields related to pattern recognition areas [3]. The main difference between CNN and a common NN is the presence of two new types of layers. Convolutional layers and Pooling layers. To have better results when predicting, these layers are usually combined with the regular layers that were already described in subsection 2.2.3. These layers are used to transform the input data in such a way that it is then passed to a regular NN as much optimized as possible [53]. A brief representation of a simple CNN is shown in Figure 2.6. This network contains two pairs of Convolutional layer (C_n) with a Pooling layer (S_n). During the training phase, the output of the first pair, after being processed by it, is passed to the next Convolutional and Pooling pair, where is computed again. After the data has been passed through these pairs of layers it is then passed to a traditional NN represented at the end of the figure (NN).

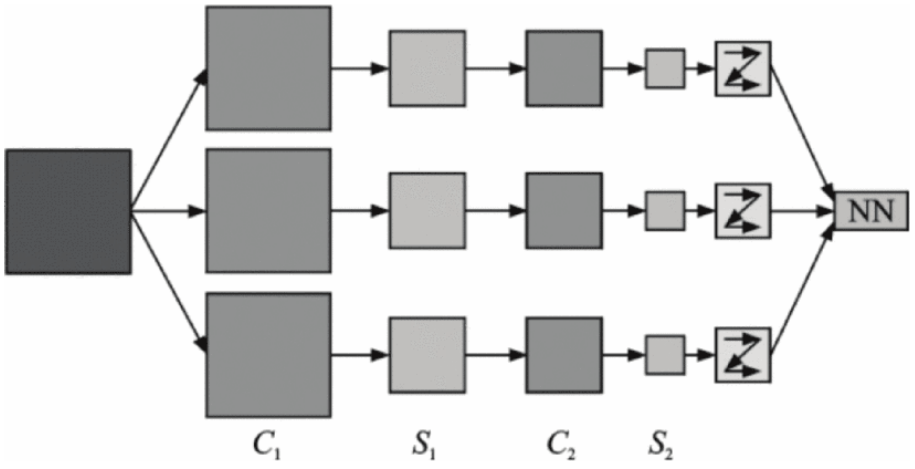


Figure 2.6: Convolutional Neural Network architecture representation, focusing on the Convolutional and Pooling layers [52]

The most beneficial aspect of CNN is the ability to reduce the number of parameters that are passed throw the NN [3]. This technique has been used to solve problems with larger models, which was not possible with common NN. CNN are an optimized tool to identify patterns in images [52]. This is possible thanks to the usage of filters that are present in the Convolutional Layers of the network. These filters are learned along with the execution of the algorithm and make possible the identification of some features in the image [72]. Some filters are simpler than others and, depending on the filter itself, it may identify characteristics such as edges, lines, or even other more complex contents. For instance, deeper convolution layers of CNN can identify face characteristics like eyes and mouths regardless of where they appear the image, whether some basic components, such edges or simple shapes, are recognized in the initial convolution layers [3]. Since these filters are produced automatically, deeper layers may be able to extract from images more subjective information. In the art identification context, this aspect is essential, since it is needed to identify unique and sometimes very abstract features in art pieces.

After applying these filters, to reduce the number of parameters, the CNN uses Subsampling or Pooling Layers [61]. These layers take the output of the previous layer and divide it into smaller regions. In the context of this problem, it takes each of these small regions that consist of groups of pixels and applies a pooling technique. It then outputs a smaller section of the original region. Although this section is smaller in size, by applying the filters to it, the properties in the section are kept and then passed to the network with fewer number of features. There are several techniques to obtain this output, such as max pooling or average pooling. In Figure 2.7 it was applied max pooling to a region, which transforms the selected region of pixels into one single pixel, the pixel with the highest value. This is useful to reduce the number of parameters while maintaining the most important properties and contents of the images. These properties depends on the filter that is applied in the image. Using this method, the most significant features are kept and propagated to the next layer [52].

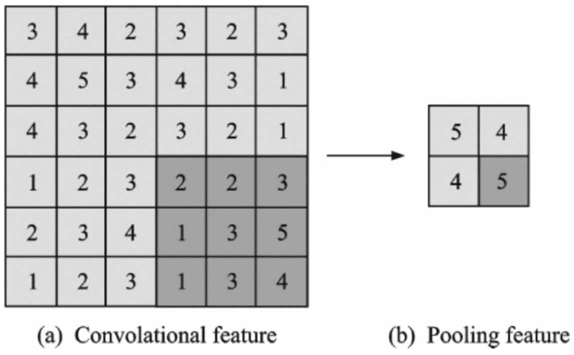


Figure 2.7: Max Pooling execution example applied to a 6x6 matrix [52]

Since this technique has been showing great results in solving image recognition problems, this must was the chosen method to be implemented to solve the identified problem of this thesis. However, to implement this algorithm it is necessary to study some aspects related to it such as variants and optimizations to it. Some of these aspects are the choice of the AF, the architecture of the network, the choice of the optimizations, and others. These elements may produce very different results and they must be used in order to create an algorithm that is optimized. Additionally, this optimization might be different from problem to problem. For example, to create the algorithm to recognize art movements, the optimizations that are done, are expected to be different from the optimizations of the algorithm that is able to distinguish the probable creation decade of the art piece. These differences are directly related to the type of data that is used in these networks.

3

Related Work

Having studied and discussed the theory behind the classifiers, a brief study of algorithms and tools that are currently available was done in this section. This study is helpful in the development of this thesis giving a view of what projects or studies already exists and, therefore, understanding in what way the developed platform can be different from these. Also, this study is a good way to compare the development of the proposed solution algorithm with some generic ML tools for image recognition available to the public and if the employment of these tools can be beneficial in this platform's implementation.

Some work has already been done in the image classification field. Some of it was developed to solve some art recognition problems. Yet, any of it tackled the problem that is proposed in this thesis, which is the need to suggest a wide range of features of one piece of art in a quick and simple way. Most of the existing work, only focus on individual elements of the art pieces, such as artist or the art movement classification. Others can give a wider range of information, but it needs human interaction, therefore it is not an instantaneous interaction to the user, which means that it is not as efficient as the proposed work.

3.1 Art features extraction algorithms

Numerous research have been conducted over the years to address the image classification challenge. Some of these were applied to art classification. Two of these studies deal with the classification of the artistic style or the author of art pieces.

3.1.1 Author Classification

The first algorithm that was discussed is designed to categorize a work of art in accordance with its author. The primary objective of this paper is to outperform the earlier efforts. This procedure is intended to be an alternative to the conventional strategy used by historians. Nitin et al. (2017) researched at several CNN architectures and ways to accomplish the objective outlined in this study [68]. An algorithm that could categorize images by 57 distinct artists was developed using a data set of around 17000 images.

With the help of this study, it was determined that CNNs are an effective technique for artist identification. The top performance network was built on a ResNet-18 network with transfer learning from ImageNet. With this solution, the author was able to produce an IC algorithm that could predict authors with an accuracy of 77.7%. Despite the proposed algorithm was successfully developed, there are certain limitations to it. The data set that was utilized is one aspect that was highlighted. Although the author mentions that the size of the data set utilized was larger than in earlier research, its complexity can be expanded. Not just in terms of data magnitude but also the quantity of possibly identified artists. This aspect and a number of other potential improvements, such as the exploration of different architectures, were used as improvements on the developed work during the planning phase for this thesis.

3.1.2 Art Style Recognition

The following algorithm was created to identify the art style from pictures of works of art. Adrian et al. (2019) created a machine learning technique that can identify the artistic style of a painting by processing a photograph of it [33]. Each classifiable art movement was represented by a set of images dispersed among 25 classes. In this investigation, two different kinds of networks were studied. A residual neural network (ResNet) and an AlexNet network. On the style prediction task, an accuracy of over 62% was achieved. This accuracy gain is mostly attributable to the usage of a residual neural network and the effectiveness of retraining, among all the evaluated networks and improvements. The limited dataset utilized to develop this machine learning algorithm is the primary drawback mentioned by the author. With the goal of developing an improved version of this art style recognition problem, the dataset that was used in this thesis is larger.

3.1.3 Art Recognition

Art Recognition [19] is an Artificial Intelligence (AI) algorithm whose main purpose is to understand and evaluate a piece of art from a picture. This picture does not have to be highly detailed, since it can be taken from a smartphone camera. The algorithm is described as only needing around a set of 200 images to produce accurate results. This is possible since the algorithm analyses the unique features of each artist. “[...] the technology can produce accurate results even for artists who significantly changed their style across their careers as every artist has a unique brushstroke” [69].

This algorithm is based on a Deep Convolutional Neural Network (DCNN). This network is previously trained with a set of original paintings labeled with the corresponding artist. Using very high detailed images, the NN knows with high precision the main characteristics of the labeled artist. Afterward, when executing, the same set of features are analyzed and compared to the previously trained ones.

The main application of this algorithm is to identify forgeries and fake paintings. This method, according to the developers, can produce great results when finding fake paintings. “Probabilities to distinguish original from fake can be higher than 90% depending on the style and artist, and we are continuously working on improving accuracy” [7]. To get these high probability values, it is necessary to have a big database of around 1 million artworks [69].

3.1.4 Art Detective

A simpler but very useful project than can be related to the proposed work in this thesis is “Art Detective”. This mobile application can recognize different art styles giving a user-inputted picture of a painting. The algorithm returns to the user the most probable art movement to which the inputted image painting belongs. To date, this algorithm can classify a painting from five different art styles, including surrealism, expressionism, impressionism, cubism, and pop art [4].

Contrarily to the previously seen projects, this application was developed using “Apple’s Create ML”. This tool uses visual recognition technology to identify the art style of the picture. Just like many others, this ML tool, provides a simple way to create these image recognition solutions. To develop “Art Detective”, it was only used 50 labeled images, across the previously described 5 art movements for training and a set of 10 images for testing. When compared to the previous project, the training data set is much smaller. After executing some tests, the testing set had a performance of around 78% of correct classifications [4]. These results show that using a previously created ML tool, even with a small number of images as a training data set, decent results can be obtained in this art style recognition problem.

3.2 Machine Learning Tools

The proposed classification problem for this thesis, compared to the previously seen project “Art Detective” is more complex. Not being limited to distinguishing between only 5 different art styles nor to only trying to classify the art style to which the art piece corresponds. Therefore, when using a similar technique, it is probable that the intended algorithm produces results with a lower accuracy since there are more variables and complexity to the problem. Despite the usage of these tools not being expected to be the most suitable to solve the proposed problem overall, using them to solve some common image recognition problems, such as object or label recognition, might be helpful in this thesis. This approach avoids the need to create and train a model to identify hundreds or even thousands of classes in an image.

There are already some trained models that accomplish this task in a very optimized way. Some examples of it are the Google Cloud Vision API [67], MobileNetV2 [66] and TensorFlow Hub [27]. All these tools can predict several features from a given image without the need of training any model.

From the mentioned tools, MobileNet and Tensorflow can predict a collection of labels that are found in the images. Despite the fact that it has limited prediction ability and can only extract one kind of information from photos, it is important for obtaining details on the objects or other items included in the images. Both of these tools depend on the machine that uses them for computing. Because of this, the time complexity of these is determined by machine hardware.

The Google Cloud Vision API is the third feature extraction tool indicated. Compared to the other two tools assessed, this one is more advanced because it can extract a wider range of attributes like colors, objects, text, and many more from an image. Another significant point to note is that this tool does not depend on the available hardware power to operate. This is a tool that uses cloud computing systems and is available through an Application Programming Interface (API). This tool was deeply studied in the context of this thesis to further supplement the characteristics that are retrieved using image classification algorithms. This might be used to extract information like labels and colors, adding even more details to assist in the cataloging of works of art.

4

Prepossessing of the Data and Database Creation

In this section of the document a deep description of the data used during the development of this platform is done. For the development of this thesis, the data is one of the most important element, since it is the core of multiple situations, such as the model's training, validation, and also testing process. This data is also a fundamental factor when users interact with the platform. In this chapter there is a description of the source of the data that was used, and what were the implemented modifications to use the data the best way possible across multiple algorithms. Also, it is described and discussed information such as how the data was split to create the algorithms, what were the main faced challenges when using it, and what individual transformations were done to the images.

4.1 Description and Analysis

The data that was used to develop the ML algorithms are images. These files are representations of painting, sculptures, and other types of art. These files are analyzed by the models in the training phase. These images were collected from different sources and originally separated in different ways. This requires the need to create a tool that can reorganize the data sets in multiple ways to be used in different contexts. In this thesis problem, the data must be able to be used in four different algorithms. The author, art movement, creation year and art type recognition algorithms. To be able to use the data in every algorithm development, it must be separated according to each context. Although being separated in different ways, the structure is the same. The data was separated into different directories. Each directory represents a class and they contain all the image files that are classified under them. One of the data sets used is "Best Artwork of All Time" [25]. This data set is available in the Kaggle platform [74]. The data was originally separated by 50 different authors. For this reason, this data set was optimized to develop the author recognition algorithm. This data set cannot be used to train the other proposed algorithms since it lacks information on any other attribute. More data must thus be obtained from additional sources. One other source of data is the “WikiArt” data set [10]. Similar to the previous data set, the classes of this are separated into different directories. The difference from the previous one is that the data available in this data set is separated by the art movement of the pieces. Since a single art piece can be classified under multiple art movements, copies of the same file can be presented in multiple directories. This is one of the challenges that this data set imposed. In this data set each file has information about the art name, author and creation year. With this information, it was possible to reorganize the data set to be used to develop the algorithms to solve some of the proposed algorithms. Finally, the last data set that was used for the development of the proposed algorithms of this thesis is “Nation Gallery’s collection” [43]. This is the most complete one when comparing the three used data sets. With a collection of more than 130 thousand files, it has vast information about every single art piece, such as creation year, type of art, and more features. Combining these data sets from these different sources, there are more than 200 thousand files that were used to create the data sets that were used to train the models for each proposed algorithm of this thesis.

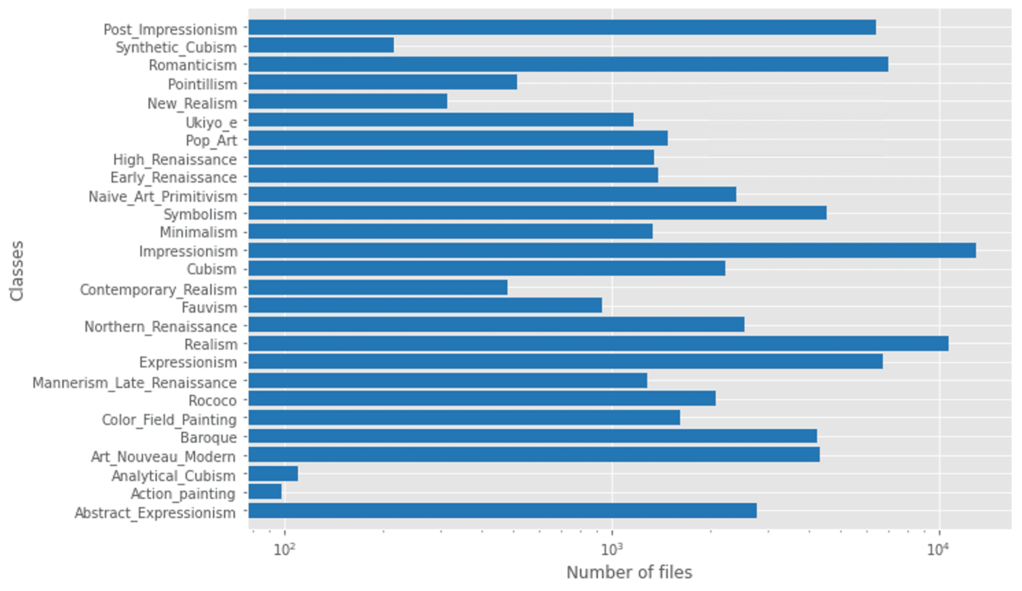


Figure 4.1: WikiArt data set file distribution

Figure 4.1 shows how the “WikiArt” data set is distributed across different art movements. In the diagram, it is represented the number of files in each directory. This data set has a total of more than 81 thousand files distributed by 27 different art movements. These classes are represented by every element on the vertical axis. Being the only data set that contains information about the art movement, this data set is ready to be used in the art movement algorithm. This means that there is no need to do any modifications or optimizations. Similarly to this data set, “Best Artwork of All Time” data set has its data distributed across more classes. This set has data labeled across 50 classes, which means, it has art pieces of 50 different artists. As seen before, these sets have common information. Particularly, they share information about the art piece author. This means that they can be combined to generate an even bigger collection of data. The main advantage of merging these databases is to increase the robustness of the data used to train the models. This translates into better prediction capabilities. The available data only offers information on part of the desired parameters, as was the case with the prior data sets. As a result, the size of the final data sets that were applied to each recognition task is different. One solution that could be used to solve this issue would be to manually label all the images presented in all the data sets. This option is not very reliable, since there is a large number of images in these data sets and it would be a high time-consuming task. Combining these sets from different sources, and removing the duplicated ones, there are more than 200 thousand files that were used to train a model for each proposed solution of this thesis.

4.2 Data Preparation

When working with the data to develop the algorithms, some problems occurred and modifications had to be done to solve them. The most common issue that was presented in the data sets was the existence of corrupted files. Some image files presented in the data sets were corrupted and had to be identified and removed. Using corrupted files presents a problem since the algorithm is not be able to read them and they are not evaluated. Therefore, it was created a script that analyses all the files in the data sets and remove the corrupted ones. Since all the files in data sets would be PNG or JPEG files, the script checked the structure of every file and see if it had a structure of any of these file extensions. One other approach would be to check the file extension value at the end of the file. Although this method could be used, a deeper analysis was required, such as the described one since a file can have a PNG or JPEG extension and still be corrupted.

At this point, the data was separated into four main groups. Each one of these groups represents a data set that could be used to train the models that are able to predict the art movement, the author, the art type, and the creation year. The basis schema of these data sets are a set of directories. Each directory represents a class and inside that directory there is every file that is part of that class. A class can be seen as a label for the corresponding images. For example, in the data set that was used to create the author prediction algorithm, there are directories that represent pieces of art from "Vincent Van Gogh" or "Pablo Picasso". In these directories, there are every gathered art piece image of those artists. After organizing the data, there are four data sets that can be used as input to the four different CNN. However, an individual context analysis must be done according to each problem.

4.2.1 Author Data Set

The data set that was used in the author recognition algorithm has a high number of art pieces by artists. This data set was created by combining the "WikiArt" and the "Best

Artworks of All Time". It comprises around 88 thousand files overall, distributed across 1125 artists. It is also a synonym for a high number of classes with a small number of files in this situation. Although it is possible to use this data as input to a CNN, there must be done an analysis of the data that was previously generated and apply a set of modifications to optimize it.

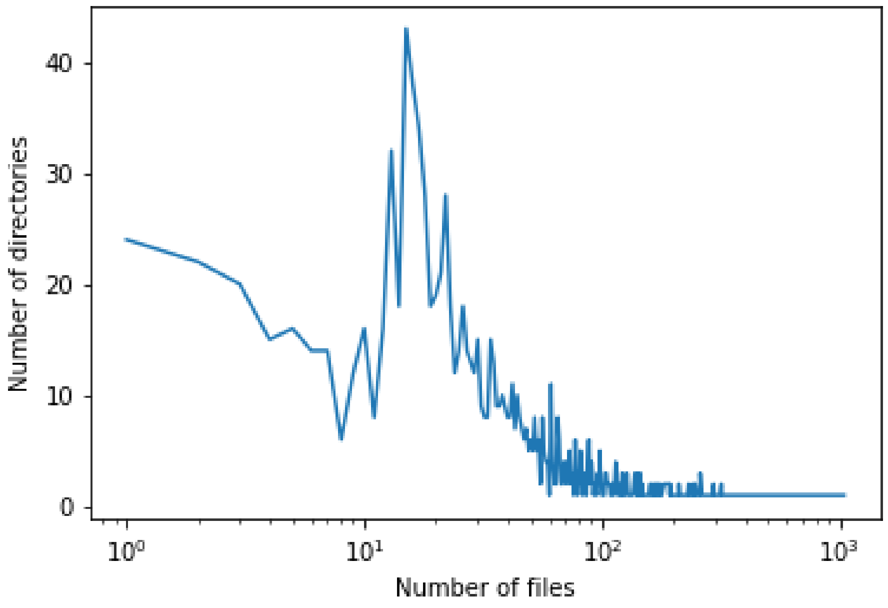


Figure 4.2: Image file distribution of artist data set

Figure 4.2 shows the number of directories with a certain number of files inside them related to the author recognition problem. The horizontal axis shows the number of files and the vertical axis shows the number of directories that have x number of files inside. For example, there are 24 directories with only one image file inside them. Analyzing this graph, it can be said that there is a great number of directories that contains under 100 files. This value is not significantly high to be used in this image classification problem [21]. Therefore, these low file number directories must be discarded for the training phase of the algorithm. In the context of the problem, it was only considered all directories that had more than 250 files per class. This value was chosen because it is an equilibrium between having directories with a high number of files and a decent number of predictable classes. If the value is lower, as seen before, this translates into a situation where there are a lot of directories with few training assets. On the other hand, if the value is higher, it would reduce significantly the number of classes that would be predicted.

Having optimized the data for the author recognition problem, the set had files among 68 classes. This means that the average number of files per directory is now higher. When previously it was an average of 78.6 files per directory, since there were a lot of directories with a low number of images, this value is now 579 files per directory. This is a better value when taking into account that, at the same time, the algorithm is able to make predictions

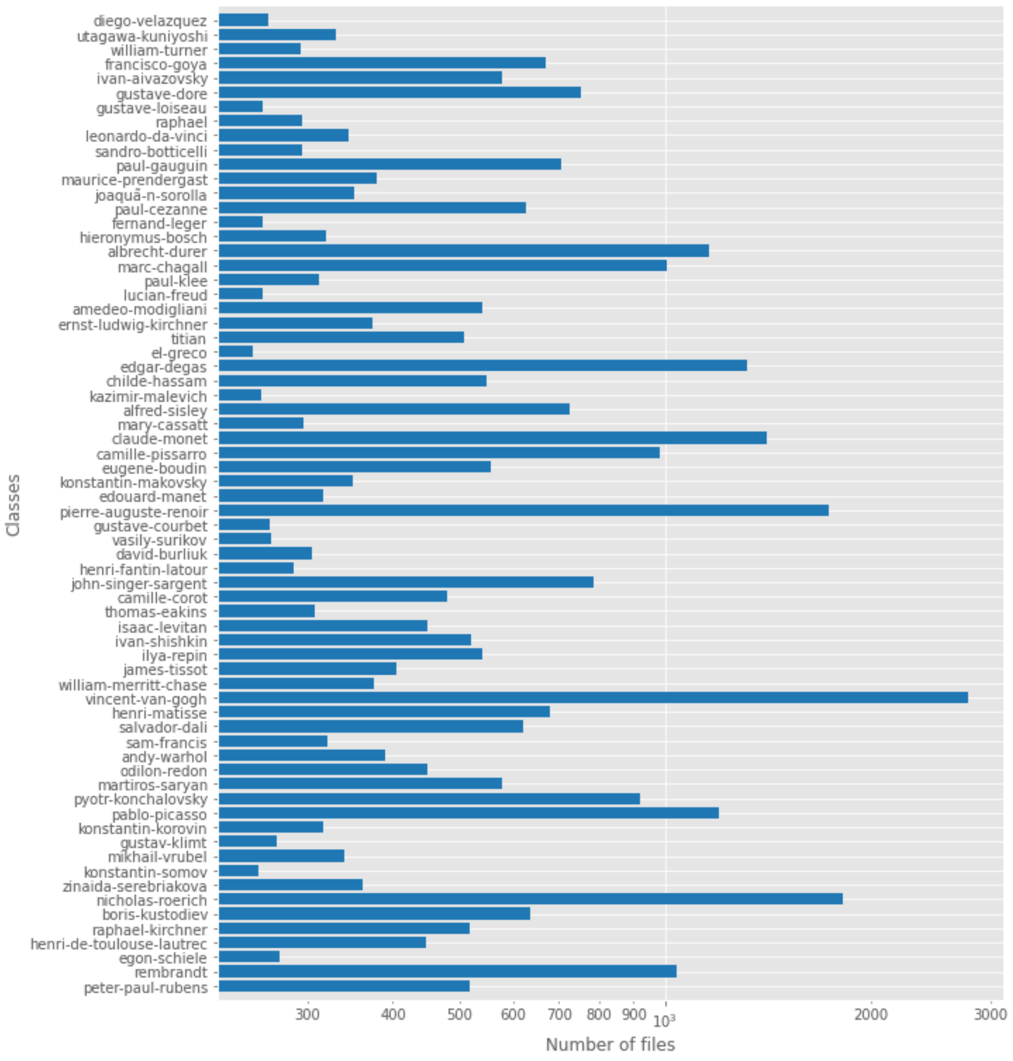


Figure 4.3: Optimized data set for the author recognition algorithm

of 68 different authors. Figure 4.3 shows the file distribution among these 68 classes. As intended, there are no directories with less than 250 files. After these modifications, this is the data set that can be used to train the author recognition problem.

4.2.2 Year Data Set

To generate the year recognition data set, the “WikiArt” data set was used as a starting point. Combining this set with the “Nation Gallery’s collection”, a bigger data set of images was generated. These were the only data sets used since they were the only ones with the creation year information. Although the vast majority of the files presented in these sets have this information, some of the files had no information about the creation year. To handle these cases, it was created an algorithm that, according to the file information, separates the data files by the year ignoring the ones with no information.

Figure 4.4 shows the file distribution in the generated year data set. The horizontal

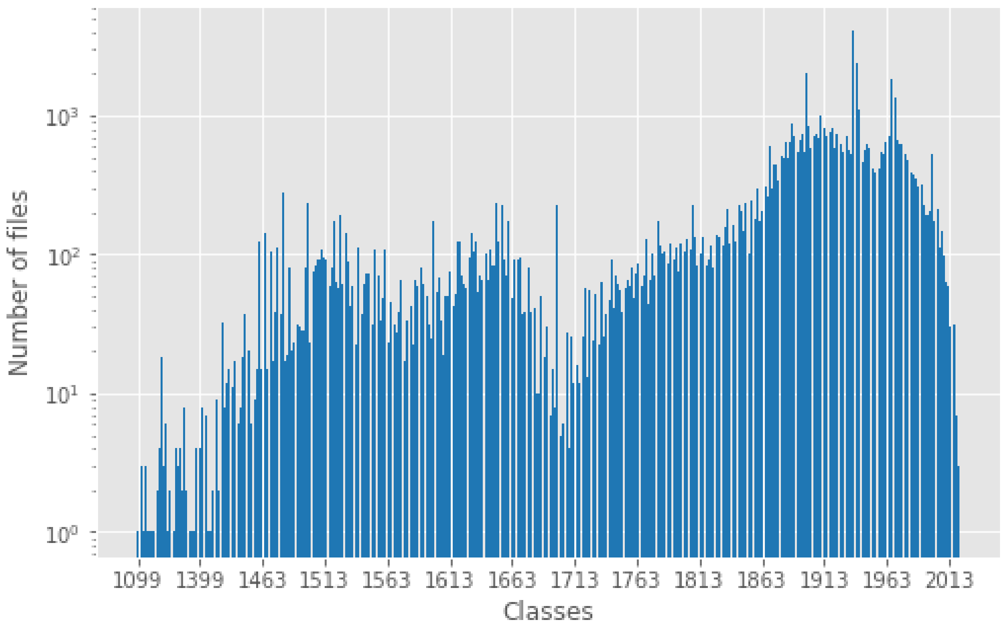


Figure 4.4: File distribution across the years

axis shows the year range of the data available in the data set. The vertical axis shows the number of files contained in the classes. This graph shows that there are a lot of classes that contain under one hundred files, which is a low number of files for the algorithm to learn something. The vast majority of files are condensed between the years 1900 and 2000. To avoid the situation where there are a lot of classes with a low number of files, it was thought to modify the data distribution. Instead of grouping files by the creation year, the data was distributed by the decade of creation. This new modification can be visualized in Figure 4.5.

When grouped by decade, the majority of classes have a higher number of files to be used to train the model. Compared with the year distribution, the majority of classes have now over 100 files, while in the previous situation, only some classes verify this situation. One other observation is that now there are a higher number of classes with over a thousand files. This modification also implies that the algorithm is not able to predict the art piece creation year, but its creation decade.

4.2.3 Art Type Data Set

The data that was used for the art type recognition was only present in the “Nation Gallery’s collection” data set. Therefore, there was no need to merge it with any other data set. This data set is mostly ready to be used directly in the type prediction model. Since there were two classes with a low number of files, the files from these classes were discarded. It corresponds to the classes “Technical Material” and “Time-Based Media Art”. These classes had under 200 files. Figure 4.6 shows, after the described optimization, that this set has its data files well distributed across the classes. This means that there is no

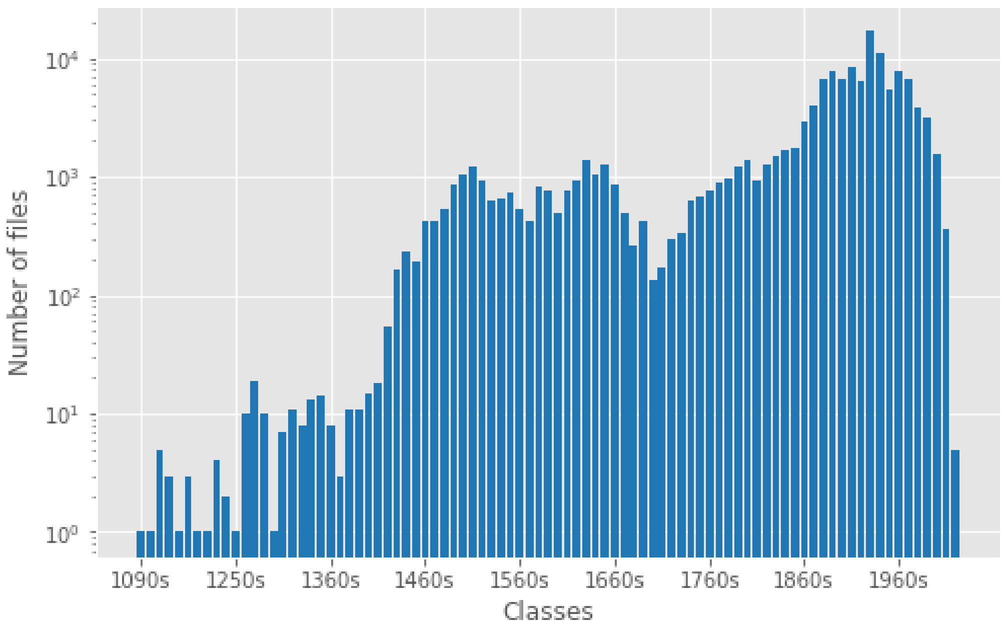


Figure 4.5: Number of files in the Year data set distributed by decades

need to do any more data transformation or optimization.

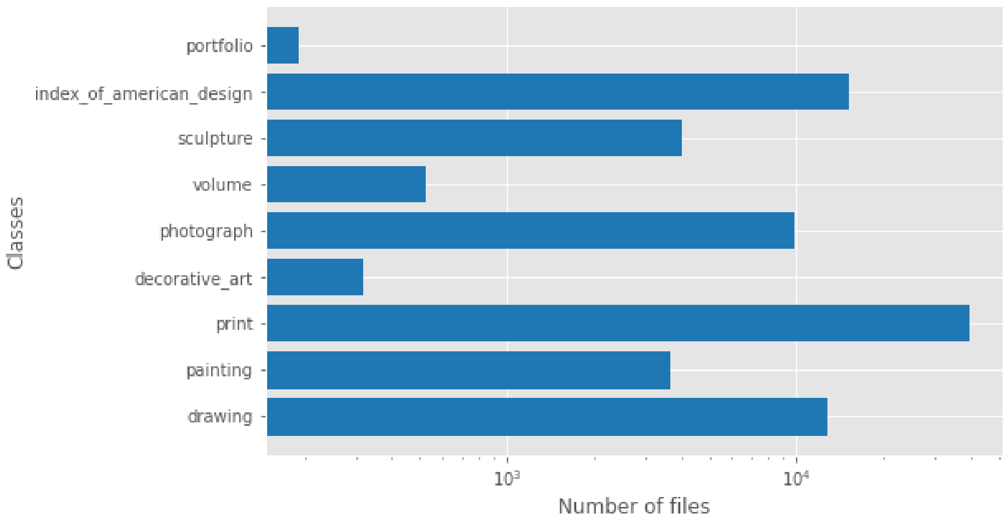


Figure 4.6: Art type data set distribution

This data set is used to train the art type prediction algorithm. It is able to classify images between nine different classes, corresponding to nine different art types, such as paintings, drawings, sculptures, and more.

4.2.4 Final Data

Having the data optimized and separated according to the different algorithms, a visualization of the data can be done to better understand its distribution according to each context. Figure 4.7 represents in a simple way the distribution of the merged data sets into the different algorithms. It shows the number of files that can be used to train each classification algorithm. For each one of them, it can only be used data that is accordingly labeled to that specific context.

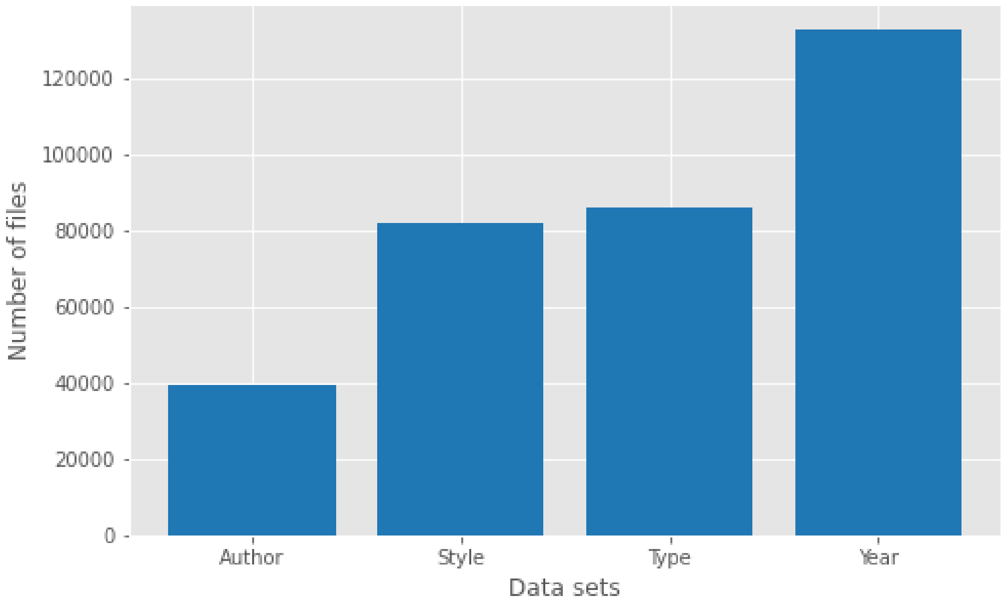


Figure 4.7: File distribution by data set

The fact that there is a different number of image files per problem can be explained due to the incomplete labeled data across the data sources. Not every data set that was used had labeled images for the four proposed problems. As seen in the diagram, the data that was labeled according to the year of the art pieces is the largest. This can be explained since it is one of the features that is labeled across the bigger data sets. On the other hand, the art author data is the smaller data set due to the optimizations that were done. In this data set, as explained, the majority of the images were discarded since there were a vast amount of classes with under 250 files.

As seen before, the data used for the development of the proposed models is a set of four file data sets distributed according to each one of the four algorithms. Although this file structure can be used to correctly train the models, some issues are raised with it. Here, a brief explanation of these issues and the proposed solution to avoid them is conducted.

In the process of the development of the algorithms, the data is used just for the training and validation phase. As soon as these ML models are developed and ready to be used in real-world situations, this data is not needed anymore. In the context of this thesis, the data is used to be displayed to the user, therefore, the studied structure is not mandatory in this future phase. The other issue is that for the developed models, as previously discussed, there must be four different data sets labeled according to each model goal. This situation implies that there could be multiple copies of the same file in the different data sets, which increases the space complexity of the data.

4.3 Database

To optimize the data structure and to avoid the need to store unnecessary or duplicated data, a single database was created. This was used not only stores every image file, but also all the required information of each file, such as the author, the style, or any other attribute. All images were stored in this database, even the ones that were not used for the training process of any algorithm. This database also stores other information for the correct functionality of the platform, such as other tables for the users and possible corrections that these users can create. In this database, every entry of the pieces of art table corresponds to a different image file, solving the duplicated issue. Also, this table can then be used to provide all the required data to train each individual model. For example, from the four proposed models, if it is needed to train the author prediction model, the data that has information on this attribute is fetched and distributed by the corresponding author. At this point, the structure is the same as the one that was previously discussed. Therefore, this data can be used to train the model, and then discarded, once the model is trained. This can be done since the labeled data is still available in the main database.

This database table contains the information of every art piece. Specifically, it stores the information about every parameter that is expected to be predicted through the created models and the information that is displayed to the users through the interface. Furthermore, this information can also be used to filter the data when fetched by the user. This information is the name, set of authors, styles, colors, and more. As previously explained, when the users submit corrections to a set of suggestions, the data is saved to be used to retrain the prediction models. Therefore, there must be a way to distinguish these submitted images, that were not used to train the models, from all the other images that were already used to train the models. For this reason, there is a parameter in each entry that describes this exact situation. Finally, for each entry, there is a URL to the actual image file. This means that every image file is stored outside this database in an optimized file storage tool. An example of an entry of this table is shown below:

```
{
    "art_uuid":"",
    "art_file":"joseph-wright_benjamin-franklin-1782",
    "art_name":"benjamin-franklin-1782",
    "art_authors":["joseph-wright"],
    "art_styles":["Baroque", "Romanticism"],
    "art_colors":["E7BDA3", "DBBDA2", "FBE2CC", "BE9678", "FCD5C0"],
    "art_tags":["Person", "Clothing"],
    "art_year":"1782",
    "art_materials":["oil"],
    "art_description": "",
    "art_type":"",
    "art_url":"",
    "art_approved": 1,
    "art_created_at": 1651528644.406398,
    "art_created_by": ""
}
```

In this example, it is possible to see all the parameters that are stored in the art pieces table. All these parameters can be edited by administration operations or suggested by regular users. This is a feature that helps with the correct cataloging of pieces of art, since the vast majority of them, so far, are automatically generated and can have wrong or badly formatted, as can be seen in the entry related to the artwork's name. Each entry of the database have the following information:

- uuid - this is a unique value that is automatically generated for each uploaded entry to the database. This value is not displayed in the user interface since it is only be used by the system to identify the pieces of art.
- file - this entry value shows the original file name. This parameter is not displayed to the user. It is meant to be used to edit other parameters that are not correctly formatted, since it may contain information that is not available in the other fields. This can also be used to correct any information.
- name - corresponds to the art piece name. An art piece can then be searched by this parameter as well as others. This information can be displayed on the user interface, therefore, must be correctly formatted before showing it.
- authors - set of the art piece authors. can have multiple elements. This information is displayed to the user. These names are generated throw the author algorithm or by user correction.
- styles - holds the information about the piece's art movements. Can hold multiple values that are generated by the algorithm or the user input.
- colors - most dominant colors that can be identified in the art piece. These values are automatically generated.
- tags - set of labels that the algorithm can identify in the image file. These values may correspond to objects or other tags that are identified in the image.
- year - generated by the algorithm or throw user input, corresponds to the creation year of the art piece. This value can also contain a decade value.

- materials - list of materials that were used when creating the piece of art. When submitting an art piece, these values can be generated automatically or provided by the user.
- description - a thorough explanation that highlights each features of the referenced work of art. When uploading a new artwork, the user can include this information.
- type - this value is generated by the art type algorithm or provided by the user. Corresponds to the type of the art piece, such as painting, sculpture, and others.
- url - url of the image file that is stored in a storage server. This file is then used to be displayed to the users and can also be transformed to be trained by the models.
- approved - this parameter can store a value of 0 or 1. It shows whether the image file was already approved by a reviewer user (value 1) or not (value 0). This helps to keep track of which images were submitted or the ones that are already in the database with the correct information. Regularly, once these images reach a certain value, all the models can be retrained.
- created_at - each object stores a timestamp of the moment that it was submitted to the platform.
- created_by - similarly to the previous parameter, this value stores the information of the user that uploaded the information of the piece of art.

In addition to this table, a simple user table that stores each user's information was also created. Initially, this table only stores basic information, such as a user id and name. In the future, this table can store more information such as a list of favorite art pieces.

One functionality that was thought to be developed is the possibility for the users to submit a change to an existing piece of art. To keep track of these suggestions, it was thought to create a table in the existing database that stores information about each suggestion made by users. Each entry of this database stores information about the user who made the suggestion, in which art piece was made the suggestion, the parameter, and the suggestion itself. More information can also be stored, such as the suggestion time instant and a unique id for each entry of this database table. An example of an entry of this table is displayed below:

```
{
    "suggestion_uuid":"",
    "art_uuid":"",
    "user_uuid":"",
    "suggestion_parameter":"art_author",
    "suggestion":"suggested_new_author"
    "suggestion_created_at":"1651531353.758255"
}
```

Combining all tables, it is possible to create queries that give all the suggestions created for a piece of art or other queries such as all the suggestions that a user made.

Having all the data ready to be used to train the proposed models, it must be separated into two different sets, the training set, and the validation set. Each one of these sets is used in different situations across the development of the algorithms. For the four data sets, it was chosen the same split ratio, 80% for the training set and 20% for the validation set.

5 Development of Machine Learning Algorithms for Art Classification

In this chapter, a deep explanation of the approach when developing the proposed work for this thesis is done. Here, it is also described and discussed all the required steps for its implementation. The initial approach that was used to create an image feature extraction tool is to start by implementing an algorithm that can predict one of the proposed features. Afterward, this algorithm was adjusted and expanded to be used to solve the other proposed problems. The previous chapter shows that the author's data set is the smallest among the four data sets. For this reason, the training process is expected to be less time-consuming, since it is directly related to the size of the data set. As a result, it is quicker to study some generic aspects that can be used over to the other algorithms mentioned. When conducting this first research on the author recognition problem is quicker, using any other algorithm would require more time for the research process. Therefore, the author's recognition problem was the first to be studied and then used to implement all the other three proposed algorithms.

For the development of the proposed platform, the base programming language that was used is Python. Python is one of the most known and used programming languages for computer science [37]. It is known by its high-level interactive nature and its vast offer of libraries [45]. Among these libraries there is TensorFlow. TensorFlow is an open source ML system that is continuously being optimized to solve problems with its main focus on training and inference on DNN in a great variety of environments. TensorFlow uses powerful computing in order to make heavy computations as fast as possible, especially training, since in a majority of cases, it is the heaviest computation operation required in these types of problems [1]. Since these computations can be done in different powerful machines, it offers flexibility when implementing solutions for classification problems. Some of them are the possibility to choose between different optimizations or different training methods.

NN related problems require the need to use parallelism computation. For this reason, common Graphical Processing Unit (GPU)s can be used to solve these problems. This is explained since GPUs are designed for high-performance rendering and repeated operations since they are optimized for parallelism and rely on more pipelined than general purpose GPUs. Therefore, a GPU can produce a better performance than a CPU in these IR problems that use NN architectures [42]. For the development of these algorithms, multiple machines were used to test the training process. When choosing different machines for testing, the GPU process power was taken into account. Finally, the most efficient one was chosen. For the vast majority of the computation, it was used a GeForce GTX 1060 GPU with 6GB of internal memory. With this GPU, the training process is much faster when compared with the same training process done in a MacBook Air with an M1 chip. It was verified that, when executing the same training process in both machines, the GeForce GTX 1060 GPU finished it in less than half the time of the MacBook.

5.1 Model architecture

As previously stated in subsection 2.2.4, CNN is the classifier that was employed in the creation of the four suggested IC algorithms. However, there are a variety of architectures that might be used with this strategy [26]. This section describes the research that was conducted to determine the optimal architecture to employ for the author recognition issue. The architecture of the network is not only the structure of the layers of the network, but also the behaviour between them. To find the best architecture to be used in these art feature extraction models, different architectures were studied and tested. Several research have been conducted in order to determine the best architectures to each specific context. Some of these have demonstrated significant advancements in the field of image classification, such as the Xception [50] [12] and the ResNet50 [49] architectures. The context of the problem has a significant impact on these architectures' performance. For this reason, they must be tested for the author recognition problem. Alongside this comparison, a custom designed architecture was also tested. After this process, their results were discussed. For every network, all the images that were used into the models have a shape of 224x224x3 values. To achieve this resolution, each image was cropped into a square shape of 224 pixels by 224 pixels. All the tests were done under the same circumstances and with the same data set. This provides the most comparable results, which helps to decide which architecture produces the best results. Although some parameters were chosen, such as the resizing method and the image size, these parameters were be optimized in further sections of this document.

5.1.1 Custom Network

This researching phase started with a simple CNN architecture. This architecture is composed of four convolution layers paired with a max-pooling layer. Each layer has a different number of generated filters. The first layer generates 32 filters, while the next layers generate 64, 128, and 256 filters respectively. This provides the possibility for deeper layers to extract more complex features from the input images. At the end of this set of layers, this architecture has a traditional NN architecture with two layers of 1024 neurons. Finally, the output of this network is a layer with the same number of neurons as the number of predictable classes (68 classes). This last layer represents the probability of the input image being part of the respective class. Despite its simplicity, this network is very robust in terms of the number of layers and the number of neurons per layer. Although its complexity, it is not expected to produce the best results when compared with the other networks that were studied. The main goal of this network is to create a foundation model that can be compared to more complex and reliable models.

5.1.2 Xception

Introduced by François Chollet [11], the Xception is a model architecture that is vastly used in image classification problems. It was used to determine the best NN architecture for this problem, among the tested ones. The idea of Depthwise Separable Convolutions is used in this model. These are improvements to regular convolutions that should increase efficiency in terms of computation time [46]. Pointwise convolution is performed first, followed by depthwise convolution. While in depthwise convolution, each filter processes only one channel of the input image independently, in pointwise convolution the 1x1 dimensional filter iterates every single point of the input image [46]. In Xception architecture, there are three main blocks. The data entry flow, the first one, where the data go through. The data then passes through the middle flow, where the computation is repeated eight times before passing through the exit flow [50]. There are in total 14 modules (4, 8, and 2 modules respectively). It consists of 36 convolutional layers in total. Except for the first module of the first structure and the last module of the exit flow [46], every module has residual connections.

5.1.3 ResNet50

The last architecture that was tested for the author recognition problem is ResNet50. This network has been used to solve a wide range of image recognition problems with excellent results. The existence of shortcut connections is the main difference between this type of network and others. These connections allow one or more layers to be skipped. Because deeper networks in common CNN are difficult to train, due to the vanishing gradients problem, these connections allow deeper layers of the network to be trained only when they are truly needed, rather than saturating the network [5]. ResNet50 is a network architecture with fifty layers, as its name suggests. There are other ResNet network architectures with a higher or lower number of layers. To adapt this network to this author recognition problem, it was added two layers of 1024 neurons each at the end of the network. A layer with 68 neurons was also added, representing the number of classes to be predicted.

5.1.4 Comparative Analysis

After analyzing some of the existing network architectures, they were used to train the author recognition algorithm alongside the custom network that was previously designed. These models were put to the test under the same conditions so that the results could be compared with as much confidence as possible. Figure 5.1 shows the outcomes of the author's model training process. The training and validation accuracy value progression for the three distinct tested architectures can be seen in this figure over the training epochs. One thing to note is that the accuracy value increases over time. This means that when the number of epochs increases, the accuracy value tends to rise to a certain number of epochs. This fact can also be seen throughout all the tested architectures, both training and validation accuracy eventually settle. This can be interpreted as each model maximal capabilities, under the provided conditions. When the accuracy values of the three analyzed models are compared, it is not easy to determine which model is the best right away. When the training accuracy data are analyzed, the custom-designed model delivers excellent results of roughly 99 percent. In comparison to the other evaluated models, the Xception achieves a value of less than 96 percent, while the ResNet50 architecture achieves a greater value (a little more than 97%). With this knowledge, it would be possible to conclude that custom architecture is the best option. Despite having the best training accuracy values, the results of the validation accuracy analysis differ from the training accuracy analysis. Among the three studied networks, the ResNet50 network exhibited the highest validation accuracy scores (59 percent). In contrast to what was observed with training accuracy values, the custom model produced the lowest validation accuracy values. This is explained by the fact that this design is not efficient for identifying a generic formula capable of predicting both training and validation data. This suggests that the model is incredibly efficient at interpreting the images it already knows, but it is inefficient at predicting new ones. This is known as overfitting [47]. Although this effect is visible in all three evaluated models, the custom-designed architecture has the strongest effect. As a result, this model should not be used to create the IC algorithms. In contrast, the ResNet50 was selected for the development of this algorithm because it achieved the highest validation accuracy scores. The validation accuracy, out of the two produced scores, is the one that most accurately depicts the model's ability to predict occurrences in actual circumstances. For this reason, it is the metric that is most taken into account when comparing results.

In this example, the majority of the model's accuracies stopped changing around the 30th epoch. The training process can be simplified by minimizing the number of epochs. At the same time, lowering this quantity has no significant impact on prediction accuracy values. These accuracy values can also be analyzed in Table 5.1. The maximum values of the training and validation accuracies are presented in this table and can be compared. Along with this information, the accuracy values of the top three and top five prediction capabilities of each evaluated model are presented.

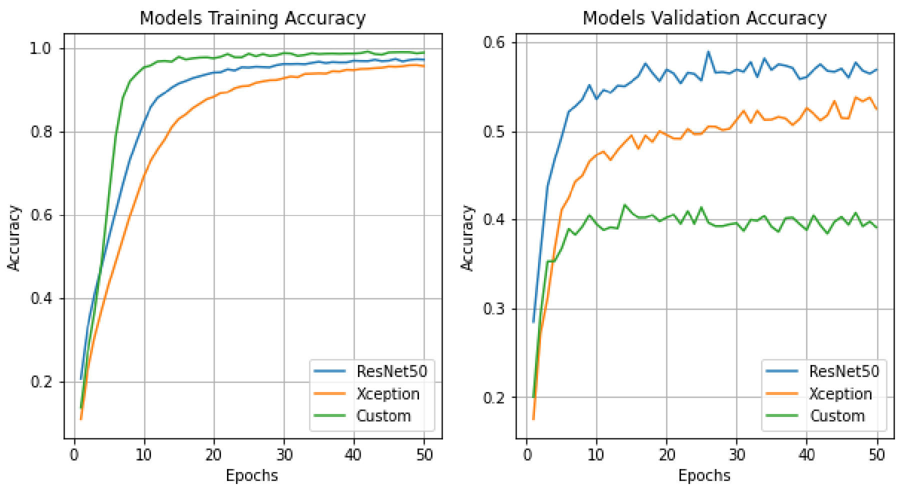


Figure 5.1: CNN architecture training comparison applied to the author recognition algorithm

Based on the analysis of these results, among the tested models, the ResNet50 architecture is the best choice for solving the author recognition problem. Moreover, because the other three proposed challenges (art style, decade, and type prediction) are similar, since they rely on the same type of data, and since a deeper analysis regarding each one of the proposed algorithms is a time consuming task, this is the architecture structure that was used as a starting point for the other three algorithms as well. Despite the fact that this model is the best among the evaluated ones for addressing this specific problem, there are numerous modifications that can be performed to increase the accuracy rates, particularly the validation accuracy, which, as can be seen, is significantly lower than the training accuracy. Transfer learning is one of the most frequently applied strategies in the development of IR models. This technique is discussed in detail in the following section.

5.1.5 ResNet50 With Transfer Learning

Transfer learning is a technique that can be used in conjunction with the CNN architecture. This approach relies on loading the model with some previous knowledge regarding one context. This knowledge is the result of a training process that was executed from external sources and under a specific context. This technique is used to improve a wide range of IC problems and has become very popular among the community because it has shown high prediction improvements even with a small number of samples [50]. For the developed algorithms, the transferred information was gathered from a model that was trained with the data set “ImageNet”. ImageNet is a database with more than 14 million labeled images divided into thousands of categories [28]. The main advantages of applying this technique, is the reduction of the training time, since there is no need to retrain the layers that contain the previous acquired knowledge. One other advantage is the increase of the prediction accuracy of the model. This technology was applied to the previously studied ResNet50 architecture.

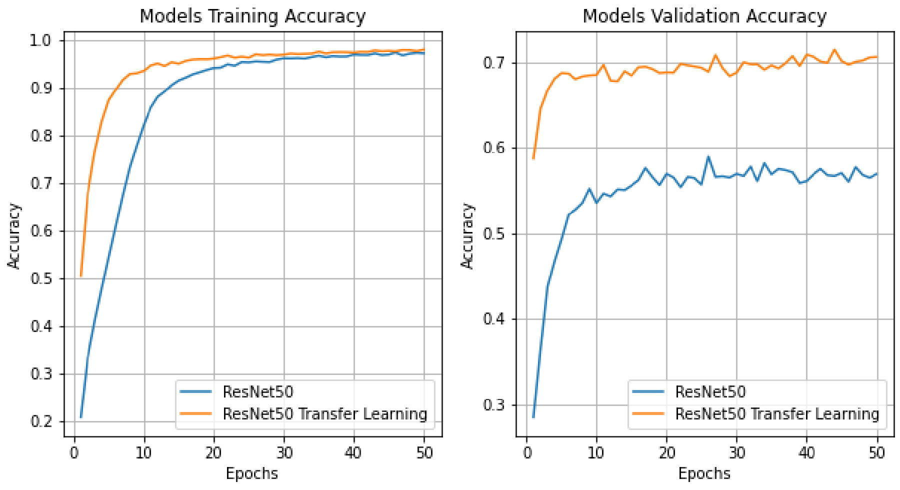


Figure 5.2: Comparison of the accuracy values between ResNet50 with and without Transfer Learning Technique

To include transfer learning in the ResNet50 model, no changes were made to the existing architecture. Only the ImageNet knowledge needs to be loaded into the existing model. The results of training the new model are shown in Figure 5.2. Here it can be seen the comparison of previously obtained accuracy values with the results by applying the transfer learning to the same model. As a result of using this technique, the accuracy values improved significantly, particularly in the validation accuracy values. Previously, the maximum value was about 59%. However, using transfer learning, the top validation accuracy value is now 71%. This effect can also be seen in training accuracy but in a smaller scale. Analyzing the accuracy curves also reveals that the accuracy values stabilize significantly sooner than previously computed accuracy values. This can be explained by the fact that the model does not require as much training as the prior model did because the majority of the knowledge already exists. Another thing to note is that when employing transfer learning, the accuracy value variation is minimal because the accuracy results are higher from the start. When transfer learning is not implemented in the first tested ResNet50 network, the accuracy values gradually increase over time. This impact demonstrates how efficiently using previously learned knowledge helps the model perform better when predicting images in this scenario.

	Training Accuracy			Validation Accuracy		
Architecture	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
Custom Network	**0.9909**	**0.9995**	**0.9999**	0.4169	0.5681	0.6540
Xception	0.9587	0.9962	0.9995	0.5377	0.7134	0.7998
ResNet50	0.9732	0.9975	0.9995	0.5895	0.7616	0.8225
ResNet50 Transfer Learning	0.9794	0.9986	0.9998	**0.7147**	**0.8534**	**0.8995**

Table 5.1: Training and Validation accuracies for four different CNN architectures

Table 5.1 can be used to perform a more in-depth analysis of the produced accuracy values. This table contains information not only about the accuracy of the first predicted value but also about the three and five first classes predicted by the classifiers. These values represent the likelihood of the algorithm successfully identifying an image's class in the first n output classes. Observing the results, the ResNet50 with transfer learning obtained the highest values for the validation accuracy metric. It not only outperforms the other architectures in the top one accuracy, as previously discussed, but also in the top three and five predicted classifications, with substantially higher accuracy values. This effect is more noticeable in the validation values, but it may also be seen in the training accuracy values, discarding the values produced by the custom network, which as seen, suffers from overfitting. It is possible to conclude that this optimization is a valuable technique that should be applied in all of the other proposed problems for this thesis.

Despite the fact that transfer learning considerably enhances ResNet50 network performance in this author recognition problem, there are various more strategies that can be used to boost prediction accuracy even further. This thesis mostly addresses two types of optimizations. Optimization of data and hyper-parameter tweaking. The first one relates directly to the data used to train the models. Some changes to the data were performed to see if it is possible to modify the algorithm's prediction skills. To complete this, using hyper-parameter optimization, various parameter modifications were performed to determine what is the optimal combination that results in the best accuracy values. These adjustments are presented and discussed in further detail in the following sections.

5.2 Data Optimization

After determining which CNN architecture would be adopted as a starting point for the suggested algorithms in this thesis, certain improvements can be made to boost the algorithm's prediction capabilities even further. Minimizing the number of features in the images used in these algorithms enhances training time performance. This suggests that the time it takes the model to complete the training phase is proportional to the number of features in the data. At the same time, these characteristics must be sufficient to produce high accuracy values. Multiple strategies were explored in this section to establish this equilibrium. This section looks at optimizations that are directly applicable to the image files. There were investigated three variations of data variables. The resizing process, the image size, and finally, image content modifications such as image transformations. The models were tested and compared for each of these trials. Following a quick evaluation, the best enhancements for the author recognition problem was selected.

5.2.1 Resizing Method

To be interpreted by the algorithm, the image resolution must be consistent throughout the data set of images. Resizing all images to a squared shape is the simplest way to represent them uniformly. Images were reduced to a square shape for each proposed algorithm. As a result, their width and height have the same value. This situation arises one problem. When transforming an image to a square shape, the contents of the image may be transformed. There are primarily three solutions to this issue.

On one hand, the image could be modified to fit the new size, which means that, in some situations, the image had to be stretched or shrunk to fit the new size. On the other hand, the image could be cropped to the new width and height values. In both these situations, some features would be lost. In the first solution, the contents of the image may lose their shape and identity. For example, if in an original painting there is represented a circle, in a newly resized image, applying the first presented solution, this circle may be represented as an oval shape. Analyzing the second solution, some contents of the image might not even be presented in the final image. This means that some features would be completely lost and never seen by the algorithms. Finally, the third solution for this image representation problem, as suggested in "Deep Learning Approaches to Art Style Recognition in Digital Images" [29], a hybrid solution can be used. With this approach, the image's shape is preserved and all the contents are visible. In this way, when an image has a rectangular shape, its highest size is resized to fit the square and the other size is re-scaled in order to keep the aspect ratio of the original image. With this solution, some images contain black bars. Although, since there is a large amount of data, it is expected that the ML algorithm interprets these black bars as noise and ignores it.

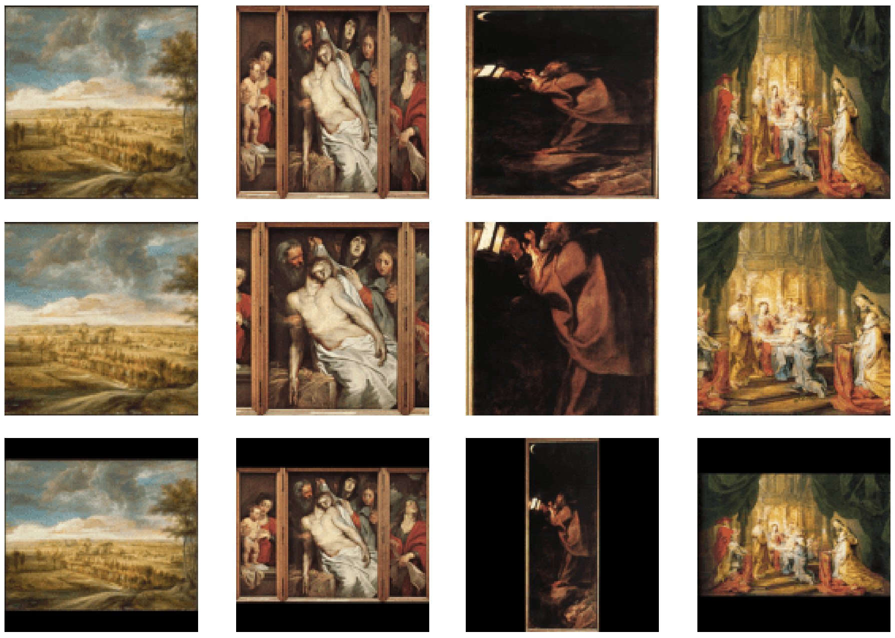


Figure 5.3: Comparison between resizing techniques applied to four art pieces. From left to right in every row: "Landscape with an Avenue of Trees" by Peter Paul Rubens in 1640; "Lamentation of Christ" by Peter Paul Rubens in 1618; "Descent from the Cross" by Peter Paul Rubens in 1614; "The Vision of St Ildefonso" by Peter Paul Rubens in 1632

To better see these image resizing approaches, Figure 5.3 shows a set of transformed images with all described solutions. For the first four images, it was applied the resizing technique. In these images, it can be seen that the contents of the image were distorted to fit all the contents into the squared shape. The next four images represent the same images when the cropping technique is applied. Although some contents are missing, such as both lateral panels in the second image, these images reproduce the original shape of the contents, which does not happen with the first four images, and do not contain any noise, which happens to the last four images. Lastly, there is represented the hybrid solution. As expected, these images preserve not only the shape of the original images but also their contents. On the other hand, these images can contain black bars that are expected to be interpreted as noise in the training phase of the algorithms.

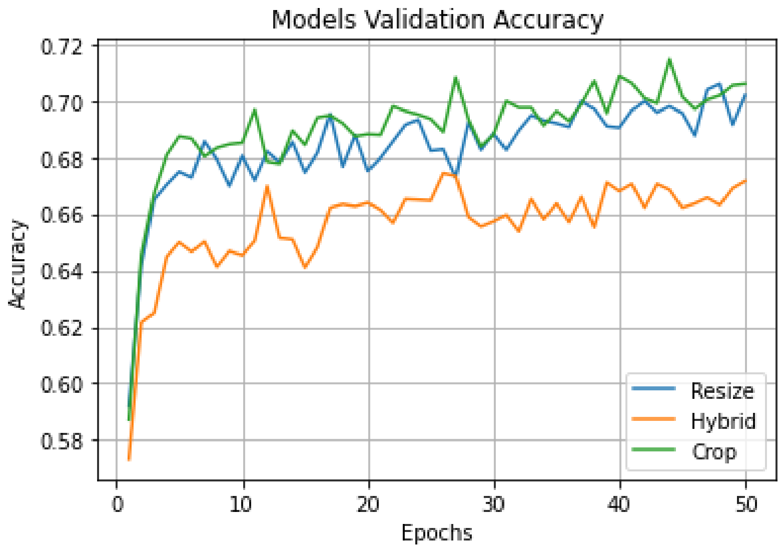


Figure 5.4: Validation accuracy curves from different resizing approaches

The accuracy scores are displayed in Figure 5.4 after applying these image modifications to the author algorithm. The validation accuracy value growth over time for the three researched approaches may be seen in the diagram. According to the results, the cropping technique provides the best prediction results to the tested model. When compared to the second best technique, which achieved 70.6 percent accuracy, this picture resizing technique improves accuracy by one percent. This enhancement is seen throughout the vast majority of the evaluated epochs. As a result this technique was applied to train all the suggested models.

5.2.2 Image Size

Following the optimization of the resizing approach in the previous subsection, it was necessary to research the optimal image size that may be applied in the images used to train these algorithms. The standard input image size for the ResNet architecture is 224 by 224 pixels [33]. So far, this had been the value used when running the preceding tests. In this step, the image size was modified using the same network and variables, and the results were compared. It was evaluated with 112, 224, 336, 448, and 560 values. Because the ResNet50 architecture's default image resolution is 224 pixels, the tested values are multiples of it.

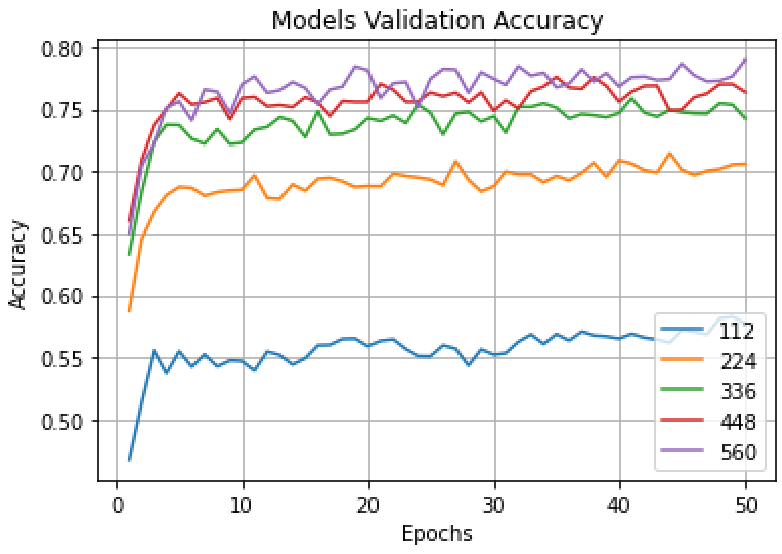


Figure 5.5: Comparison of validation accuracies using different image resolutions to train the models

The accuracy values obtained after training the ResNet50 with the preceding optimizations are shown in Figure 5.5. Among the tests performed in this subsection, only the image resolution was altered. Based on this, it can be concluded that the photos with a resolution of 112 by 112 pixels performed the worst accuracy values. This is understandable given the difficulty of extracting enough information from a low-resolution image to train a model. On the other hand, this was the best time performance training execution. This execution took 120 seconds on average per epoch. Table 5.2 contains a detailed comparison of these values.

		Validation Accuracy		
Image Resolution	Time/epoch	Top-1	Top-3	Top-5
112 x 112	**120 s**	0.5829	0.7504	0.8236
224 x 224	160 s	0.7147	0.8534	0.8995
336 x 336	390 s	0.7588	0.8809	0.9207
448 x 448	530 s	0.7762	0.8992	0.9366
560 x 560	830 s	**0.7896**	**0.9041**	**0.9396**

Table 5.2: Top validation accuracies and average time per epoch of different tested image resolutions

In opposition to the 112 by 112 pixels resolution, when the model is trained using images with a resolution of 560 pixels, not only is the time required to process each epoch substantially longer, but the validation accuracies are also higher. Both of these outcomes can be explained by the fact that the given photos include a vastly larger number of features. In fact, images with a resolution of 560 pixels are 25 times larger than images with 112 pixels resolution.

When compared to all other evaluated resolutions, the tests with 448 and 560 resolutions achieved the highest performance values. Despite the high accuracies values obtained by the 224 and 336 resolution images, these prediction results are significantly outperformed by the other two models, with a difference of more than 7 percent in Top-1 performance in some cases. When comparing the best two performed models, the main difference is the time performance. When processing data with a resolution of 448 by 448 pixels, the time performance is better to that of the 560 resolution photos, requiring less time to complete the training process. On the other hand, the accuracy performance obtained with the 560 resolution data is superior, which justifies the longer time required to analyze each epoch and, therefore, the entire training procedure execution. Using a lower resolution, it is possible to attain almost the same accuracy results as when using 560 as the resolution for the data. As a result, since with a lower image resolution similar validation accuracy values may be achieved with a significantly quicker training period, the images fed into the model have a resolution of 448 by 448 pixels.

5.2.3 Data Augmentation

Data augmentation is a technique that is frequently utilized when developing an IC algorithm. The data augmentation principle involves applying a set of visual changes to the training data. These modifications provide additional information for the model to learn from. Rotations and zooms are two examples of these transformations. This can be useful at times because it simulates real-world situations. In this context, the users are expected to upload photographs that can have some of these alterations, such as a rotated images or ones that have been zoomed.

Taking this into consideration, this data augmentation technique was applied in this part to determine whether it has a good impact on the accuracy. It is believed that by using this strategy, a model can be created that can generalize better with images that it has never seen before because it is, theoretically, ready to understand images with visual transformations. Despite this, some tangible tests were carried out, followed by a thorough analysis to determine the true consequences of this technique.

The data augmentation transformations that were implemented for this model are random flip, random zoom, and random rotation. An image can be flipped horizontally using the random flip transformation. It was only chosen to be horizontally flipped because a vertical flip would more likely compromise the essence of the contents of the art piece. Second, the images are randomly zoomed. An image may be gently zoomed in with this technique. This is meant to mimic zoomed-in photographs provided by users. After that, a random rotation transformation was used. This transformation rotates an image at random to a maximum of $0.2 * 2 * \pi$ radians.

Figure 5.6 illustrates multiple examples that can be obtained by applying the various

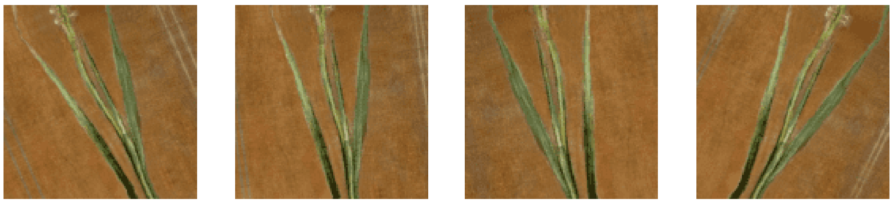


Figure 5.6: Image example transformed with simple data augmentation applied. "Gladiolus" by Claude Monet in 1881

data augmentation methodologies discussed previously. As can be seen, the contents of the images are aesthetically intact despite their varied representations, which is thought to improve the model's prediction capabilities. Despite the visual transformations attempts to offer the model a more realistic perspective, there is one typical case that is not handled by this collection of data augmentation modifications. It is also referred to as occlusion. As suggested in [17], it is critical to prepare the model to be used in the real world, where it may meet occlusions. This means that additional objects may obscure some sections of the intended object. As a result, the model must be prepared to make decisions in scenarios like these. The major goal of this strategy is to improve the robustness and overall performance of the CNN even more. To replicate these conditions, an image can have a black rectangle anywhere in the image with a probability of 25 percent. Both data augmentation strategies are represented in Figure 5.7. The first technique, with the rotation, zoom, and flip, and the second technique, with the occlusion simulation. The third image contains a black rectangular shape which was generated by this technique.

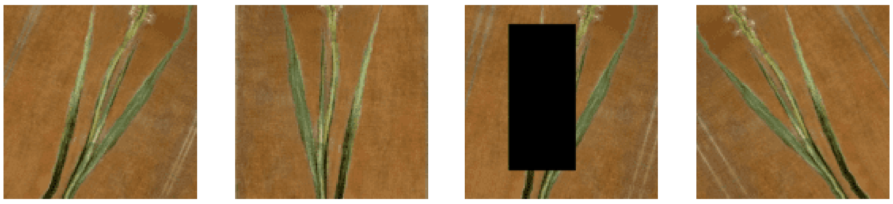


Figure 5.7: Image example transformed with all data augmentation techniques applied. "Gladiolus" by Claude Monet in 1881

Following the execution of tests for each of the aforementioned procedures, the obtained findings are shown in Figure 5.8, as was previously done in this document. The validation accuracy curves for each of the tested methods are shown in it.

Analyzing the diagram, the data demonstrates that what was previously expected did not occur. The graph represents the validation accuracies of four different models trained using four different data augmentation methodologies. The curve labeled "none" shows the previously computed model after any data augmentation procedures have been used. The remaining three curves reflect three models that used three different data augmentation approaches. The "augmented" labeled curve indicates the model that was trained using the set of simple transformations like rotation, flip, and zooming, as

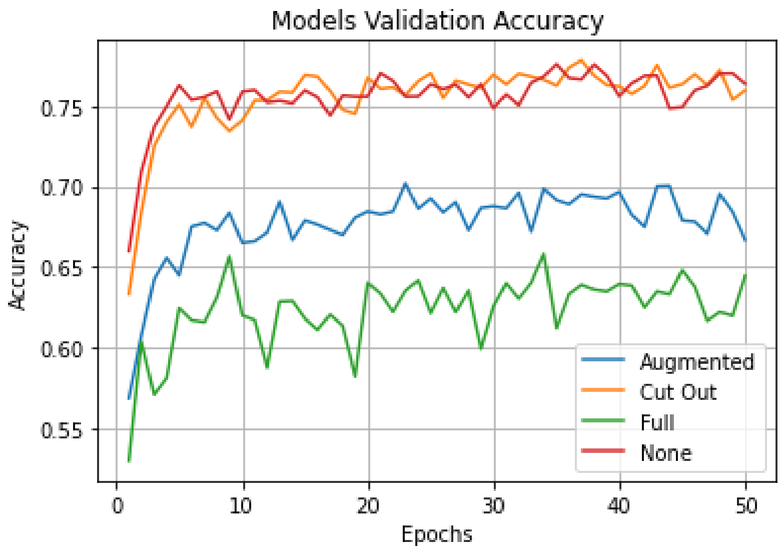


Figure 5.8: Comparison of validation accuracies according to each data augmentation technique

represented in Figure 5.6. The "cutout" labeled curve indicates the accuracy of a model trained solely on the simulated occlusion scenario. Finally, the third mentioned generated curve indicates the accuracy of the a model using all previously methodologies. Figure 5.7 shows an example of the data that was used to train this last model.

Contrary to expectations, the data augmentation technique using simple rotation, flip, and zooming transformations had no positive influence on the existing model. In fact, when these visual changes were applied, the model's accuracy scores decreased by almost ten percent. In contrast, the accuracy rates had no negative influence while training a model that can accurately predict images in occlusion scenarios, as indicated by the "cutout" labeled curve. Although there was no negative influence, the accuracy levels did not improve significantly, remaining quite similar, both reaching the same Top-1 accuracy values of almost 78 percent. Finally, after concluding that the tested model using visual transformations do not improve the overall model efficiency, it was expected that combining both visual transformations and the cut-out technique would result in even lower accuracy values, as there are more visual transformations when compared to the simple transformations previously tested. The curve with the label "full" in the diagram confirms this effect. It has the lowest validation accuracy of all tested models.

The observed decline in accuracy when applying transformations to images can be explained by the fact that these transformations affect the unique contents of the pieces of art, and, while the applied transformations are rather simple, they can obliterate the identity of the art pieces. The contents of the images, on the other hand, are not distorted when using the occlusion technique. As a result, it is apparent why the model's accuracies decrease when these transformations are used. It is reasonable that the contents of the image must be kept when training a model for this specific problem. On the other hand, while developing the model that predicts the style of the art works, these image alterations may be advantageous because the style may be characterized largely by colors or patterns and not so much by the contents of the images. Although this is expected behavior, it was thoroughly examined in this thesis in further sections to confirm it. As a result, the only data-related transformation that were used to construct this author recognition algorithm is the cut-out method, avoiding the need to apply any visual alterations to the existing data.

Along with these data-related optimizations, there are other architecture-related modifications that may be made to improve the model prediction capabilities. In the next section of this document, these optimizations are thoroughly explored and evaluated.

5.3 Hyper-parameters Optimization

Along with the creation of a robust CNN architecture, it is essential to establish some particular parameters, also known as hyper-parameters. Several parameters can be tweaked to increase the model's performance. Some of them are explored and manipulated in this phase of the thesis to determine whether it is possible to increase the model's accuracy and reduce the number of committed errors even further. Some of these hyper parameters have to be previously defined in order to execute the tests that have been done so far. The fact that these hyper-parameters have specific values does not imply that they were optimized. This is the work that was performed in this section. Understanding some of these parameters thoroughly before attempting to optimize the author recognition algorithm as much as possible.

5.3.1 Dropout

Dropout refers to the temporary removal of neurons from a NN, as well as all of their incoming and outgoing connections. The selection of which units to drop is random and follows a predefined probability value p. Dropping out is done separately for each unit and training epoch [59]. This technique is useful for minimizing overfitting and developing a more robust model architecture. As a result, in addition to the other hyper-parameters, this section includes an examination of the application of this approach.

5.3.2 Activation Function

Each neuron in the NN accepts the output value of the previous layer's neuron as its input value and sends it on to the next layer. There is a functional relationship between the output of the upper node and the input of the lower node in a multi layer NN. This is known as the activation function [22]. Three of the most frequent activation functions were tested for this author prediction algorithm. Tanh, Softmax, and ReLu [56]. These activation functions were used in this particular problem, and their influence was measured. Finally, the best result can be selected as the default value for the other problems.

5.3.3 Keras Tuner

As previously discussed in the subsection 5.1.5, when using the ResNet50 architecture, a simple NN had to be concatenated to the end of the existing network. This simple network is one of the parameters that was tweaked in this section, mainly the number of layers needed and the number of neurons in each of this layer. Alongside this network, it was also tested if using dropout has any positive impact on the performance of the algorithm. Finally, all of the previously described hyperparameters were explored in order to obtain the best possible performance model.

It would take a long time to test all of these proposed hyper-parameters, especially considering the context of this problem, where each training step is a very time-consuming operation. This is due to the fact that each training phase takes around eight hours with the used machine. As a result, testing all of the intended parameters would be a lengthy task, since it would require to test multiple combinations of it. As a result, an external tool, the Keras Tuner, was used [41]. Keras Tuner is a scalable, user-friendly hyperparameter optimization framework that tackles the problem of hyperparameter search. This tool allows to specify which hyperparameters must be explored and a range or group of values to be explored. The main goal is to test several combinations and keep track of the best reached performance set of values for these hyperparameters. Finally, the best architecture is kept along with these values, which may then be trained for this specific context. Although this procedure is fully automated and is supposed to provide a set of values corresponding to a greater performance model, it is not expected to return the optimal set of values. To achieve this, the execution of this tool would require to be repeated for a longer period of time in order to test more combinations. There are an endless amount of combinations for the proposed variables. Therefore, to retrieve an even better set of values, the Keras tuner execution must do a more in-depth and thus longer study. Although this is crucial in order to achieve an even higher performance model, the executed method is intended to identify a collection of variables that can build a model with very similar performance to the best one. As a result, the necessity to discover the ideal collection of values and hence spend additional time doing so is unjustifiable.

All of the previously specified parameters were evaluated when the Keras tool was employed. However, for each hyperparameter, the value range that was examined must be specified. The algorithm examines several possibilities to choose the best achieved architecture to then be implemented. It specifically tests a network with various number of layers, from a single layer to five layers, as well as the number of neurons in each layer. It tests multiples of 512 neurons to a maximum of 4096 neurons per layer. The dropout usage was also examined in this additional network. For this variable, models were tested both with and without the Dropout technique. Not only was this tested, but the probability value was also researched. When the dropout is active, this value was evaluated from a 10 percentage to 50 percentage value, with intervals of 10%. The Keras Tuner tool could also quantify the learning rate's influence on model performance, however this required a more in-depth investigation in further sections. For now, this value could be investigated from a range of 10^{-4} to 10^{-3}. Lastly, different options were used to examine the impact of the activation function on model performance. So far, the ReLu activation function has been adopted in all tests. Other functions, such as the Tanh and the Softmax were evaluated during the execution of this tool.

Upon running the Keras algorithm, the best set of values to be used to design the new architecture with the optimized values are generated. The comparison between the architecture used up to this point and the new architecture designed with the values generated by the previously executed tool is shown below in Table 5.3.

	Hyperparameters Values	
	Non Optimized	Optimized
Number of layers	2	1
Number of neurons in first layer	1024	2048
Number of neurons in second layer	1024	-
Dropout	True	True
Dropout probability	0.2	0.3
Activation Function	ReLu	ReLu
Learning Rate	0.001	0.0001

Table 5.3: Comparison between non optimized and fully optimized hyperparameters

This set of values corresponds to the final architecture as well as the set of hyperparameters utilized to solve the author recognition problem. Despite the fact that the Keras Tuner tool was able to find a better set of hyper parameters, two additional parameters must be manually tweaked to achieve even better performance of the model. The learning rate and batch size. These values cannot be fully optimized by the Keras tool.

5.3.4 Learning Rate

This parameter is directly related to how easily the model's weights are updated [32]. Choosing an adequate learning rate is a difficult challenge since a learning rate that is too low may result in a very slow and stalled training process, whereas a learning rate that is too high may result in diverging away from the optimal point rather than converging towards it [51]. This value can be tuned using a variety of techniques. As has been done previously, a simple technique is to select a constant value and use it during the entire training process. On the other hand there are various approaches in which this value is adjusted during the model's training phase. J. Konar et al. tested some of these techniques on a CNN [32]. They concluded that the approaches that produced the highest accuracies were the Cyclical and the Step Decay Learning Rates. For this reason, both these approaches were tested in the author recognition problem. Other technique that was tested in this article, the Step Decay, was also explored for this challenge. It is a variation of the previous mentioned Exponential decay, where the decrement value process is done in phases. Having explored these techniques, one other was also tested, the Linear Decay.

Even though the Keras tool produced an optimal value for the learning rate, this corresponds to an optimal value when using the constant value technique. This indicates that the learning rate value remains constant throughout the training period. As a result, from the previous studied possibilities [32], multiple learning rate approaches were explored in this phase while no other variables are changed. In contrast to the constant learning value, four different strategies were used, and their results could be analysed to see if they have any effect on the algorithm's performance. Figure 5.9 illustrates four distinct functions that were utilized to alter the learning rate value during the training phase. These are the four techniques that were evaluated in this section.

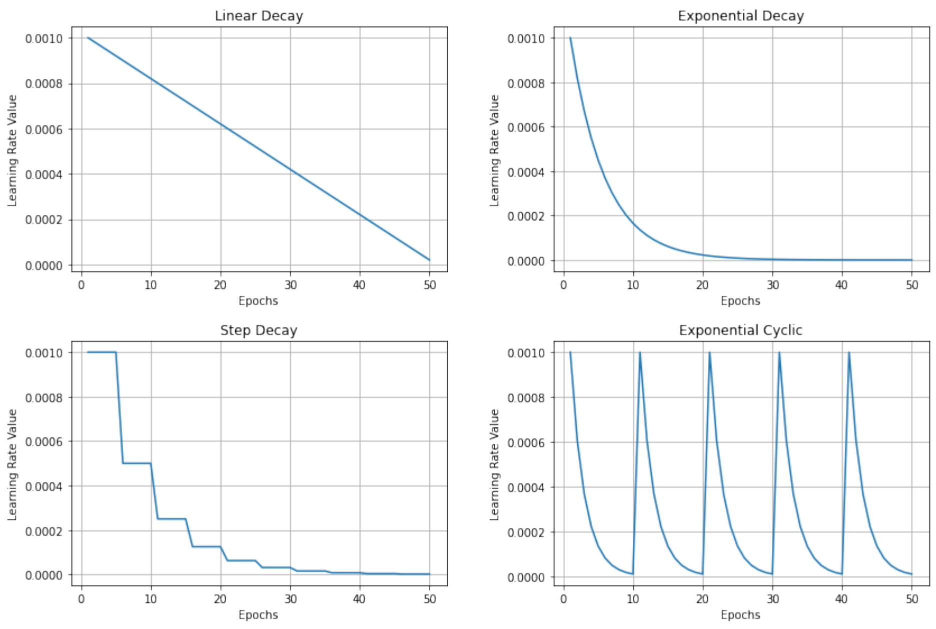


Figure 5.9: Four alternatives to the constant learning rate value approach. Linear, Exponential, Step decay and Cyclic learning rate mathematical representations.

- Constant Value - This is a simple solution to the problem. The learning rate value is one and does not change at any point during the training procedure. Since it was generated by the Keras tool, for the executed tests, this value is 0.0001.

- Linear Decay - The learning rate value begins at a high value and drops linearly over time when linear decay is used. This means that, at first, the algorithm explores many alternatives before settling on one.

- Exponential Decay - It is similar to the linear decay learning rate process. However, the value of the learning rate diminishes exponentially in this case. The following mathematical equation influences the learning rate:

$$0.001 * e^{-0.2*epoch}$$

- Step Decay - The value in this technique begins with a relatively high learning rate and subsequently declines during training execution. The main difference between this and the previous two approaches is that the reduction occurs every five epochs and stays constant while it does not change.

- Cyclic Learning Rate - This technique varies the learning rate value between two values repeatedly. The value starts at 0.001 and decreases exponentially, similar to what happens in the exponential decay approach. The main difference is that in every ten epochs, the value is updated to the initial value. The advantage of implementing cyclical learning rate over other learning rate strategies is that other processes do not guarantee that the model don't become stalled in a plateau region. In each cycle, the learning rate decays according to the following mathematical expression, where the percentage symbol represents the quotient remainder between the two values:

$$0.001 * e^{-0.4*(epoch\%20)}$$

Figure 5.10 illustrates the outcomes of evaluating various learning rate approaches. This diagram represents the evolution of validation accuracy during the training phase for each of the distinct techniques. The chart also displays the progression of the validation loss value for each tested methodology during the same training phase.

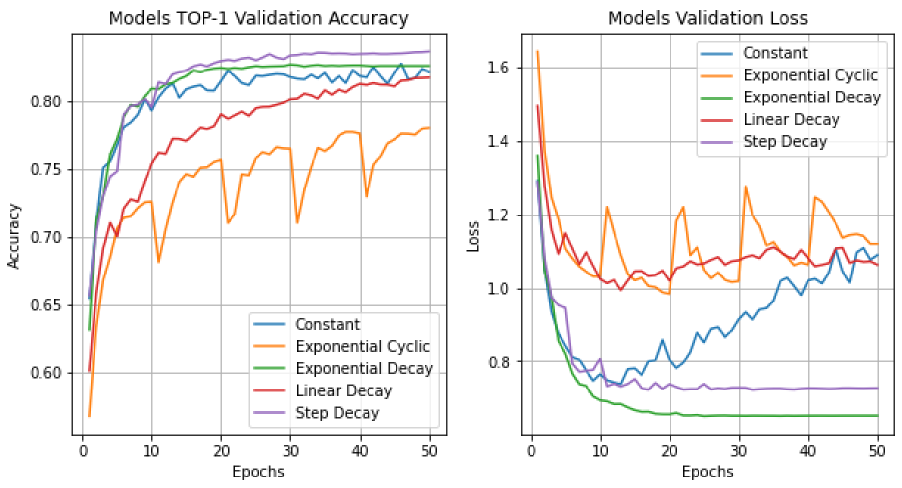


Figure 5.10: Validation accuracies and loss function when using different learning rate techniques applied to the author recognition algorithm.

Observing the generated results by performing the learning rate variation tests, it is shown that the best accuracy performance was reached when applied both the step decay learning rate variation, having reached 83.61%, and the exponential decay, reaching a maximum of 82.64%. When compared to the constant learning rate value, both of these techniques achieved greater Top-1 validation accuracy scores. Another consequence that is obvious when using both of these strategies is that the loss function decreases. This is a significant improvement, as the loss function increases when the naive technique is used. The loss function increases not just in the constant learning rate value, but also in the other two approaches. The linear decay and the exponential cyclic decay. This suggests that both the exponential and step decay techniques have shown significant improvement. Because the accuracy difference between these two approaches is minor, the appropriate learning rate value approach was determined by the validation loss value. Because the exponential decay provided the lowest results regarding the loss value, this was the strategy that was employed for this solution.

5.3.5 Batch Size

The batch size is one of the most commonly modified hyper-parameters in image classification problems. The batch size variable refers to the number of elements that are sent to the network at the same time in each epoch of the training phase. Its main purpose is to boost parallelism and reduce communication costs in the training phase of the model implementation. In general, the higher the batch size, the faster the model completes each epoch during training, and therefore the entire training process. This is explained because the network processes a larger number of images in each epoch. This computation process is directly related to the available computing resources. The consequence of adopting a large batch size is that, even if the computer can handle a large number of parallel picture processing, the model's performance may deteriorate. This occurs because it can cause the model to be unable to generalize properly on new data. Also, if the batch size is set to a large number, the computer may not have enough computational capability to process all of these images concurrently, slowing down the training process [23]. This would imply that the batch size should be reduced. This variable, in general, is one of the hyper-parameters that must be tested and tuned based on each model context. According to the publication [48], the batch size must be a power of two or multiples of ten. However, as previously stated, neither numbers to the power of two nor numerical multiples of ten cause a significant difference in recognition accuracy. Different batch size values were tested for this author recognition problem. The training procedure was performed with the following group of values:

$$B \in \{16, 32, 48, 64, 80\}$$

Although one of the same paper's conclusions was that batch sizes of more than 200 produced the greatest outcomes, the maximum tested value for this problem is 80 because it is dependant on computer resources.

Observing the Figure 5.11, where the validation accuracies and loss results are displayed, it is conceivable to draw certain conclusions. To begin, the validation loss function value decreases as the batch size increases. This indicates that the more concurrent image processing there is, the fewer errors the system makes when predicting new images, which is the expected outcome. According to the returned results, the loss function decrease is proportional to the batch size. This suggests that the loss function is smaller

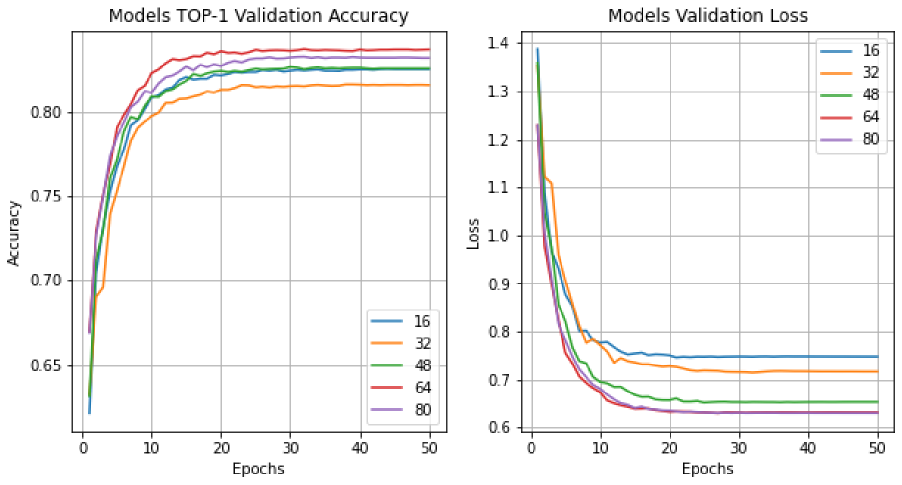


Figure 5.11: Validation accuracy and loss function of multiple batch sizes used to train the author recognition model

when using larger batch sizes. Furthermore, when testing the algorithm with different batch sizes, it is possible to detect certain differences in validation accuracy. In most circumstances, the highest achieved value for validation accuracy is related to the batch size value. That is, the larger the batch size, the greater the maximum Top-1 validation accuracy. Although this effect may be seen across the majority of the tested values, there is one notable exception. This pattern was broken during the training phase when batch size was set to 16. The maximum validation accuracy for this batch size is nearly as high as the identical procedure when conducted with a batch size three times larger, hitting roughly 82.5 percent. This can be explained by the fact that having different batch size values does not restrict the algorithm from achieving better accuracy levels than anticipated. This effect can also occur in the opposite direction, with a training procedure producing results that are lower than expected. Although it is unlikely, it is possible since the training execution becomes unable to identify the most appropriate weights. Alternatively, the training execution may identify a better set of weights than expected for that batch size value. As a result, the best conclusion that can be drawn is that with larger batch size values, the method is more likely to achieve better validation accuracy values. However, this phenomenon does not always occur. Based on the results, it is plausible to conclude that the batch size of 80 is the ideal value to adopt in this author recognition situation. If there are additional computational resources, a greater value is likely to produce even better outcomes. Therefore the batch size is set to the largest available value.

This section of the document comes to an end with these conclusions. With the help of this study, it was possible to gain the necessary knowledge about the optimum set of techniques for improving the prediction accuracy of an IC model. Figure 5.12 represents the results of all of the work done in this section. It shows a visual comparison of the validation accuracies and loss function of a non-optimized model and the final highly

	Training Accuracy			Validation Accuracy		
Model	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
Not tunned	0.9780	**0.9987**	**0.9998**	0.7016	0.8422	0.8947
Tunned	**0.9824**	**0.9987**	0.9997	**0.8371**	**0.9332**	**0.9610**

Table 5.4: Comparison between the maximum training and validation accuracies of a non-tuned model and a tuned model

optimized model. The non-optimized model is a simple ResNet50 architecture with a basic NN and a non optimized set of hyperparameters. On the other hand, there is a fully optimized model that has been submitted to data augmentation techniques, specific image resolutions, resizing method, and hyperparameter optimization using the Keras Tuner. In addition, based on the available machine resources, the learning rate and batch size were manually optimized to this context. As can be seen, the combination of the various optimizations used to solve this specific author recognition problem has a significant impact on the validation accuracy as well as the validation loss function, achieving an improvement of over 13% in the Top-1 validation accuracy and significantly improving the validation loss. When the loss function in the first model is evaluated, it worsens linearly with the number of epochs. In opposition, with the optimized model, the loss function demonstrates that the correctness of the model's predictions improves with each epoch, becoming lower over time. These results indicate that, with the optimization of the model's architecture, it can not only accurately predict in more circumstances, but also, when it fails to do so, the committed mistakes are reduced. The other significant difference between these results, which is visible in the diagram, is the smoothness of the curves. The curves of the highly optimized model have minimal variations, and their smoothness increases linearly with the number of epochs. The basic model's curves, on the other hand, are quite irregular and exhibit substantial variances throughout the training process.

Table 5.4 displays the highest achieved values of the Top-1, Top-3, and Top-5 accuracies of both the basic ResNet50 with transfer learning model, and the fully optimized model produced by the Keras tool execution and the manually tweaked learning rate and batch size variables. Not only was the validation accuracy substantially improved overall, but the training accuracy of the non-tuned model was slightly outperformed by the optimized model in some of the maximum Top accuracies. The fact that it was unable to improve on a greater scale in training accuracy values indicates that the non-tuned model is overfitting. As a result, it is quite good at predicting previously seen images but not so good at predicting new ones. This is something that the optimized model also helps with.

Despite the fact that this research was directly applied to the development of the author recognition model, similar techniques were applied to all of the other problems mentioned for this thesis. Because all of the other proposed algorithms to be developed are relatively similar to the one developed in this section, several optimizations, such as

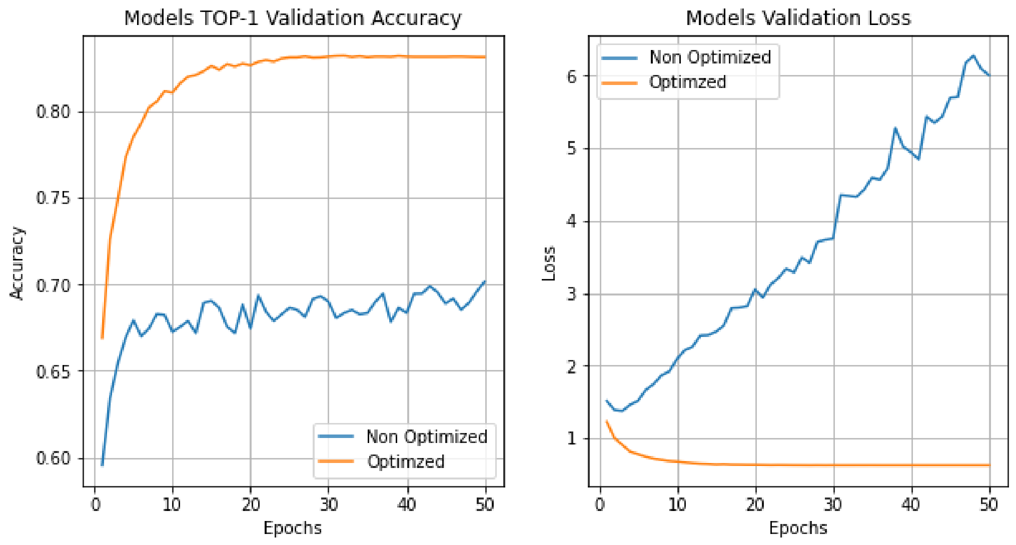


Figure 5.12: A comparison between the validation accuracies and loss function of a non optimized model and a fully optimized model applied to the author recognition problem.

image resolution, did not require a thorough investigation. The author prediction algorithm's results were expected to be identical to the other obtained outcomes in this matter. However, other parameters, such as data augmentation strategies, must be explored, because the context of the other problems may affect the usage of different values.

5.4 Context Related Architectures Implementation

As previously mentioned, because the four algorithms proposed for his thesis are somehow context related, and because they are directly related to the extraction of features from the same type of images, in particular, art pieces, it is reasonable to bypass the study of the same parameters. The network configuration, a ResNet50 with the transfer learning technique, is one of the parameters that were maintained. In addition to this parameter, the resizing method and image resolution settings are the same as in the previous produced model, using the crop technique and a resolution of 448 pixels by 448 pixels. Also, both the learning rate approach and the batch size value were maintained in all algorithms. In particular, the exponential decay learning rate was used with the maximum batch size value of 80. As a result, while building the models for the other three proposed art classifiers, a brief evaluation of the data augmentation approaches and the optimization of the additional NN architecture was required. Despite the fact that this extra NN structure could be optimized, some of its parameters remains unchanged. Both the dropout technique, with a value to be determined, and the activation function, which was the ReLu function, were retained.

These parameters for each of the following algorithms are thoroughly examined in the next section. This study, identical to that done in the previous section, required to create the final architecture for each proposed models. Because distinct challenges are expected to be solved, each algorithm requires a different approach. These disparities were also investigated and taken into account in this section to design each model based on its context.

5.4.1 Art Style Prediction

This algorithm is directly related to the art piece's art style. This art style can be divided into 27 distinct styles, as it was concluded in section 4.1. The main distinction between this challenge and the other three is that an artwork can be categorised into multiple styles. As a result, when analyzing the results of the training process, it is critical to consider not only the Top-1 accuracy scores, but also the Top-3 accuracy values. As a result, it is not expected to have high Top-1 validation accuracy scores. This occurs because the algorithm can classify an image as a different value than the expected during the model's training phase. This does not imply that the expected value is incorrect. As a result, it is critical to seek for a collection of expected values rather than just the initial predicted value. To evaluate which is the best data augmentation technique to be applied to the art style recognition challenge, different tests were executed applying these variations. The training phase was carried out in three conditions. The first by using no data augmentation approaches, one other with the data with a fully visual set of transformations, and lastly the use of the cut out methodology. This is the same procedure that was done to examine the influence of data augmentation in the author prediction problem in subsection 5.2.3.

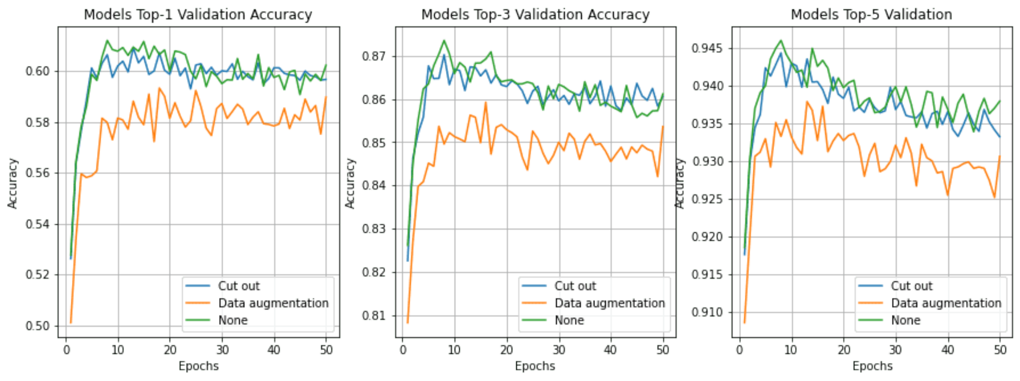


Figure 5.13: Top-1, Top-3 and Top-5 validation accuracy graphs using different data augmentation techniques

The evolution of the Top-1, Top-3, and Top-5 validation accuracy values over the course of the various training cycles is represented in Figure 5.13. It is evident from these curves that the results are cohesive across the three different measurements. The validation accuracy scores are significantly worse when using visual transformations on the data used in this context than they are when using the other two procedures. On the other hand, the training procedures that do not used any technique or the one with the cutout approach achieved the highest accuracy levels across all three metrics. Despite the fact that both accuracy curves produced are fairly similar, the model that produced the highest accuracy values among the three metrics was the one that did not visually modify the data. The model that used the data without modification reached 61.20 percent, 87.36 percent, and 94.60 percent in the Top-1, Top-3, and Top-5 metrics, respectively, whereas the model that used the cut out technique only reached a maximum of 59.33 percent, 85.93 percent, and 93.79 percent in the same metrics. It is possible to conclude that the model provides higher prediction capabilities without the usage of data augmentation strategies than those that the model can achieve while employing the other two ways. For this reason, there were not employed any modification to the data that is used in the style recognition problem. After determining the best strategy for the data transformations that should be used to increase accuracy, it is time to research the neural network architecture that must be created and added to the end of the ResNet50 network. Similar to how it was done in the author's problem, the Keras Tuner tool was used for this assignment. Using this tool, an optimal layer number, matching neuron count, and dropout probability value for this model was determined. For the style classification challenge, this collection of optimized settings define a more effective network architecture.

Training Accuracy			Validation Accuracy		
Top-1	Top-3	Top-5	Top-1	Top-3	Top5
0.8629	0.9845	0.9971	0.6394	0.8889	0.9539

Table 5.5: Maximum Top-1, Top-3 , and Top-5 training and validation accuracy values reached when training the style recognition model with the final architecture design and set of parameters

After applying this tool, the final neural network's ideal values were obtained by using two layers, each having 1536 and 2048 neurons. In each of these layers, dropout with a probability value of 0.2 was employed. The activation function is identical to the one that was applied to the author problem. Having found these values, this was the architecture that was used to the final style recognition problem. The generated model could achieve a greater validation accuracy with these enhancements. Prior to the use of the Keras Tuner, the Top-1 validation accuracy score was 61.20%. Now it is 63.94%. Additionally, the Top-3 and Top-5 accuracy both increased. These values may be shown in Table 5.5 along with the training accuracy values.

5.4.2 Art Decade Prediction

Two methods were investigated to extract time-related information from an artwork, as mentioned in the data description section of this document (4.2.2). Creating an algorithm that is able to predict the year of an artwork was the first strategy. On the other way, it was evaluated the prediction of the artwork's decade to understand if it had any positive impact on the prediction accuracy of this matter. The amount of classes that are provided for each problem in these two approaches is the main distinction between them. When compared to the decade technique, the year prediction approach has a significantly higher number of classes. It was estimated that this would result in reduced precision prediction capabilities. However, this was deeply investigated to determine the true impact of each scenario.

The results of the two discussed procedures are shown in Figure 5.14. It displays two curves that represent the evolution of the validation accuracy over the 50 training epochs of these two explored methods. As anticipated, the decade strategy yielded the higher accurate validation results. Therefore, an algorithm that can identify the decade of a piece of artwork was developed in order to solve this time-related classification model.

Once the method for incorporating the time-related information has been chosen, in this case, the decade prediction approach, the performance of this model must be improved before it can be implemented. Although the decade prediction strategy has demonstrated the highest accuracy results among the two tested approaches, as can be seen in the accuracy results presented in Figure 5.14, the best reached accuracy is the lowest when compared to the other algorithms created so far. This can be explained by the difficulty of the task. Even for an algorithm, predicting the decade or the year of a work of art can be challenging. The fact that various artistic genres, artists and other variables might share same date values is the most reasonable explanation for this. Because of this, it was anticipated that this decade prediction challenge would have inferior accuracy capabilities than the other prediction algorithms, even after the implementation of multiple optimizations.

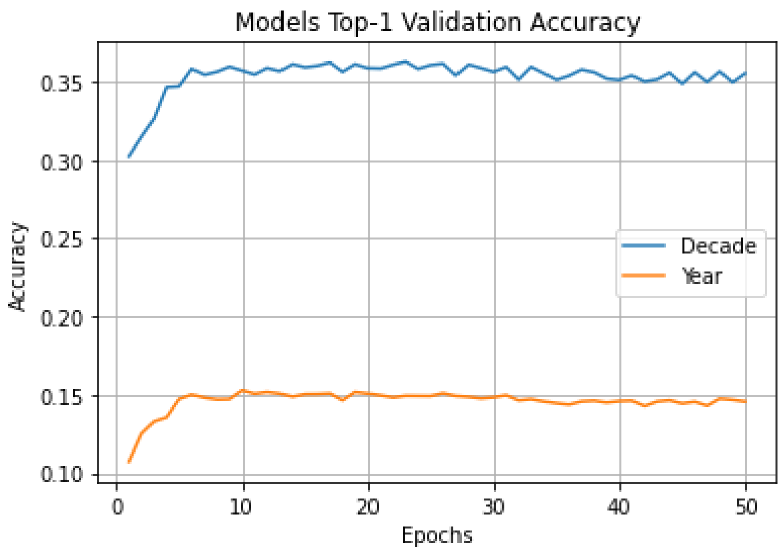


Figure 5.14: Comparison between the Top-1 validation accuracy during the training process using data divided into the art piece year and the art piece decade

Visual modifications were applied to the model's training data, just like they were for the prior models. The outcomes of applying these changes were then analyzed in order to determine how they affect the model's prediction performance. Following the same procedure as in other cases, the image first suffer a number of transformations, including rotation, flipping, and zooming. Using a different technique, separately, dark rectangles were added to the images. To determine which strategy should be further examined, these two are initially evaluated in different scenarios. After the images have been modified, the model was provided with these in order to carry out the training phase. The output from this phase shows how well each model can predict. Figure 5.15 shows how each strategy affects the model's capacity of prediction. As can be observed, the validation accuracy and loss function under the cut out procedure are significantly poorer than under the other two scenarios. On the other hand, comparing the model that was trained on raw data with the one that uses a set of three changes labeled as "Data Augmentation", the validation accuracy is not very different. However, analyzing the loss function of these two scenarios reveals the only distinction between them. As can be seen, the "Data Augmentation" strategy's validation loss function significantly outperforms the default approach's validation loss function. This indicates that the capabilities of the overall model can be improved when a set of visual transformations, such as rotation, flipping, and zooming, are applied to the data.

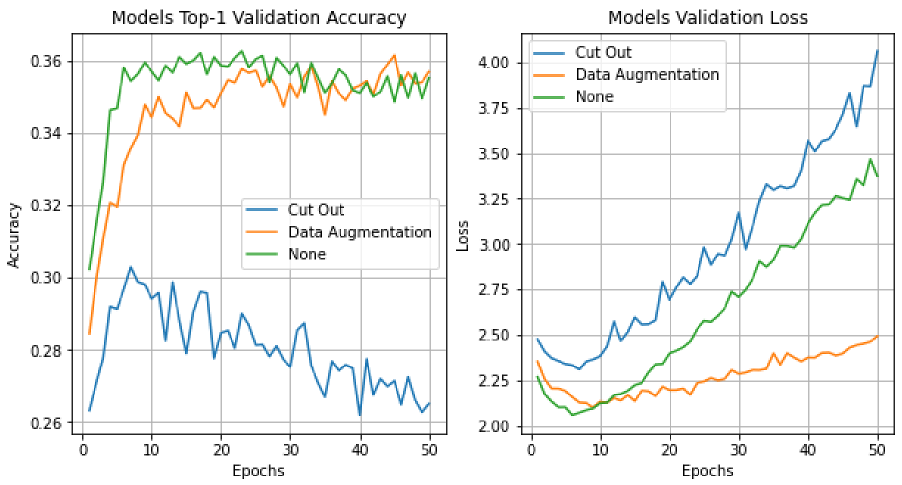


Figure 5.15: Validation accuracy and loss function from different data transformation techniques applied to the decade model training phase

It is possible to do an additional investigation on the chosen data transformation approach. Since the tested method employs three distinct visual modifications, each one may be examined independently to determine whether there is any that has a negative effect on the model. If so, the ones that negatively impact the model's performance, must be discarded. To comprehend this, the model must be trained on images that have undergone each visual transformation. The results of this test are shown in Figure 5.16. When the tests' findings are analyzed, it becomes clear that the four various methodologies' validation accuracy is relatively similar. As was previously observed, the validation loss function is the only difference. All three approaches demonstrate an improvement on this aspect when compared to the baseline approach using data without any modification.

All of the implemented approaches have validation loss functions that are lower than the base model. As a result, each one lowers the model prediction error rate. This makes it easy to come to the conclusion that the decade prediction model is positively affected by all three of the adjustments that were previously used. As a result, the set of these transformations, must be used for the algorithm's development.

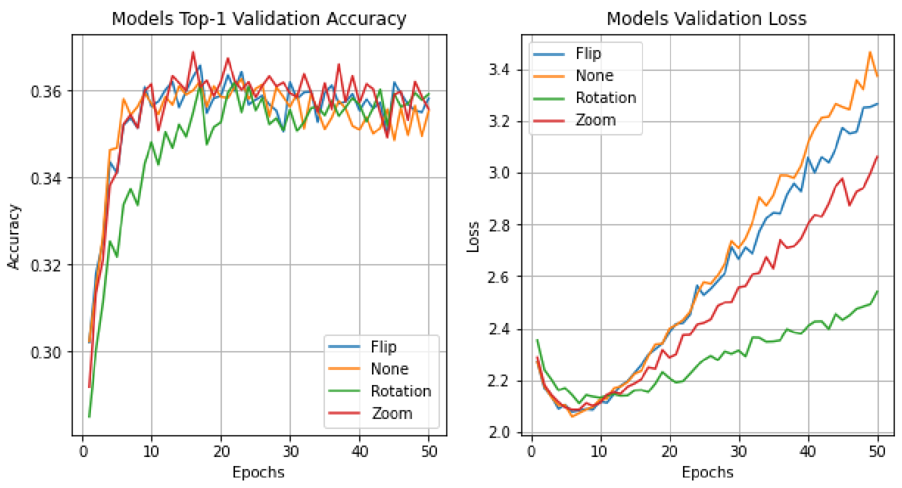


Figure 5.16: Validation accuracy and loss function from models trained with data that suffered rotation, flipping and zooming transformations.

Keras Tuner was used to determine whether there is a better architecture for the network that can be used to train the model. With the help of this tool, a set of values were determined. The usage of these values is expected to improve the model's prediction power. Namely, the number of layers, the amount of neurons in each layer, and even the probability of dropout. After finishing the tuning procedure, the model's final architecture for this problem is established. Table 5.6 provides a representation of this network. With this new architecture, the Top-1 validation accuracy of the decade categorization improved by over 2%, achieving its maximum value of 38.75%.

5.4.3 Art Type Prediction

To implement the last algorithm of the proposed ones, the recognition of the type of art, the same optimization process was done. The type of art recognition problem consists of the classification of an art piece among 9 different classes (4.2.3). When compared to the other three art feature extraction challenges, the number of predictable classes is much lower (4.2.4). This means that the algorithm is expected to have the highest training and validation accuracy values of the four algorithms. Having trained a model with the gathered data, it was able to make predictions with about 92% of validation accuracy, proving the previous observation. This training stage was repeated using two additional data transformation strategies, similarly to what was done with the prior challenges. This is done to determine how well various approaches affect the models' accuracy.

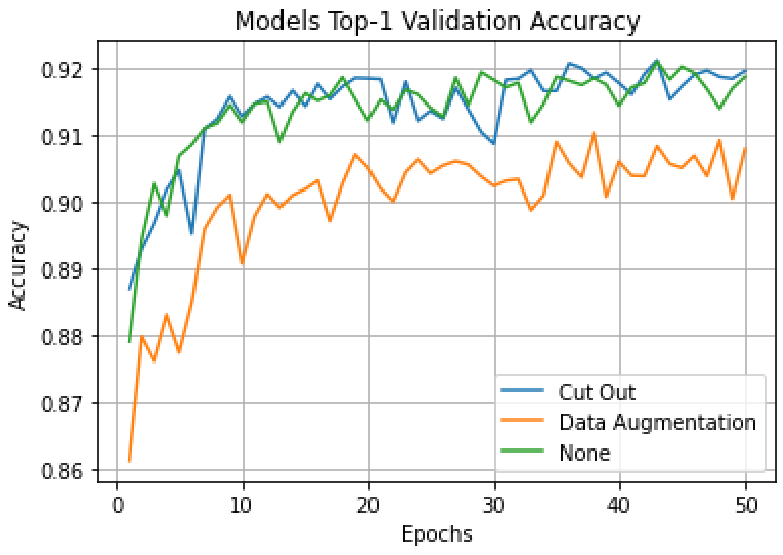


Figure 5.17: Validation accuracies of the art type's recognition problem when training with different data augmentation techniques

Figure 5.17 compares the accuracy of the same model using the various tested methodologies. The model's poor prediction abilities when trained on data that suffered a series of visual modifications are the first obvious observation that can be made. Comparing to the validation accuracy scores from the other two procedures, it is substantially lower. On the other hand, over the training process, the validation accuracy curves produced by these two approaches are quite similar. The method that uses the cut out technique resulted in an algorithm that achieved a 92.11 percent validation accuracy. On the other hand, the model with no modifications yielded a maximum of validation accuracy of 92.09 percent. Despite the slight variation between these two values, only one strategy must be used to develop the prediction algorithm. Due to the cut out approach's ability to improve the model's behavior in real-world scenarios, and the fact that the original model is not negatively impacted by its use, for the model's development, the data is submitted to a set of changes related to visual occlusion circumstances.

Lastly, the Keras Tuner must be used to optimize the final neural network's architecture. This tool was used to define the final architecture and the set of optimal parameters. The maximum Top-1 validation accuracy value was improved by almost 1% with this last strategy, achieving the 92.85 percent accuracy value. Although this 1% gain might appear insignificant, considering the current validation prediction is set to about 90 percent, it is unrealistic to anticipate significant improvements.

For each one of the developed models, Table 5.6 displays the best set of values that could be optimized. The number of layers and neurons alongside with the best dropout probability value. All the other hyperparameters are also displayed in this comparison in addition to this data.

	Author	Style	Decade	Type
Number of layers	1	2	1	1
Neurons in first layer	2048	1536	2048	1536
Neurons in second layer	-	2048	-	-
Dropout	True	True	True	True
Dropout probability	0.3	0.2	0.2	0.3
Activation Function	ReLu	ReLu	ReLu	ReLu
Learning Rate	Exp. Decay	Exp. Decay	Exp. Decay	Exp. Decay

Table 5.6: Final neural network architecture and set of hyperparameters values of each one of the implemented algorithms.

5.5 Additional Data Acquisition Method

The extraction of four features from art pieces is one of the thesis' main propositions. The image's author, art movement, art type, and decade. Not only are these features expected to be extracted, but a number of other characteristics were originally proposed to be detected as well. As previously stated, there are multiple tools that can be used to complement the information that can be extracted from images. Google's Cloud Vision API, MobileNetV2 and Tensorflow Hub were three of these techniques that were studied in section 3.2. More information may be extracted from images using these tools. A larger variety of characteristics that were previously very hard to obtain because of the data complexity issue, may now be extracted. This section undertakes a research and analysis to see whether these tools have a positive effect on the proposed platform. Having made that decision, one of them was picked, and an analysis of the data that may be acquired from this instrument was done. Finally, it was tested, and the results were examined. In order to properly understand how this tool may be used in this thesis, a conclusion was composed.

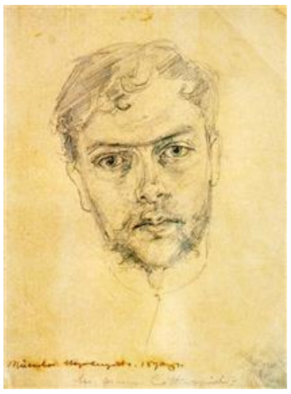


Figure 5.18: "Autoportret" by Stanisław Wyspiański in 1890

The three tools stated above were put to the test with various inputted images, and the results were compared in order to determine which one should be chosen. In general, Google Cloud Vision produces a wider range of recognized characteristics than MobileNetV2 and TensorFlow Hub, including facial expressions, objects, labels, the most

Google Cloud Vision API		**MobileNetV2**		**TensorFlow Hub**	
Label	Confidence	Label	Confidence	Label	Confidence
Forehead	98%	Spotlight	25%	Book Jacket	79%
Nose	98%	Shower Curtain	9%	Pickelhaube	1%
Head	97%	Window Screen	4%	Envelope	0.8%
Eye	94%	Lampshade	4%	Comic Book	0.7%
Painting	85%	Fire Screen	4%	Bearskin	0.5%
Self-portrait	75%				
Sketch	58%				

Table 5.7: Comparison of results acquired by applying to the same image different label detection tools

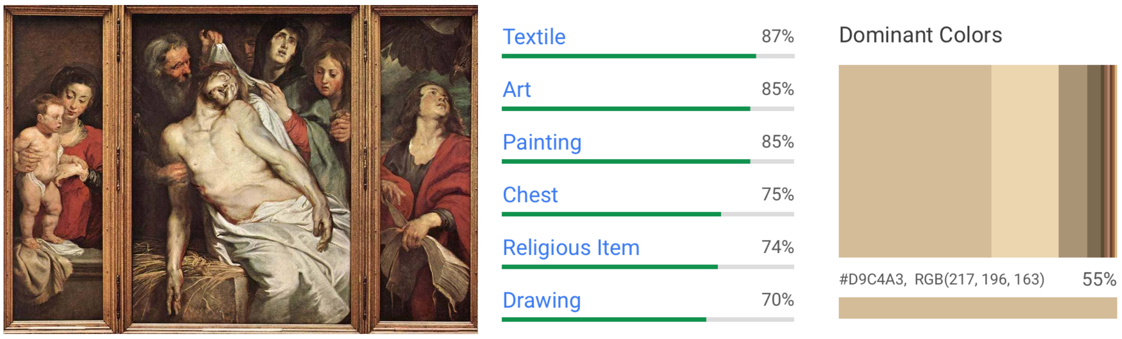
prevalent colors, and safe search. The Google's tool not only provided results with a greater number of predictable characteristics than those produced by the other tools, but it also produced results with higher accuracy in image classification. Table 5.7 helps to visualize this.

Analyzing a concrete example from the testing phase, when comparing the results giving the Figure 5.18 as input for these three ML tools, MobileNetV2 and TensorFlow Hub were not able to identify any correct label of the inputted image, while the Cloud Vision tool was able to not only recognize some facial features, such as "Forehead" and "Nose" but also it was able to identify that the image was a "Painting" and a "Sketch". One of the returned labels was also the "Self-Portrait" label, which indicates that this tool is able to correctly identify the type of art. It was observed that the algorithms had similar behaviors to the previously described ones throughout several testing images, having the Google Cloud Vision producing more accurate results in all of them. Concluding, it can be said that the Cloud Vision is the best tool to use for this art problem when trying to identify a non-concrete number of labels, as said previously, without the need to train a complex model that can identify those labels.

Face, text, logo, and other feature detection are just a few of the features that may be extracted from images using the Google Cloud Vision API. There is a wide range of data that may be collected from images using the Google API. Only a few of these elements, however, are appropriate for application in this thesis platform given the context of the art problem. Only two of the features that can be extracted using this tool's collection of features were used. They are the label detection, and the image properties, which gives the colors that were detected from the images.

A tool that may extract several labels that are somehow related to the uploaded image is provided by the label detection feature. Generalized labels for objects, locations, activities, animal species, and more are available. Since it may extract more information than the developed algorithms, this is a crucial tool to be used in the art characteristics recognition problem. While the established algorithms are only capable of identifying the author, style, type, and decade of an artwork, with label detection, additional information, such as objects, can be collected. The capability of this tool to complement the current algorithms is another significant application. It's possible that this label detection yields information on the type of artwork. Consequently, be utilized to supplement the currently existing and previously established art type algorithm. The materials used to create the artwork is another piece of information that could be acquired from this label recognition tool. Even though it is uncertain whether this data could be presented, when it does, it can be used to give the artwork's used materials.

In addition to the label detection, the image properties extraction feature was also used in this art feature extraction. This tool detects general attributes of an image, such as the dominant colors. With knowledge of the most prevalent colors, it is feasible to filter the data based on this characteristic and perform operations on the data as a result. Some of these include retrieving data depending on a selection of colors.



Textile	87%
Art	85%
Painting	85%
Chest	75%
Religious Item	74%
Drawing	70%

Dominant Colors

#D9C4A3, RGB(217, 196, 163) 55%

Figure 5.19: Labels and dominant colors detection applied to the paining "The lamentation over the dead Christ" by Peter Paul Rubens in 1618

After deciding which data retrieval methods could be used to the images of the pieces of art, a quick test was conducted to determine how well the method performs in real life scenarios. Figure 5.19 displays the labels and primary colors that Google's identification technology retrieved after receiving the image of the painting. Despite the incompleteness of the presented list of detected labels, some inferences can be drawn from the supplied set of labels. The algorithm's confidence level is used to order this list in decreasing order. Therefore, presumably, the first entries are the ones that are most truthful. After doing a quick examination of these facts, it is possible to understand the connection between every label provided and the submitted painting. The "Textile", "Chest", and "Religious Item" tags, out of the ones that were detected, are the ones that offer more details to the algorithms that are already developed. In addition to these labels, some of them may be used to complement the data that is collected from the developed algorithms, as was previously described. The terms "Painting" and "Drawing" in this instance are both directly related to the art type classification. This is crucial to further strengthening the type of art's recognition IC problem. The system correctly detected the set of most dominating colors in the given painting in addition to the labels, delivering additional information based on a single image file.

It is possible to draw the conclusion that from an image file, it is possible to retrieve information regarding the art author, style, type, decade, materials, a set of relatable labels, and even the set of most dominant colors after testing all the tools that were used to extract the highest number of features from images. All of this data is valuable for accurately complement the cataloging process of any artwork submitted.

6

Infrastructure Design and Platform Implementation

The topic that has gotten the most attention so far in this thesis is the machine learning algorithms. This is due to the fact that it has proven to be the most challenging aspect of it. However, to achieve the goal of developing a full platform, several components are required. The infrastructure that was designed and created to ensure the successful operation of the platform that is proposed is fully described in this section of the document.

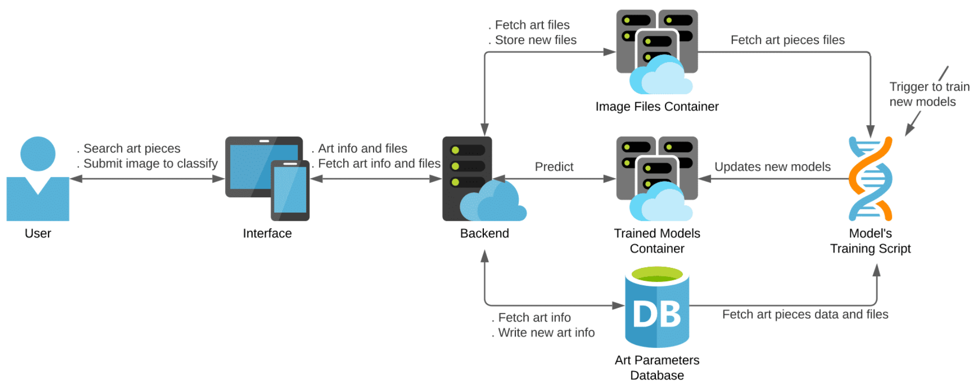


Figure 6.1: Diagram of the final infrastructure and all its components and main operations between them.

The infrastructure is composed of various independent components. This division and the basic interactions between each component of this infrastructure are represented in Figure 6.1. The user is one of the most important aspect of this system, and their interactions with the interface are the primary concern. This interface is dependent on the back-end system, which communicates directly with three elements. The components that hold the files and the database that has the information of each individual work of art available in the system. The machine learning algorithms, which are kept in a separate component, are also directly communicated with by this back-end component. Finally, there is a component whose primary purpose is to be used to retrain the models as necessary.

6.1 Interface

Having determined the algorithms and tools required for the efficient acquisition of information from works of art, the interface that serves as the connection between this information and the user must be established. Since the interface serves as the primary point of contact between the platform and the end user its main objective is to present information to the user in a clear and straightforward way. Every user must be able to use the interface easily and intuitively. For this reason, it must only include the components that are necessary for its use. The interface must simultaneously contain all the data that needs to be given to the user. As a result, the user must be presented with information that is clearly organized and presented.

Only a few fundamental functionalities are supported in this phase of the platform's development. One of this functionality is the capability for a user to find works of art using known parameters, like the name or the author of the art piece. The capability for users to submit new works of art is another significant functionality. The interface can be divided into two different sections as a result. Users can search for images and information of art pieces in the first one. The functional component for adding new works of art is in other section.

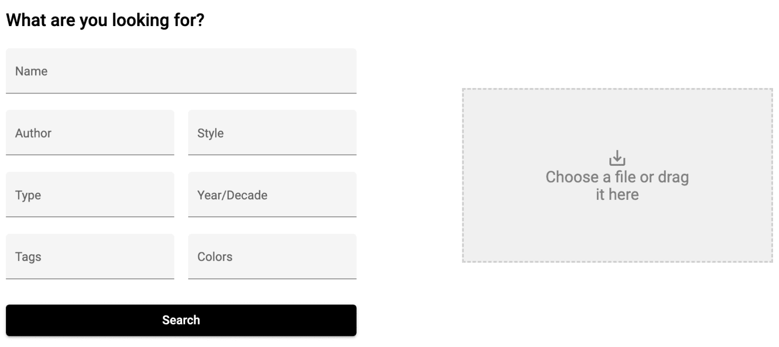


Figure 6.2: First page of the interface and all the components of it.

The platform's home page is the first page that is presented to the users. It must plainly display all the functionalities that are offered by the platform. Figure 6.2 is a representation of this first page. Here it is possible to visualize the two main functionalities of this platform. As was already mentioned, users interact with this page to search for artworks and the information related to them. For this, the user need to enter known parameters whenever seeking for any information. The name, author, style, type of art, year or decade, tags and materials of the art work are all potential search criteria that the user may utilize. This search algorithm does not have a single parameter restriction. As a result, a user can input a set of multiple parameters to further filter the data. It is a requirement for this platform to explore for works of art. Users are able to find previously known works of art or learn about new ones thanks to this. Additionally, with this feature, it is possible to lessen the number of duplicate pieces of art. Users are less likely to submit the same artwork if they are aware that it is already in the system. The main goal of this capability is to provide consumers with a quick and easy approach to find works of art.

The platform displays two sets of information whenever a user enters information to conduct a search. The first one consists of a collection of objects that meet all the conditions that were provided by the user. However, the user's combination of values may result in this first set of data having poor information, thus the system returns a second set of data. The works of art that are displayed in this second collection have at least one parameter value that was supplied by the user. Both of these sets have a 50-artwork restriction to avoid having excessively data transfers between the platforms components. Additionally, each returned piece of art displays more details when the user moves the cursor over them. This is essential to give the user key details about each work of art in a simple way. The name, authors, styles, type, and year of the artwork are all included in this. Figure 6.3 shows this page and all the information that has been described. A search of works of art by Vincent Van Gogh and in the Realism style was conducted to get this information. Data that satisfies both requirements is shown in the first row. The second row, however, offers works of art that are either by Vincent Van Gogh or part of the Realism art movement. The information from the fourth piece of art in the first row of results is also visible in the figure.

Users are also able to locate an indication for the other mentioned functionality on the first page. However, only the second screen has the functional portion of it. The user can interact with newly uploaded works of art on the second page. Still in this first page there is a section where the user can select an image file to then be processed. All of the previously used algorithms operate in this environment to recommend as many parameters as possible. The three main elements of this page are a collection of labels that hold the data collected from the algorithms and methods previously discussed, a button that displays comparable images to the user, and finally a submission button. The platform processes the image file that is received from the user and extract as many features as it can from it with the help of the developed algorithms. The acquired labels are subsequently displayed in the appropriate sections of the interface for each label. The labels provide details on the materials, type, decade, potential authors, and styles. Additionally, the interface describes the detected colors as well as the detected tags from the artwork. Finally, the user are also able to enter the name and a brief description of the artwork in addition to these. Figure 6.4 illustrates this second page of the interface.

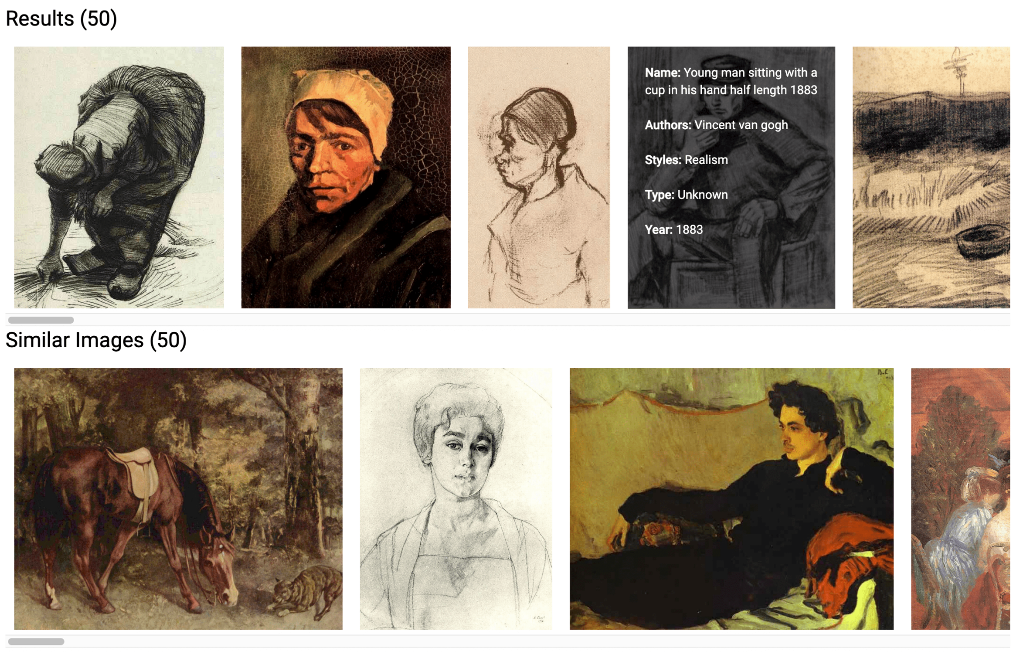
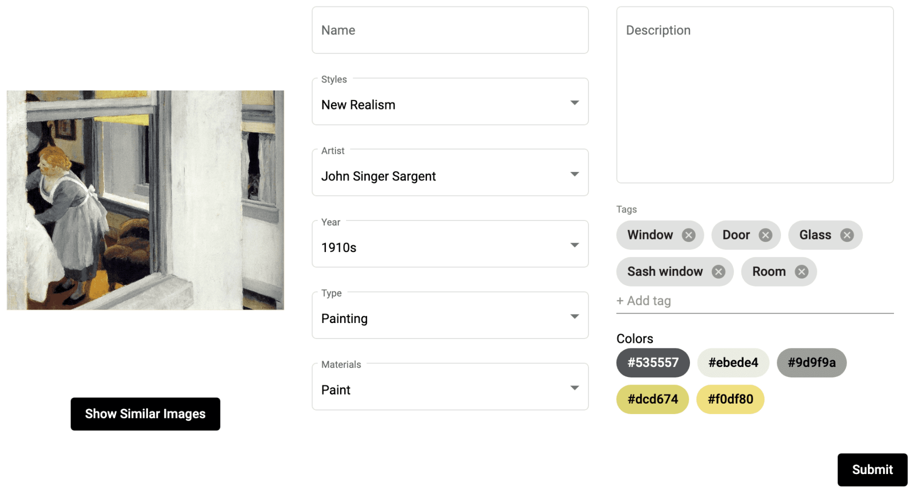

Results (50)
Name: Young man sitting with a cup in his hand half length 1883
Authors: Vincent van gogh
Styles: Realism
Type: Unknown
Year: 1883
Similar Images (50)

Figure 6.3: Data representation of a search of art pieces from Vincent Van Gogh and from the Realism art style.

Following a submission of a new piece of art to the system from a user, it is possible to view all of the recommendations made by the machine learning algorithms according to the submitted image. These values are available in each field that is displayed in the Figure. With the exception of the detected colors element, all of these parameters are editable by the user. The user can add new information or remove data that has been provided to it. This gives the user the ability to not only correct any information, but also to provide information that the algorithms were unable to acquire. The only piece of information that does not require editing is the information about the detected colors because it is always given and is not depending on ML algorithms. For this reason, the provided colors are expected to always be correct.

When submitting artworks, the user gets the opportunity to interact with pieces of art that are in some way connected to the submitted one. With this, users can view further works of art by the suggested criteria. The author, art movement, type of art, materials, and decade are the attributes which can be used to filter the data in order to see similar works of art. The existing data is filtered and presented to the user as any of these parameters has suggested values by the algorithms. Only a maximum of five works of art per value are shown to the user in order to reduce the amount of data transmitted. Figure 6.5 is a representation of this functionality applied to the previous submitted image in Figure 6.4. In this case, the algorithm was successful in identifying five photos that were associated with some of the recommended values for each of the variables. The interface displays five photos associated to the authors "John Singer Sargent" and "James Tissot". Despite not being visible in the mentioned Figure, the single art movement that



Name

Description

Styles: New Realism

Artist: John Singer Sargent

Year: 1910s

Type: Painting

Materials: Paint

Tags: Window, Door, Glass, Sash window, Room

Colors: #535557, #ebede4, #9d9f9a, #dcd674, #f0df80

Figure 6.4: Representation of the screen where the system suggests all the values to the parameters applied to an image of an art piece submitted by the user.

was suggested, “New Realism” was likewise represented via illustrations. The algorithms only proposed one option for the indicated decade (“1910s”) and the suggested type of artwork (“Painting”), therefore it only looked for data linked to that, which was achievable. Although the algorithm was able to indicate materials like “Paint” and “Tints and Shades” the interface does not display any artwork linked to these values because the database at this stage does not include any works of art that contain information about these materials.

This functionality offers comparable photos dependent on the suggested parameters. The major goal of this functionality is to assist the user in updating the data and determining whether or not the similar images are actually comparable to the one that was submitted. The ability to prevent the submission of duplicate works of art is another reason why this functionality is helpful. Users may be able to notice that the image they are trying to upload already exists by showing them photographs of works of art that share similar criteria.

Finally, a submit button is included at the end of this page. When interacting with it, the information that was generated by the algorithms or entered by the user is submitted and saved in the system for further use. This concludes the cataloging process of artwork from a user’s perspective. The interface requires information from an external source in order to carry out all of the proposed operations. The back-end component comes into play here.

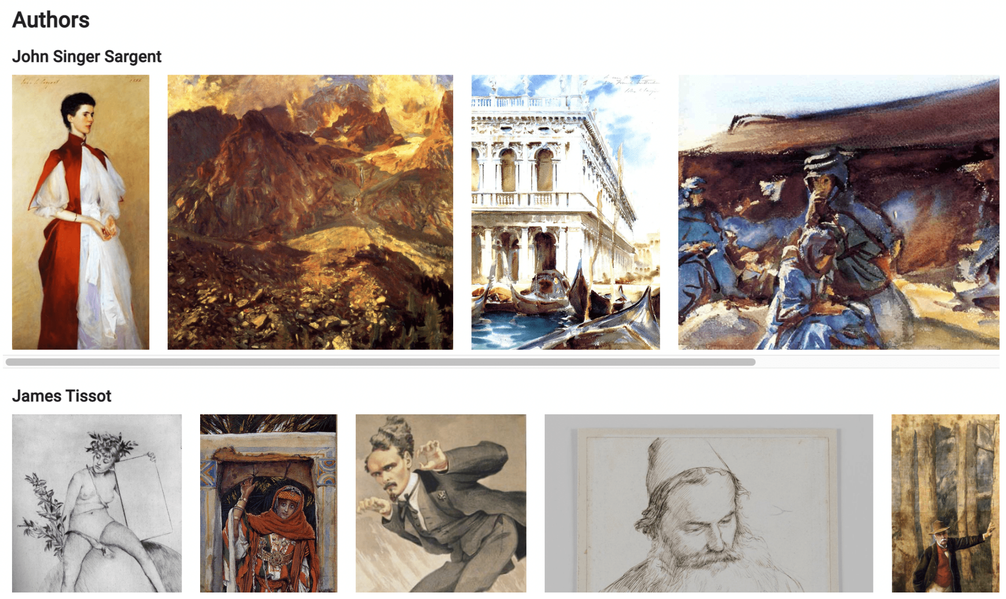


Figure 6.5: Example of the interface behaviour for some of the values related to some parameters when submitting a new image to the platform. Contains a representation of images from the suggested authors.

6.2 Back-end

Every data computation is carried out in the back-end, which then transmits the results to the front-end interface. This is the component that holds the higher number of connections with other components. This component's primary objective is to aggregate all available data sources into a single location. In this instance, communication not only occurs with the front-end component but also with all components involved in data storage and the developed machine learning algorithms. A RESTful API is used for all connections between this and the other components.

When interacting with the interface, the back-end mostly computes user requests, such as requests for searching for works of art. It returns to the user the desired information after interacting with the data storage components. The processing phase of submitting new artwork is also provided via the connection between these two components. When receiving input from the user, it not only provides the user with the predicted value for each parameter but also, upon submission, accepts new data and sends it to the appropriate elements of the platform, which stores the new data.

Information is essentially exchanged between the file storage and database components as part of their connection to the back-end component. Whether it is to write new information or to get data that the user requested. The back-end interacts with both of these components to collect the relevant data when a user searches for an artwork or when similar artwork is needed. The database has all the necessary information. As a result, this is the initial exchange of information. The back-end is also able to locate the image file that belongs to a specific work of art after interacting with the database. Whenever the user wants to input new information about a new work of art, the back-end starts by storing the new image file into the file storage system and then writing a new record in the database.

This element's third key relationship corresponds to the component that houses the fully trained machine learning models. This connection's primary purpose is to send new image predictions to the user. The component that stores the models receives the image that was supplied by the user, through the interface, from the back-end in order to obtain this data. Then, it gets a reply with the appropriate predictions that are then transmitted to the user. Using a RESTful API, this connection is established.

6.3 Database

The database's primary function is to keep information on each work of art that has been registered in the system. The platform's various functionality accesses this information. This information is necessary whether it is to find artwork based on qualities or to gather data from other artworks to retrain the IC algorithms. The database was developed under a relational schema. With this, it would be possible to interact with the data with Structured Query Language (SQL) language. This simplifies the information retrieval procedure. This language was chosen since it is among the most widely used and user-friendly. It enables straightforward data handling in a uniform and standardized way [15]. It is fairly simple to write any necessary queries that could be used to retrieve the data. Utilizing this language also has the benefit of being efficient when performing operations on the data. Two components of this infrastructure have access to this database. The first one is the system's back-end, which can read or write any necessary information depending on the operations made by the user through the interface. The script that is used to train the models is where the other communication is established.

This database's entries each represent a unique piece of art and include all the information that is available related to them. These parameters include fields like the title of the piece of art, the styles used, the Uniform Resource Locator (URL) of the associated image, and more. Only the known, existing, or required information is written in this database because some of these parameters do not need a value. Some of these characteristics are necessary for the system to operate properly. They are the entry's unique identity, the filename, a potential title for the artwork, and the corresponding image URL. The associated art piece's image is subsequently saved in the file storage component and is accessible via the URL value. The remaining parameters are all optional.

6.4 File Storage

The image files that each piece of artwork is represented by are kept in the file storage component. As previously seen, each file corresponds to a particular work of art and is accessible via the URL that is kept in one database column. There are two primary uses for these files. These image files are acquired during data fetching for a variety of reasons, including to be displayed in the interface. This may be required in a variety of circumstances, such as when a user searches for the images. In this instance, the file is provided via the file storage component to the back-end for presentation. When users contribute with new works of art, the new submitted photos are kept in this file storage system.

These files are also used to routinely train the machine learning algorithms. When this is required, the script retrieves the necessary data and retrain the models using the appropriate files.

6.5 Models Storage

The machine learning algorithms that were created in chapter 5 are kept in the model's storage component and are ready for usage. Through a RESTFull API, these elements are called directly from the back-end. Only one specific action uses this module. When a set of suggestions are required for a new image submission. In this scenario, the component receives the raw image from the back-end, that was submitted by the user, and returns a collection of predictions. These recommendations simply provide a set of ten predictions for each of the four problems to the back-end in order to reduce excessive data transfer. Only the pertinent values are then displayed to the user after these values have been processed in the back-end. The user-provided data must undergo some adjustments in order to be comprehended by the prediction models. In order to make the image file compatible with the models, it must be converted into an array of values. The dimensions of this array must be 448x448x3. This value needs to match the one used to build the models, as it was done before in this document in section 5.2.2. Each array value must be modified such that it can range between 0 and 1. All of these modifications are necessary for these models to evaluate an image. This component computes all of these transformations.

6.6 Training Script

Only the data collected thus far was used to carry out the training of the existing models. This indicates that the models are not well-suited for forecasting some values of parameters that they have never encountered. For instance, the author prediction algorithm won't be able to advise an author if there are no submitted art pieces for that author. The other algorithms may suffer from similar problems. This calls for a method of enhancing the algorithms continuously. This is the area of the system where this component is crucial. The models are retrained in this component before being exported and used on the platform. Here, information from the file storage component and the database are gathered, structured, and used to train the models.

A retraining process would be acceptable in two main situations. Whenever an algorithm's ability to predict outcomes can be enhanced through optimization is one of them.

The other reason is that new data can be used in order to increase the number of classes that the model can predict. As a result, it was anticipated that the models continue to be improved over time. The training procedure must be done according to some kind of logic. An approach is to use the back-end to provide a trigger to this system component to start this execution. A manually execution is also a possibility for this process. Using an automated approach, a trigger that is time-related or data-related can be used. On one hand, the back-end would regularly connect with this component to initiate the training process. Since a situation where the training process can be performed using the same data as the prior training process, this approach does not necessarily imply that the models get improved. This would result in the wasteful use of resources. On opposition, there could be a data-related trigger. In this method, the models are retrained using the user-submitted data. The back-end frequently looks for additional data. This component is activated and retrains the models, or a single one, if there is enough data available to be utilized to retrain the model. This strategy unavoidably enhances the algorithms by providing for the utilization of more data. This new information may consist of additional photos belonging to an existing class or a brand-new set of classes. It is expected that the models improves either way.

The use of new data during the retraining process might also demand the application of a new set of optimizations. For this reason, an optimization investigation must be conducted in addition to the training execution process. When the model retraining procedure is accomplished, this component's behavior is complete. The produced models are then made available for usage in the platform.

7 Experimentation and Analysis

After the development of the platform for this thesis has been completed. A detailed examination of it is done in this chapter. It is discussed what was done in accordance with expectations and what tasks presented difficulties throughout the development of this thesis. The development process itself and some of the accomplished actions are also object of analysis. The developed algorithms are primarily tested, and some observations were made in response to the outcomes. It is crucial to note that the vast majority of the goals that were originally outlined for this thesis have been reached.

7.1 Art Features Algorithms

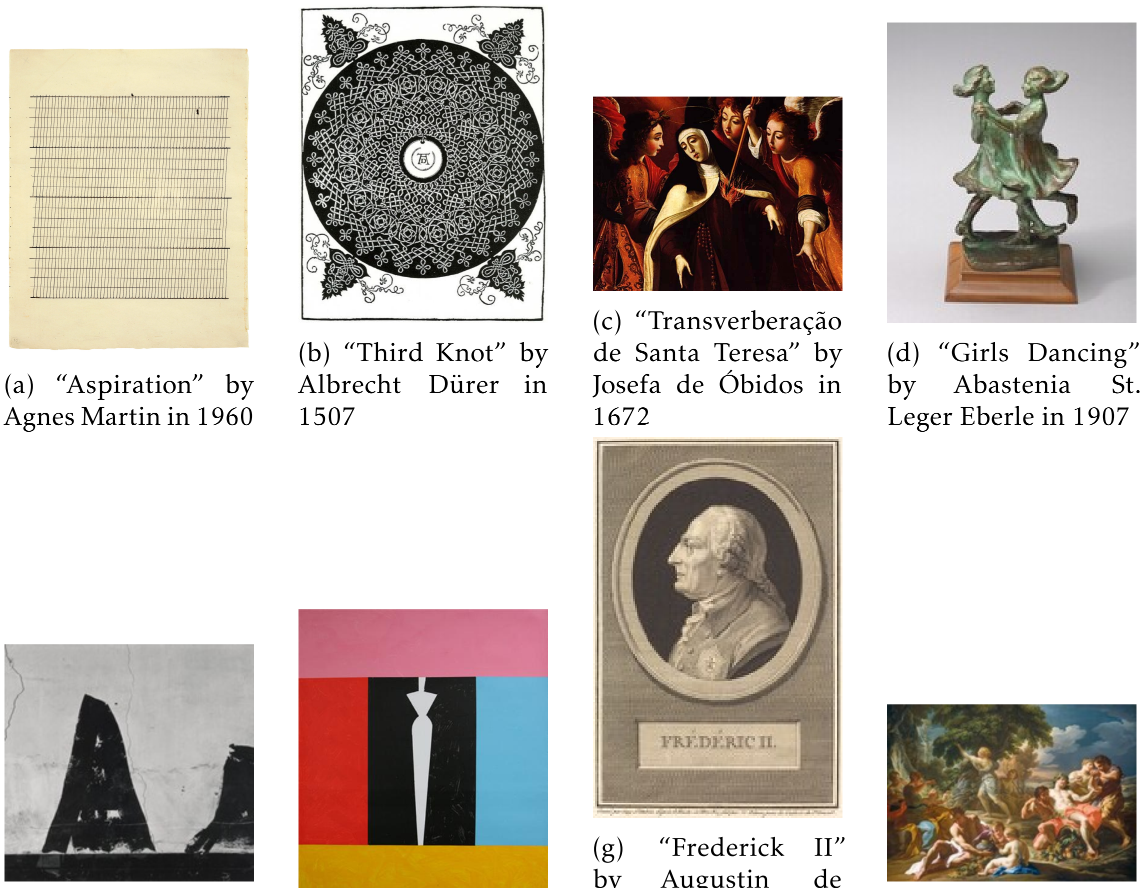
One of the major goals of the platform was to create a tool that could offer a collection of artistic qualities from an image of an artwork. This was the task that took the longest to complete throughout the development of this was this, and it was successfully completed, as was firstly thought. This chapter's analysis and discussion begins with a testing and analysis phase for this first challenge.

Four different IC algorithms were developed. Each one can categorize a work of art according to a different topic. The algorithms were created to indicate the author, the styles of the artwork, the decade of creation, and the type of the piece of art. Not only were the four algorithms developed, but they were also integrated into a user-friendly platform. One of the main goals to be accomplished was the ability to offer a straightforward and understandable approach to use these algorithms. Some experiments were executed in order to better understand how these four methods are actually used. Understanding how the four algorithms operate concurrently is facilitated by the collection of all suggested values that the algorithms generate.

	Known Values				Predicted Values			
Test Image	Style	Author	Type	Year	Style	Author	Type	Decade
7.1a	**Minimalism**	Agnes Martin	-	**1960**	**Minimalism**	Edgar Degas Nicholas Roerich Ernest Ludwig Kirchner Pablo Picasso Utagawa Kuniyoshi	Print Drawing	**1960s** **2000s**
7.1b	Northern Renaissance	**Albrecht Dürer**	-	**1507**	Symbolism	**Albrecht Dürer**	Print Illustration	**1500s**
7.1c	**Baroque**	Josefa de Obidos	-	1672	**Romanticism** **Baroque** **Rococo** **High Renaissance**	Sandro Botticelli	Photograph Painting Illustration	1620s 1630s 1520s
7.1d	-	-	**Sculpture**	1907	Cubism Impressionism	Nicholas Roerich Sandro Botticelli	**Sculpture**	1870s 1520s
7.1e	-	-	**Photograph**	**1979**	Symbolism Expressionism	Vincent Van Gogh Nicholas Roerich	**Photograph**	**1960s** **1970s** **1930s**
7.1f	**Color Filed Painting** **Pop Art**	Aki Kuroda	-	2011	**Pop Art**	Henri Matisse	Print Painting	1960s 1990s 1970s 2000s
7.1g	-	-	**Print**	**1803**	Baroque	Raphael Kirchner	**Print**	**1800s**
7.1h	-	-	**Painting**	**1750**	Baroque Romanticism Mannerism Late Renaissance	Sandro Botticelli	**Painting**	**1760s** **1750s** **1770s**

Table 7.1: Comparison between known and predicted values of eight different tested images of art pieces

The outcomes of a few experiments that were run using various images are displayed in Table 7.1. Eight distinct images of works of art were uploaded to the platform during the testing procedure. While the values of some of these image files' parameters were already known, others were not. The major goal of these tests was to determine whether the algorithms could accurately detect the known labels. Understanding the quantity of additional information provided for each submitted photograph is another conclusion that was expected to be done from this testing phase. The eight selected images were picked with consideration in order to test the platform with as many different scenarios as possible. These images are represented in Figure 7.1



(a) "Aspiration" by Agnes Martin in 1960

(b) "Third Knot" by Albrecht Dürer in 1507

(c) "Transverberação de Santa Teresa" by Josefa de Óbidos in 1672

(d) "Girls Dancing" by Abastenia St. Leger Eberle in 1907

(e) "Lima 306" by Aaron Siskind in 1979

(f) "Night" by Aki Kuroda in 2011

(g) "Frederick II" by Augustin de Saint-Aubin after Barthelemy Blaise in 1803

(h) "Autumn" by Corrado Giaquinto in 1750

Figure 7.1: Set of images used to test the machine learning algorithms

The first (7.1a), third (7.1c), and sixth (7.1f) images that were put to the test relate to works of art by authors whose names the author prediction algorithm does not know. This test's major objective is to demonstrate that the algorithm can never accurately guess these authors. The cause of this circumstance is that there was insufficient information regarding these authors in the data used to train the author prediction algorithm. This author detection tool can currently exclusively identify 68 different authors at this level of development. However, the system recommends the authors who are most likely to have created the submitted artwork. It is evident in these three cases that the majority of the parameters were correctly recognized by the other methods. The style, author, and year were the three known fields in the first (7.1a) scenario. The other two criteria were accurately recommended, with the author being the sole exception, as was just stated. The algorithm also proposed “Print” and “Drawing” as choices for the type parameter, which were unknown. In the other two mentioned tests, the style of the piece of art was correctly guessed, from the predicted classes.

The sixth (7.1f) example was also created to comprehend how the algorithms behave with a work of art that is representative of several different art movements. The known values for the style of the piece of art are “Color Filed Painting” and “Pop Art” as can be seen in the table. Only one of the styles, “Pop Art” was recognized by the style algorithm when this file was submitted to the platform. Even though the algorithm could only identify one of the known labels, it was the only predicted class, which indicates that the result was produced with a high degree of probability compared to the other possible classes.

The second (7.1b) scenario is one in which a work of art by an author who is known to the algorithm is submitted. “Albrecht Dürer” is the name of the artist who created this work of art. This is the only creator the algorithm suggests when this image was uploaded to the platform. This indicates that the algorithm had a high likelihood of correctly predicting the value. The year parameter saw the same outcome in this instance as well. This field’s known value is 1507. The single suggestion made is accurate even though the time-related algorithm can only advise decades, as it was discussed in this thesis in section 4.2.2. The seventh (7.1g) case similarly involves this situation, however, instead of correctly identifying the author, the algorithms were able to detect the type of art piece.

The platform’s type detection algorithm was tested using the fourth (7.1d), sixth (7.1f), seventh (7.1g), and eighth (7.1h) scenarios. It was possible to draw the conclusion that this is the most accurate algorithm when developing it, as concluded in chapter 5. The system was able to accurately identify the classes in each of the four scenarios while also only suggesting one value in each case. This indicates that the anticipated values had a higher likelihood than any other potential values. These outcomes validate the algorithm’s high degree of accuracy, which was already anticipated. It is also important to mention that for each of these tested scenarios, the algorithm always provided different prediction values, which means that the confidence level is high across every predictable class.

Not only do the type and decade year properly identify the known values in the eighth (7.1h) and final examined circumstance, but it is also feasible to see that the decade model predicts a sequence of values that are extremely similar to one another and sequential. The algorithm suggests the decades of 1750s, 1760s, and 1770s, as evident. This suggests that there is a similarity between the works of art produced over this group of decades. Although this last scenario makes this effect more obvious, it is also possible to observe a replication of this in other tested images, such as the third (7.1c), fifth (7.1e), and sixth (7.1f).

Overall, it is clear that the algorithms were able to correctly identify at least one parameter in each investigated circumstance. One other observation is that every field contains a set of recommended values for each parameter in every circumstance. There are numerous instances where the suggested values are incorrectly guessed, despite the fact that the models' prediction abilities are sufficient to accurately anticipate at least one parameter in every circumstance. The one that stands out the most is when the algorithm is asked to forecast a class that it has never seen or been trained with. Retraining the models with all of the current classes is the only way out of this dilemma. However, this results in a situation where some classes have a small quantity of images to train on. Because of this, increasing the number of trainable classes is the greatest method to have more predictable classes. This is what is anticipated to occur when users use the platform. The platform gathers more data that can be used to retrain the models. In addition to producing more classes that can be suggested by the algorithms, this retraining process is anticipated to produce more precise predictions.

7.2 Interlinked Art

The connection between works of art was another feature that was initially though to be developed. This feature was successfully created, just like the feature for extracting art features. The platform employs this interlinking process. Users have the option to examine related works of art from several parameters when they submit an image to classification. This feature makes it feasible to do that. This interlinking functionality is examined in depth and put to the test in this section. In the end, it is clear if this technology has effective interlinking capabilities, that is, if it accurately displays images that are comparable to those provided.

An image is uploaded to the platform and similar images are requested in order to carry out this testing scenario. The platform's returned image collection is examined. The image that was uploaded to the platform is displayed in Figure 7.2.

The submitted image is known to be a 1945 Salvador Dali painting. The algorithms correctly recognized the author and suggested two decades when the photograph was submitted to the platform, with one of the proposed decades being correct (1940s), and the other one being 1930s. The "Painting" label was also correctly recognized by the type detection algorithm. The painting was also categorized as "Expressionism" and "Naive Art Primitivism". When requested, the platform provides with multiple sets of images of works of art that shares the identified values for the various factors. Figure 7.3 shows these collections of photos. The algorithms' recommendations are the values that determine the groups of items that are displayed in this feature. In addition, the platform offers a collection of up to five photos for each value. It is anticipated that this set of photographs provide users with a set of images that are comparable to the one they

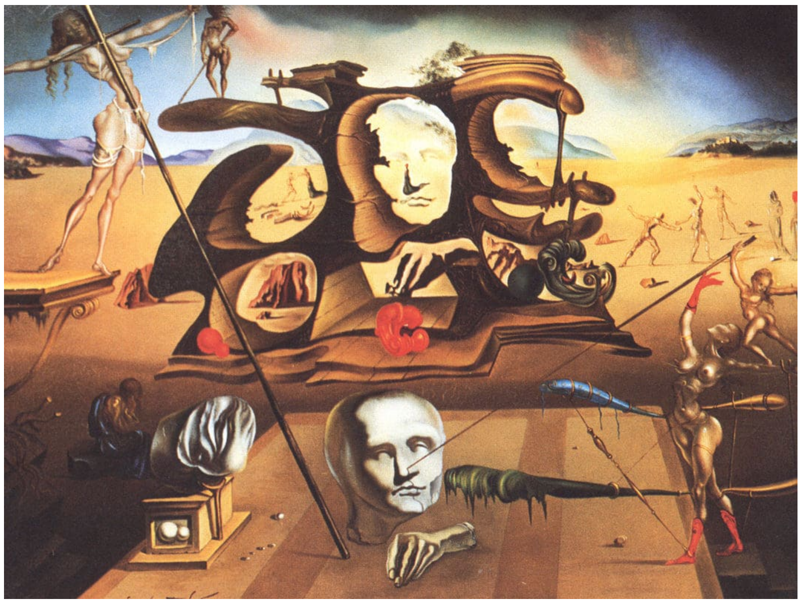


Figure 7.2: "Napoleon's Nose, Transformed into a Pregnant Woman, Strolling His Shadow with Melancholia amongst Original Ruins" by Salvador Dali in 1945. Image used to test the interlinking feature developed in the platform.

supplied. The effectiveness of this strategy for displaying related images are assessed through this analysis. The algorithm successfully suggested the majority of the examined classes for the provided image. This implies that certain similarities between the observed and expected findings are expected. The offered results, however, might not be the best to match with the submitted image if the algorithm offers wrong classes.

When comparing the results, it becomes clear that the most similar collection of images provided by the platform are those that share the same author and style. The other collections of photographs do not closely resemble the submitted image, although having the same decade or type of art. However, with this interlinking feature, there are other aspects of an art piece that are anticipated to be observable in addition to visual resemblance. For example, the type of the piece of art can be a good parameter to interlink multiple artworks. This parameter's primary objective is to display related types of art, such as painting or sculpture, rather than a visual resemblance. This, however, does not mean that it is less or more relatable to the original piece. For this reason, the images that are expected to have the most visual resemblance are therefore those with the same author or style. On the other hand, the other parameters are useful to interlink an image with others by other features. The other Salvador Dali's artworks can be seen to have some aesthetic resemblance to the one that was submitted for this test. Due to the possibility that some authors could create wildly divergent works of art, this is a circumstance that cannot be validated throughout the entire body of work of an artist. By applying this reasoning, it is anticipated that while analyzing the shared style, the best visual commonalities are discovered. Which is true in this case, and it is very certainly proven in practice. Still, in this tested scenario, by analyzing the other generated data, it is possible to recognize the relationship between the provided image and the images that were displayed. In this case, both the submitted image and the set of images are all paintings. Analysing the decade parameter, even if there doesn't seem to be any connection between the submitted image and the ones that were produced, there may be a more subjective connection, such as historical, use of techniques, materials or any other factor.

In conclusion, even though this function doesn't always offer the closest visual resemblance to the submitted work of art, it is nevertheless significant to display these linked items to the user because there must be a more subjective connection between them. The author's and the artist's styles, though, were the ones that were most visually similar among the criteria utilized for this association.

7.3 Classification of Algorithm's Development

In contrast to the preceding section, which focused on discussing and analyzing the outcomes of the tools and algorithms created, this section provides an overview of the art classification models themselves. It is highlighted some anticipated aspects as well as some challenges and obstacles encountered during the course of their development. Finally, a brief comparison to the initially studied projects, was done.

Overall, the accuracy of prediction of the models was consistent with what was anticipated. The type detection method was the model with the best accuracy. The model that was anticipated to do this was this one. On the other side, the decade algorithm was the model whose accuracy was predicted to be the lowest, which also occurred. However, it was not anticipated that this particular problem would have accuracy levels this low. This was one of the challenges encountered in the creation of the suggested algorithms. The model's accuracy did not have much improvement even after a series of optimizations. However, even with a lower prediction accuracy, the model could consistently identify the decade of the works of art in various situations, as previously tested in subsection 7.1. Overall, the models are capable of making accurate predictions in practical scenarios. An analysis of the Table 7.1 shows this fact. The values that were successfully predicted are highlighted in this table. It is clear that at least one class was properly predicted for each case evaluated. Another observation that may be made is that the models can always give further information about the images when a field is unknown. Even though not included in the table, the platform also offers details on a number of tags and colors that were found in the images. All of this information is crucial for cataloging works of art.

During the development of the models, the use of a variety of data augmentation techniques to the data was one aspect that went against to what was originally anticipated. Initially, it was expected that these data enhancements would increase the models' accuracy since they would reflect a greater variety of possible scenarios in everyday life. For instance, adding rotation or zoom changes to the data might be seen as a simulating of real-world conditions. Hence, the models should theoretically be improved since it would be ready for this kind of scenarios. However, when applying these adjustments, only some of the suggested image classification algorithms was improved. In certain circumstances, the model's prediction skills deteriorated with these techniques. With the occlusion approach, this impact was equally apparent.

With this work, it was possible to develop four different image classification algorithms that were able to classify art pieces. An author prediction algorithm with a Top-1 accuracy of 83.71%; a style prediction algorithm with a Top-1 accuracy of 63.94%; a decade prediction algorithm with a Top-1 accuracy of 38.75%; and finally, a type of art prediction algorithm with a Top-1 accuracy of 92.87% were developed. When comparing these values to the state of the art projects that were initially studied in chapter 3, it can be said that the developed work outperformed them.

Making a deeper comparative analysis, the model that was developed for the author prediction from images of pieces of art, is able to forecast through more classes (being able to detect from a set of 68 different classes, while the studied project could only predict from a set of 57 different authors). Along with this enhancement, the created method in this thesis significantly increased prediction accuracy. More than 6 percent of accuracy could be achieved. As noted in subsection 3.1.1, a larger dataset, a different architecture and set of optimizations were applied for the construction of this classification model. The impact shown in the author classification model is also seen in the style recognition challenge, but to a smaller degree. In comparison to the project that was initially under study in subsection 3.1.2, a higher number of styles could be predicted, and the prediction accuracy was improved from a value of 62% to roughly 64%.

While both of the aforementioned methods were improved, the major objective of this thesis was to create a tool that could compile data acquisition from several sources into a single place. That was able to do with the aid of the four established algorithms. As originally believed, extra data could be obtained from the existing image recognition technologies in addition to the created methods. A centralized data gathering platform for works of art was established using all these components that were created for this thesis. And in that regard, the suggested work may be considered to have been effectively developed since every main goal outlined at the start of this thesis was achieved.

When analysing the overall platform, a technology that could extract this much information information, such as the author, style, type of art, time-related information, materials, labels and a list of the most prominent colors, from images of works of art was developed. This was not the case of any other studied platform in chapter 3. For this reason, it can be concluded that the existing work was outperformed by the platform that was developed during this thesis development.

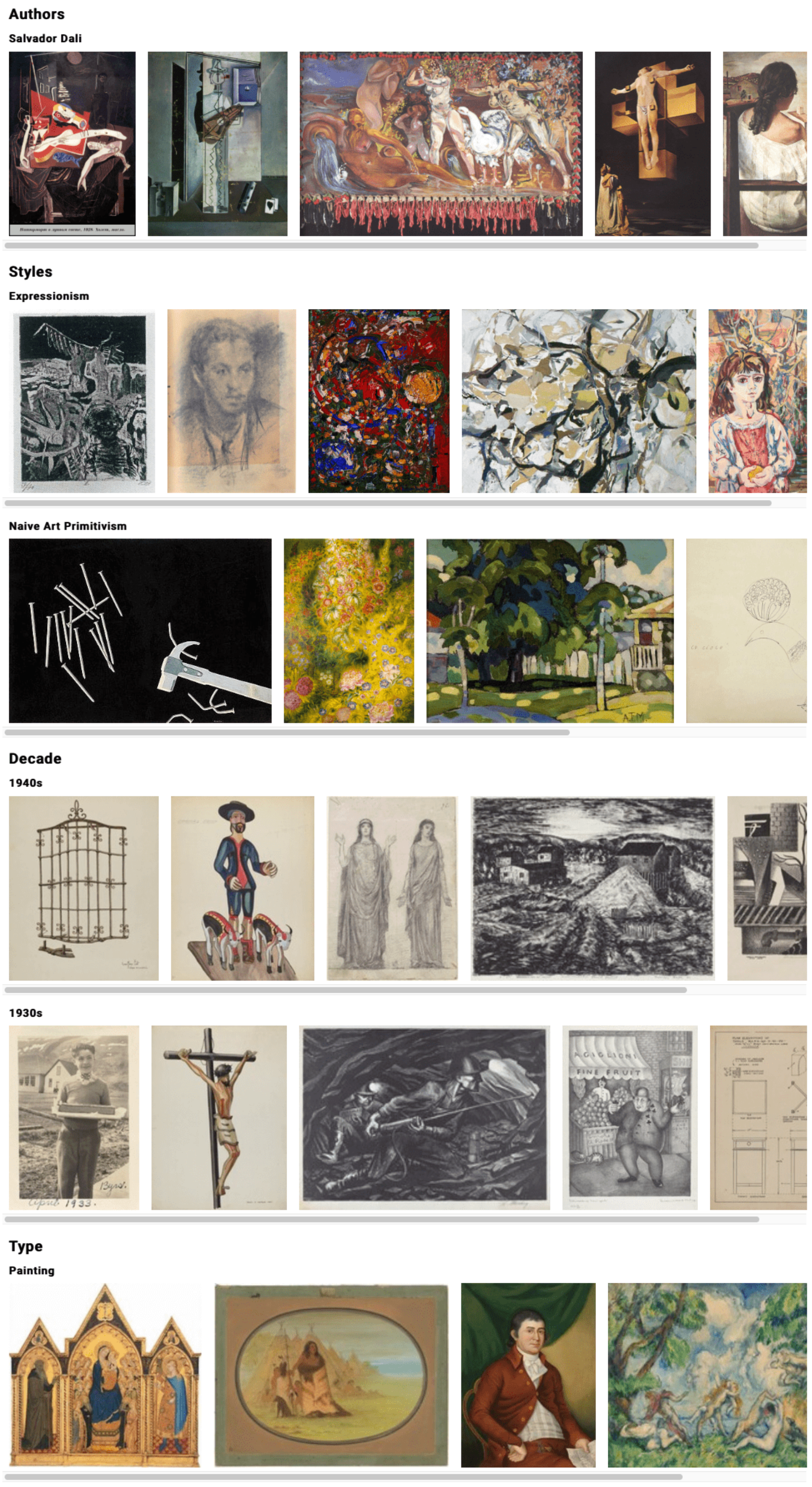


Figure 7.3: Similar pieces of art by different parameters related to the submitted image of the art piece represented in the Figure 7.2.

8 Conclusions and Future Work

Following the development of this master's thesis proposed project, and a brief analysis of it, the conclusions are now made available. The findings of this work are analyzed and discussed in this chapter. There are also some suggestions for potential future improvements of the developed work. The effectiveness of the prediction models and the methodologies employed may both benefit from this. Complementary work that is relevant to the continuous growth of the platform created is discussed at the end of the chapter.

8.1 Conclusions

The conducted work in this thesis examined a variety of machine learning techniques and algorithms in order to construct a tool that can categorize works of art based only on their visual representations. The probable author, styles, decade, and type of art were the criteria that were picked to be taken using IC algorithms. Other features, such as the materials used, most dominant colors, and a set of relatable labels could also be identified from external tools. With this, a vast set of information can be suggested to any user that tries to catalog a piece of art. The handling of the data was an essential aspect during the produced work. The information was gathered from various sources and had to be transformed into a common format so that the several created algorithms could analyze it. This data could then be used in all the proposed features of this platform, such as the cataloguing process, interlinking and representation of pieces of art.

One primary goal of this thesis was to use data to create the art classification algorithms. To achieve this, machine learning models were trained. CNN was employed to build these models. After a careful analysis on a few possible classifiers, mainly the K-NN, LC, and NN, this was the one that was chosen since it has proven to have the best capabilities to classify wide variety of images. Additionally, an investigation on the best architecture to be used and on the various methodologies was carried out and based on each problem. All the created models were optimized according to their context. With this, their performance was improved when compared to their initial prediction's accuracy.

Overall, for the situations of author recognition problem, which was the one that was first researched, the algorithm that was built showed good prediction results when applied to a realistic scenario, being able to accurately predict on the expected tested scenarios, such as in sections 7.1 and 7.2. This effect was also visible in all the other developed models, as concluded by the testing analysis done in section 7.1. The developed work allowed for the development of four prediction algorithms, which was what was initially proposed. Additionally, as an original hypothesis suggested, more data could be collected from the images of the works of art using other ML technologies, as concluded in section 5.5. This would make it possible to extract a vast amount of information and aid in the process of cataloging works of art. The platform was developed in a way that the developed IC models get continuously improved with its utilization. Thanks to the submission of new images, the models can be retrained and be improved not only by being able to predict over more classes, but their accuracy is also expected to be improved. Finally, a complete infrastructure was developed to incorporate all the components necessary for the platform to operate properly and to support all of its functionalities, including the interlinking and representation features. Thanks to this, it was possible, as initially proposed, to develop a platform that could not only offer the possibility to easily catalog pieces of art, but also to represent and interlink them. It is possible to conclude that this platform is an improvement to the others that were also studied. It adds social value by providing ubiquitous access to art as well as by reducing the barrier between the users and the engagement and cataloging of works of art.

8.2 Future Work

The development of this work and the findings it produced led to a better understanding of how to make an image classification tool that can categorize images. Applicable to art pieces in this instance. The cataloging procedure, that is one of the main features of the created platform, is supported by this technique. There are several improvements that may be made for this platform's future development. The author recognition problem, among the four proposed models, was the one that received a deeper analysis. The findings from this process were then reused to the other models. For this reason, some parameters were not deeply investigated (such as the model's architecture, the batch size, or the learning rate approach). Hoping to increase the other model's accuracy, this deep investigation process could also be done to them. The decade prediction model, for example, is one of the models where this study may have a bigger improvement because of its lower accuracy, when compared to the others. This accuracy differences may be related to the number of predictable classes as well as the problem's complexity, as concluded in sections 5.4.2 and 7.2.

Hardware was one of the constraints encountered while this project was being developed. Most of the training stages took many hours to complete since the hardware that was utilized to train the models allowed for longer executions. The amount of data, the set of optimizations and formatting methodologies, in combination with the hardware computing power, are what cause this time complexity problem. Faster training phases would result from better hardware, since the data is required and should not be compromised. This would also suggest that the use of the Keras Tuner tool may be investigated in further detail. With the existing hardware setup, each run of this tool required a substantial amount of time, sometimes, multiples days of execution. Therefore, with higher computational power, more hyperparameter options could be investigated. A higher batch size value in the model's training phase may be another advantage of more powerful hardware. As concluded in subsection 5.3.5, this has proven to be beneficial for the accuracy forecast of the model.

The interlinked feature was created in a data-dependent approach. In other words, the system only searches for data based on the values of the parameters, such as authors, styles, and more. Future research might be done on a different strategy, such looking exclusively for images that are identical to the submitted image. This would probably be of considerable assistance in identifying duplicate images, which was a weak point of this thesis's development. Still relevant to the future growth of this work, a more robust interface may be created, as just the essential elements and functionalities were used for this phase of development. Future developments may also include the creation of functionality that are user-dependent, such as the possibility to have a collection of a favorite artworks or other individualized features. The developed work is exclusively compatible with images of works of art. Meaning that the created algorithms can only classify images. Extending its compatibility to additional media types, such videos, might be a potential improvement. It is also possible to create an audio compatibility in which data is taken from an audio source. More data from works of art, such as the dimensions of a painting or sculpture, might be retrieved using modern hardware tools like LIDAR sensors, which are becoming a common technology in smartphones. With the utilization of the platform, information that is now gathered by external tools can start to be predicted with custom IC models. Mainly the materials of the art piece.

The goal of all the work done throughout the production of this thesis and the recommended enhancements is to obtain as much accurate and comprehensive information as possible regarding works of art. This assists in the cataloging process of the pieces of art and proving more data to be represented and interlinked.

Bibliography

[1] M. Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 265–283. ISBN: 978-1-931971-33-1. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (cit. on p. 34).

[2] H. A. Abu Alfeilat et al. “Effects of distance measure choice on k-nearest neighbor classifier performance: a review”. In: *Big data* 7.4 (2019), pp. 221–248 (cit. on p. 8).

[3] S. Albawi, T. A. Mohammed, and S. Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186 (cit. on pp. 15, 16).

[4] Allegra. *Art Detective: AI for Art Style Recognition*. Oct. 2020. URL: https://medium.com/swlh/art-detective-ai-for-art-style-recognition-4632e05c3496 (visited on 01/07/2022) (cit. on p. 20).

[5] *An Overview of ResNet and its Variants*. 2022. URL: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035 (visited on 05/16/2022) (cit. on p. 36).

[6] E. R. Arboleda, A. C. Fajardo, and R. P. Medina. “Classification of coffee bean species using image processing, artificial neural network and K nearest neighbors”. In: *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. IEEE. 2018, pp. 1–5 (cit. on p. 8).

[7] J. Bailey. *Can AI Art Authentication Put An End To Art Forgery?* Sept. 2019. URL: https://www.artnome.com/news/2019/9/12/can-ai-art-authentication-put-an-end-to-art-forgery (visited on 01/07/2022) (cit. on p. 20).

[8] F. Bre, J. Gimenez, and V. Fachinotti. “Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks”. In: *Energy and Buildings* 158 (Nov. 2017). DOI: 10.1016/j.enbuild.2017.11.045 (cit. on p. 14).

[9] R. Caruana and A. Niculescu-Mizil. "An empirical comparison of supervised learning algorithms". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 161–168 (cit. on p. 8).

[10] Cs-Chan. *ArtGAN/WikiArt dataset at master · CS-chan/ArtGAN*. URL: https://github.com/cs-chan/ArtGAN/tree/master/WikiArt%5C%20Dataset (visited on 04/11/2022) (cit. on p. 23).

[11] F. Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195 (cit. on p. 36).

[12] F. Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258 (cit. on p. 35).

[13] *CIFAR-10 and CIFAR-100 datasets*. URL: https://www.cs.toronto.edu/~kriz/cifar.html (visited on 12/14/2021) (cit. on pp. 8, 13).

[14] M. Cilimkovic. "Neural networks and back propagation algorithm". In: *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin* 15 (2015), pp. 1–12 (cit. on p. 15).

[15] R. Deari et al. "Analysis and comparison of document-based databases with sql relational databases: Mongodb vs mysql". In: *Proceedings of the International Conference on Information Technologies*. 2018, pp. 1–10 (cit. on p. 72).

[16] Z. Deng et al. "Efficient kNN classification algorithm for big data". In: *Neurocomputing* 195 (2016), pp. 143–148 (cit. on p. 9).

[17] T. DeVries and G. W. Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017) (cit. on p. 45).

[18] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. "How does the brain solve visual object recognition?" In: *Neuron* 73.3 (2012), pp. 415–434 (cit. on p. 2).

[19] *Evaluation of art authenticity made easy*. Sept. 2021. URL: https://art-recognition.com/ (visited on 01/07/2022) (cit. on p. 19).

[20] *Global art market volume of transactions from 2007 to 2021*. 2022. URL: https://www.statista.com/statistics/885518/global-art-market-volume-of-transactions/ (visited on 09/26/2022) (cit. on p. 1).

[21] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016 (cit. on p. 25).

[22] W. Hao et al. "The Role of Activation Function in CNN". In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. 2020, pp. 429–432. DOI: 10.1109/ITCA52113.2020.00096 (cit. on p. 47).

[23] T. He et al. “Bag of tricks for image classification with convolutional neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 558–567 (cit. on p. 52).

[24] S. Hijazi, R. Kumar, C. Rowen, et al. “Using convolutional neural networks for image recognition”. In: *Cadence Design Systems Inc.: San Jose, CA, USA* 9 (2015) (cit. on p. 1).

[25] Icaro. *Best artworks of All time*. Mar. 2019. URL: https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time (visited on 03/09/2022) (cit. on p. 22).

[26] *Illustrated: 10 CNN Architectures*. 2022. URL: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d (visited on 09/30/2022) (cit. on p. 35).

[27] *Image classification with tensorflow hub*. URL: https://www.tensorflow.org/hub/tutorials/image_classification (visited on 02/17/2022) (cit. on p. 20).

[28] *ImageNet*. URL: https://www.image-net.org/index.php (visited on 05/17/2022) (cit. on p. 38).

[29] R. H. Johnsen and A. Gradevcak. “Deep Learning Approaches to Art Style Recognition in Digital Images”. In: () (cit. on p. 41).

[30] *K-Nearest Neighbors Demo*. URL: http://vision.stanford.edu/teaching/cs231n-demos/knn/ (visited on 12/16/2021) (cit. on p. 10).

[31] I. Kanellopoulos and G. G. Wilkinson. “Strategies and best practice for neural network image classification”. In: *International Journal of Remote Sensing* 18.4 (1997), pp. 711–725 (cit. on p. 13).

[32] J. Konar, P. Khandelwal, and R. Tripathi. “Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network”. In: *2020 IEEE International Students’ Conference on Electrical,Electronics and Computer Science (SCEECS)*. 2020, pp. 1–5. DOI: 10.1109/SCEECS48394.2020.94 (cit. on pp. 49, 50).

[33] A. Lecoutre, B. Negrevergne, and F. Yger. “Recognizing Art Style Automatically in Painting with Deep Learning”. In: *Proceedings of the Ninth Asian Conference on Machine Learning*. Ed. by M.-L. Zhang and Y.-K. Noh. Vol. 77. Proceedings of Machine Learning Research. Yonsei University, Seoul, Republic of Korea: PMLR, 15–17 Nov 2017, pp. 327–342. URL: https://proceedings.mlr.press/v77/lecoutre17a.html (cit. on pp. 1, 19, 42).

[34] J. M. Lourenço. *The NOVAthesis LaTeX Template User’s Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[35] *Luminism Movement Overview*. URL: https://www.theartstory.org/movement/luminism/ (visited on 12/07/2021) (cit. on p. 2).

[36] J. Morris. *Five valuable works of art discovered in people's attics and garages*. Oct. 2022. URL: https://www.archute.com/valuable-art-discovered-attics-garages/ (visited on 10/05/2022) (cit. on p. 1).

[37] *Most used programming languages among developers worldwide as of 2022*. 2022. URL: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/ (visited on 09/26/2022) (cit. on p. 34).

[38] P. Mulak and N. Talhar. "Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset". In: *Int. J. Sci. Res* 4.7 (2015), pp. 2319–7064 (cit. on pp. 8, 9).

[39] X. Ning et al. "HCFNN: high-order coverage function neural network for image classification". In: *Pattern Recognition* 131 (2022), p. 108873 (cit. on p. 14).

[40] *Number of internet users worldwide from 2005 to 2021*. 2022. URL: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/ (visited on 09/26/2022) (cit. on p. 2).

[41] T. O'Malley et al. *KerasTuner*. https://github.com/keras-team/keras-tuner. 2019. (Visited on 05/09/2022) (cit. on p. 48).

[42] K.-S. Oh and K. Jung. "GPU implementation of neural networks". In: *Pattern Recognition* 37.6 (2004), pp. 1311–1314 (cit. on p. 35).

[43] *Open data at the National Gallery*. URL: https://www.nga.gov/open-access-images/open-data.html (visited on 04/27/2022) (cit. on p. 23).

[44] F. Osisanwo et al. "Supervised machine learning algorithms: classification and comparison". In: *International Journal of Computer Trends and Technology (IJCTT)* 48.3 (2017), pp. 128–138 (cit. on p. 8).

[45] F. Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830 (cit. on p. 34).

[46] Ö. Polat. "Detection of Covid-19 from Chest CT Images using Xception Architecture: A Deep Transfer Learning based Approach". In: *Sakarya University Journal of Science* 25.3 (2021), pp. 813–823 (cit. on p. 36).

[47] L. Qian et al. "Sequence-dropout block for reducing overfitting problem in image classification". In: *IEEE Access* 8 (2020), pp. 62830–62840 (cit. on p. 37).

[48] P. M. Radiuk. "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets". In: (2017) (cit. on p. 52).

[49] E. Rezende et al. "Malicious software classification using transfer learning of resnet-50 deep neural network". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2017, pp. 1011–1014 (cit. on p. 35).

[50] Rismiyati et al. “Xception Architecture Transfer Learning for Garbage Classification”. In: *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*. 2020, pp. 1–4. DOI: 10.1109/ICICoS51170.2020.9299017 (cit. on pp. 35, 36, 38).

[51] S. Roy. “Factors influencing the choice of a learning rate for a backpropagation neural network”. In: *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94)*. Vol. 1. IEEE. 1994, pp. 503–507 (cit. on p. 49).

[52] A. A. M. Al-Saffar, H. Tao, and M. A. Talab. “Review of deep convolution neural network in image classification”. In: *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*. 2017, pp. 26–31. DOI: 10.1109/ICRAMET.2017.8253139 (cit. on pp. 15, 16).

[53] S. Saha. *A comprehensive guide to Convolutional Neural Networks-the eli5 way*. Dec. 2018. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (visited on 02/17/2022) (cit. on p. 15).

[54] R. Saravanan and P. Sujatha. “A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification”. In: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2018, pp. 945–949 (cit. on p. 8).

[55] M. Sharma, R. Pal, and A. K. Sahoo. “Indian sign language recognition using neural networks and KNN classifiers”. In: *ARPN Journal of Engineering and Applied Sciences* 9.8 (2014), pp. 1255–1259 (cit. on p. 8).

[56] S. Sharma, S. Sharma, and A. Athaiya. “Activation functions in neural networks”. In: *towards data science* 6.12 (2017), pp. 310–316 (cit. on pp. 14, 15, 47).

[57] *Smartphone market share from March 2014 to December 2015, by camera resolution*. 2022. URL: https://www.statista.com/statistics/619743/smartphone-market-share-by-camera-megapixels/ (visited on 09/26/2022) (cit. on p. 7).

[58] *Smartphones Cause Photography Boom*. 2022. URL: https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/ (visited on 09/26/2022) (cit. on p. 4).

[59] N. Srivastava. “Improving neural networks with dropout”. In: *University of Toronto* 182.566 (2013), p. 7 (cit. on p. 47).

[60] *Subjectivity and Objectivity in Art*. 2022. URL: https://christopherpjones.medium.com/subjectivity-and-objectivity-in-art-cc41d55c76a5 (visited on 09/26/2022) (cit. on p. 6).

[61] F. Sultana, A. Sufian, and P. Dutta. "Advancements in Image Classification using Convolutional Neural Network". In: *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. 2018, pp. 122–129. DOI: 10.1109/ICRCICN.2018.8718718 (cit. on p. 16).

[62] R. Suwanda, Z. Syahputra, and E. Zamzami. "Analysis of euclidean distance and manhattan distance in the K-means algorithm for variations number of centroid K". In: *Journal of Physics: Conference Series*. Vol. 1566. 1. IOP Publishing. 2020, p. 012058 (cit. on p. 10).

[63] A. C. Tan and D. Gilbert. "Ensemble machine learning on gene expression data for cancer classification". In: (2003) (cit. on p. 8).

[64] Tate. *Cubism – Art Term*. URL: https://www.tate.org.uk/art/art-terms/c/cubism (visited on 12/07/2021) (cit. on p. 2).

[65] E. Team. *Why the time complexity for training K-nearest neighbors is O(1)*. Feb. 2021. URL: https://medium.com/nerd-for-tech/why-the-time-complexity-for-training-k-nearest-neighbors-is-o-1-5b8f417104cf (visited on 02/17/2022) (cit. on p. 9).

[66] K. Team. *Keras documentation: MobileNet and MobileNetV2*. URL: https://keras.io/api/applications/mobilenet/ (visited on 01/17/2022) (cit. on p. 20).

[67] *Vision AI | Derive Image Insights via ML | Cloud Vision API | Google Cloud*. URL: https://cloud.google.com/vision (visited on 11/21/2022) (cit. on p. 20).

[68] N. Viswanathan et al. "Artist identification with convolutional neural networks". In: *Standford193CS231N Report* (2017) (cit. on pp. 1, 18).

[69] *Was famed Samson and Delilah really painted by Rubens? No, says AI*. Sept. 2021. URL: https://www.theguardian.com/artanddesign/2021/sep/26/was-famed-samson-and-delilah-really-painted-by-rubens-no-says-ai (visited on 01/07/2022) (cit. on pp. 19, 20).

[70] M. Wecker. *7 of the greatest long-lost art historical masterpieces that were found in attics and basements-ranked*. May 2021. URL: https://news.artnet.com/art-world/art-found-in-attics-ranked-1962993 (visited on 09/06/2022) (cit. on p. 1).

[71] L. G. Wright et al. "Deep physical neural networks trained with backpropagation". In: *Nature* 601.7894 (2022), pp. 549–555 (cit. on p. 11).

[72] K. Yasaka et al. "Deep learning with convolutional neural network in radiology". In: *Japanese journal of radiology* 36.4 (2018), pp. 257–272 (cit. on p. 16).

[73] F.-F. L. J. J. S. Yeung. *Image Classification pipeline*. URL: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture2.pdf (visited on 12/14/2021) (cit. on pp. 8–13).

[74] *Your machine learning and Data Science Community*. URL: https://www.kaggle.com/ (visited on 03/09/2022) (cit. on p. 22).