Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University Mohamed Khider of Biskra

Faculty of Exact Sciences and Science of Nature and Life
Department of Matter Sciences

Ref: ……..

Thesis Presented to obtain the degree of
Doctorate in computer science
Option: Image and Artificial Life

Entitled:

Recognizing the artistic style of fine art paintings with deep learning for an augmented reality application

Presented by:
Baya Lina MENAI

Publicly defended on: 07/12/ 2023

In front of the Jury committee composed of:

Mr. CHERIF Foudil	Professor	University of Biskra	President
Mr. BABAHENINI Mohamed Chaouki	Professor	University of Biskra	Supervisor
Mr. MELKEMI Kamel Eddine	Professor	University of Batna 2	Examiner
Mr. ABBAS Fayçal	MCA	University of Khenchela	Examiner
Mr. DJEFFAL Abdelhamid	Professor	University of Biskra	Examiner

Acknowledgments

First and foremost, thanks and praise go to ***ALLAH*** at the beginning and the end. I thank Allah for giving me the health and well-being to accomplish this work.

I would like to express my sincere gratitude to my supervisor Professor ***Mohamed Chaouki BABAHENINI***, who provided me with constant guidance, motivation, and expertise. His support and belief in my abilities have been helpful in overcoming challenges throughout this journey.

I would also like to extend my gratitude to all my friends and colleagues in the LESIA laboratory, especially ***Fatma Zohra OUAMENE, Rima BENELMIR, and Romaissa BEKIRI*** for their support throughout the realisation of this work.

Lastly, I would like to thank my family for their support, understanding, and encouragement throughout this journey. Their love and belief in me have been the driving force behind my determination and perseverance.

To all those who have been a part of this research journey, I am truly grateful for your contributions and support. Thank you for being there every step of the way.

Baya Lina

Dedication

I am honored to dedicate this thesis to my dear parents, who have been my source of love, encouragement, and support throughout my academic journey. Your belief in me and your constant encouragement has been the driving force behind my accomplishments, and I am forever grateful for your prayers.

I would like to dedicate this thesis to my sisters, ***Nejla, Yesmine, Nesrine, and Khadidja***; thank you for always being there for me. Your love, understanding, and motivation have been a source of strength for me. I am blessed to have you as my sisters.

I would also like to dedicate this thesis to my friends, ***Amani Boussaa, Fatma Zohra Ouamane, Selma Bedrani, Leila Lalia Mizat, Aya Mayouf, and Yasmine Aboulkacem***; thank you for your love, support, and friendship. Your presence in my life has brought joy, laughter, and inspiration. I am grateful for having you in my life.

Lastly, I dedicate this thesis to my grandmother, my aunts, uncles, cousins, and all my family members, who have been a constant source of love and encouragement. Your support and presence in my life have been invaluable, and I am forever grateful for your unlimited support.

Baya Lina

Abstract

The rapid digitalization of artwork collections in libraries, museums, galleries, and art centers has resulted in a growing interest in developing autonomous systems capable of understanding art concepts and categorizing fine art paintings as it became difficult to manually manipulate the content of these collections. However, the task of automatic categorization comes with significant challenges due to the subjective interpretation and perception of art elements and the reliance on accurate annotations provided by art experts. As in recent years, deep learning approaches and computer vision techniques have shown remarkable performance in automating painting classification; this research aims to develop efficient deep learning systems that can automatically classify the artistic style of fine-art paintings. In this thesis, we investigate the effectiveness of seven pre-trained EfficientNet models for identifying the style of a painting and propose custom models based on pre-trained EfficientNet architectures. In addition, we analyzed the impact of deep retraining the last eight layers on the performance of the custom models. The experimental results on the standard fine art painting classification dataset, Painting-91 indicate that deep retraining of the last eight layers of the custom models yields the best performance, achieving a 5% improvement compared to the base models. This demonstrates the effectiveness of leveraging pre-trained EfficientNet models for automatic artistic style identification in paintings. Moreover, the study presents a framework that compares the performance of six pre-trained convolutional neural networks (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3) for identifying artistic styles in paintings. Notably, Xception architecture is employed for this purpose for the first time. Furthermore, the impact of different optimizers (SGD, RMSprop, and Adam) and two learning rates (1e-2 and 1e-4) on model performance is studied using transfer learning. The experiments on two different art classification datasets, Pandora18k and Painting-91 revealed that InceptionResNetV2 achieves the highest accuracy for style classification on both datasets when trained with the Adam optimizer and a learning rate of 1e-4. Integrating deep learning algorithms and transfer learning techniques into fine art painting analysis and classification offers promising avenues for automating style identification tasks. The proposed models and findings contribute to the development of automatic methods that enable the art community to efficiently analyze and categorize the vast number of digital paintings available on the internet.

Keywords: Computer vision, Image processing, Convolutional neural network, Style Classification, Optimizers, Transfer learning.

Résumé

La numérisation des collections d'œuvres d'art dans les bibliothèques, les musées, les galeries et les centres d'art a suscité un intérêt croissant pour le développement de systèmes autonomes capables de comprendre les concepts artistiques et de catégoriser les peintures d'art avec précision, car il est devenu difficile de manipuler manuellement le contenu de ces collections. Étant donné que les approches d'apprentissage profond et les techniques de vision par ordinateur ont montré des performances remarquables dans l'automatisation de la classification des peintures ces dernières années, cette recherche vise à développer des systèmes d'apprentissage profond efficaces capables de classer automatiquement le style artistique des peintures d'art. Dans cette thèse, nous étudions l'efficacité de sept modèles EfficientNet pré-entraînés pour identifier le style d'une peinture, et nous proposons des modèles personnalisés basés sur les architectures EfficientNet pré-entraînées. De plus, nous analysons l'impact de la réentrainement profonde des huit dernières couches sur les performances des modèles personnalisés. Les résultats expérimentaux sur la base de données Painting-91, indiquent que la réentrainement profonde des huit dernières couches des modèles personnalisés offre les meilleures performances, avec une amélioration de 5 % par rapport aux modèles de base. Cela démontre l'efficacité de l'utilisation des modèles EfficientNet pré-entraînés pour l'identification automatique du style artistique dans les peintures. De plus, cette étude présente une comparaison des performances de six réseaux neuronaux convolutionnels pré-entraînés (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121 et EfficientNet B3) pour l'identification des styles artistiques dans les peintures. Notamment, l'architecture Xception est utilisée pour la première fois pour ce but. Par ailleurs, l'impact de différents optimiseurs (SGD, RMSprop et Adam) et de deux taux d'apprentissage (1e-2 et 1e-4) sur les performances du modèle est étudié à l'aide de l'apprentissage par transfert. Les expériences menées sur deux bases de données différentes, Pandora18k et Painting-91, révèlent qu'InceptionResNetV2 obtient la meilleure précision pour la classification des styles sur les deux ensembles de données lorsqu'il est entraîné avec l'optimiseur Adam et un taux d'apprentissage de 1e-4. L'intégration des algorithmes d'apprentissage profond et des techniques d'apprentissage par transfert dans l'analyse et la classification des peintures d'art offre des perspectives prometteuses pour l'automatisation des tâches d'identification des styles. Les modèles proposés et les résultats contribuent au développement de méthodes automatiques permettant à la communauté artistique d'analyser et de catégoriser efficacement le grand nombre de peintures numériques disponibles sur Internet.

الملخص

تطورت عملية رقمنة التشكيلات الفنية في المكتبات، المتاحف، المعارض ومراكز الفن، مما أثار اهتمامًا متزايدًا في تطوير أنظمة ألية قادرة على فهم الفن وتصنيف اللوحات الفنية، نظرًا لصعوبة التصنيف اليدوي لمحتوى هذه التشكيلات. ومع ذلك، فإن مهمة التصنيف الآلي تواجه تحديات كبيرة بسبب التفسير الذاتي وإدراك عناصر الفن، واعتماد التعليقات الدقيقة التي يقدمها خبراء الفن. للتغلب على هذه التحديات، يتطلب تطوير خوارزميات وتقنيات متقدمة تستطيع التقاط التفاصيل المعقدة والتعقيدات في الأنماط الفنية بفعالية، مما يسمح بتحليل وتصنيف تلقائي أكثر دقة وموثوقية للوحات الفنية الجميلة.

في السنوات الأخيرة، أظهرت مناهج التعلم العميق وتقنيات رؤية الحاسوب أداءً ملحوظًا في التصنيف التلقائي للوحات الفنية. تهدف هذه الدراسة إلى تطوير أنظمة تعلم عميق فعالة تستطيع تصنيف أسلوب الفن في اللوحات تلقائيًا. في هذه الأطروحة، نقوم بدراسة فعالية سبعة نماذج EfficientNet مُدرَّبة مسبقًا في تحديد نمط اللوحة، ونقترح نماذج مخصصة مبنية على تلك الهياكل المُدرَّبة مسبقًا. بالإضافة إلى ذلك، نحلل تأثير إعادة تدريب آخر ثماني طبقات في النماذج المخصصة على أدائها. تشير نتائج التجارب على مجموعة بيانات اللوحات الفنية (Painting-91)إلى أن إعادة تدريب آخر ثماني طبقات في النماذج المخصصة تؤدي إلى أفضل أداء، مع تحقيق تحسين بنسبة 5% مقارنة بالنماذج الأساسية. وهذا يُظهِر فعالية استخدام نماذج EfficientNetالمُدَرَّبة مسبقًا في تحديد النمط الفني للوحات تلقائيًا.

علاوة على ذلك، تقدم الدراسة مقارنة أداء ستة شبكات عصبية حوسبية مُدَرَّبة مسبقًا(Xception ، ResNet50، InceptionV3، InceptionResNetV2، DenseNet121، و EfficientNet B3)في تحديد الأنماط الفنية في اللوحات. يستخدم تحديدًا Xceptionلأول مرة لهذا الغرض. وعلاوة على ذلك، يُدرس تأثير مُحسِّنات مختلفة مثل(SGD ، RMSprop، و Adam)ومعدلات تعلُّم مختلفة على أداء كل نموذج باستخدام التعلم النقلي. تُظهِر التجارب على مجموعتي بيانات مختلفتين، Pandora18kوPainting-91، أن نموذج InceptionResNetV2 يحقق أعلى دقة في تصنيف الأنماط على كلا المجموعتين عند تدريبه بمحسِّن Adam ومعدل تعلُّم (1e-4).

تكامل خوارزميات التعلم العميق وتقنيات التعلم النقلي في تحليل وتصنيف اللوحات الفنية الجميلة يوفر آفاقًا واعدة لتطبيقات التعرف على الأنماط تلقائيًا. تسهم النماذج المقترحة والنتائج المستندة في تطوير الأساليب التلقائية التي تمكِّن المجتمع الفني من تحليل وتصنيف العديد الهائل من اللوحات الرقمية المتوفرة على الإنترنت.

Contents

List of Figures

List of Tables

List of Abbreviations

Adam Adaptive Moment Estimation

AR Augmented Reality

CSIFT Color Scale Invariant Feature Transform

CNN Convolutional Neural Network

DCEC Deep Convolutional Embedding Clustering

HOG Histogram of Oriented Gradients

ILSVRC ImageNet Large-Scale Visual Recognition Challenge

GAN Generative Adversarial Network

GIST Global Image Structure

GLCM Gray-Level Co-occurrence Matrix

GNN Graph Neural Network

KNN K-Nearest Neighbor

LBP Local Binary Patterns

LSTM Long-Short Term Memory

MBConv Mobile Inverted Bottleneck Convolution

MTL Multi-Task Learning

O-SIFT Opponent-SIFT

PHOG Pyramid of Histograms of Orientation Gradients

ROC Receiver Operating Characteristic

ReLU Rectified Linear Unit

RMSprop Root Mean Square propagation

RNN Recurrent Neural Network

SE Squeeze and Excitation

SIFT Scale-Invariant Feature Transform

SGD Stochastic Gradient Descent

SOM Self-Organizing Map

SVM Support Vector Machine

T-SNE T-distributed Stochastic Neighbor Embedding

UFLK Unsupervised Feature Learning with the K-means

WNN Weighted Nearest Neighbor

List of Publications

International journal paper

- B. L. MENAI and M. C. BABAHENINI, “The Effect of Optimizers on CNN Architectures for Art Style Classification,” International Journal of Computing and Digital Systems, vol. 13, no. 1, pp. 353–360, 2023.

International conference paper

- B. L. MENAI and M. C. BABAHENINI, “Recognizing the Style of a Fine-Art Painting with EfficientNet and Transfer Learning,” in 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), pp. 1–6, IEEE, 2022.

Workshop

- B. L. MENAI and M. C. BABAHENINI, ”Comparative Analysis of Pre-trained Convolutional Neural Networks and Optimizers for Artistic Style Classification”, in 1st workshop on Advances in Deep Learning for Images and Immersive Technologies (ADL2IT), 10-12 October 2023

Chapter 1

Introduction

1.1 Problem Statement

Fine art paintings play a significant role in society, culture, and history as they hold a part of the cultural heritage and can provide a window into the past, reflecting the values, beliefs, and customs of different societies and time periods.

Art specialists, art historians, or curators usually study and classify fine art paintings into several groups depending on artistic style, genre, historical time, or artist to facilitate their understanding, comparison, and manipulation.

Art experts use various techniques to recognize the artistic style of fine art paintings; it includes analyzing the painting's composition, color palette, brushwork, and other visual elements to identify its style. Different styles have distinct visual characteristics that are recognizable to experts trained in art history. This process is very time-consuming and extremely expensive as it is performed manually.

However, this process can also be automated using machine learning techniques as computer vision has advanced significantly in recent years, with deep learning techniques providing the ability to recognize patterns and features in previously difficult images to detect. Automatically classifying paintings by style and authorship appears to be among the most challenging computer vision issues. It has recently attracted a growing amount of attention from the academic community because of the rise of databases that contain high-resolution digital copies of fine art paintings.

The organization of digitized painting collections and the retrieval of artworks from these collections become increasingly challenging as the size of these collections grows. It is required to organize the painting databases into classes and sub-classes in order to achieve efficient management of search and other activities of a similar kind. To manually tag each of these databases as they continue to expand in size would be both extremely expensive and time-consuming.

As a result, of the difficulty of the problem stated above, many studies have been conducted in the areas of painting analysis, artistic style and genre categorization, artist identification, and automatic annotation of paintings with these tags.

This thesis aims to contribute to the field of art by exploring the use of deep learning algorithms specifically for recognizing the artistic style of fine art paintings. By leveraging the power of deep learning, we seek to develop a model that can accurately identify and classify paintings based on their unique artistic styles.

The developed model for recognizing the artistic style of a fine art painting could be integrated into an augmented-reality application, allowing users to point their smartphone camera at a painting and receive information about its style, historical context, and the artist who created it. By bringing together the worlds of art and technology, people can engage with fine art paintings and deepen their understanding and appreciation of them.

1.2 Research Contributions

The main contributions of this research are:

- Providing a comprehensive literature review of some of the most relevant machine learning studies in classifying and analysing fine art paintings by style.
- Investigating the effectiveness of the pre-trained EfficientNet models family from B0 to B6 for the task of identifying the style of a painting.
- Proposing custom models based on the pre-trained EfficientNet models and using transfer learning to fine-tune the models.
- Analyzing the effect of deep retraining the last layers of our custom models.
- Proposing a framework to compare the performances of six pre-trained convolutional neural networks (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3) for identifying the artistic style of a painting.
- Studying the effect of various optimizers (SGD, RMSprop, and Adam) with different learning rates (1e-2 and 1e-4) on the performance of pre-trained models.

1.3 Thesis Structure

The thesis is structured as follows:

- Chapter 1: describes the research motivation and the research problem. The thesis contributions and the thesis structure follow it.
- Chapter 2: provides an introduction to fine-art paintings, offering an overview of the significance of artistic style and its role in the world of art. It also

explores the methods art experts employ to identify and distinguish the unique style of a painting, highlighting the challenges and difficulties they encounter in this process. Furthermore, this chapter explores augmented reality (AR) integration in the art world, discussing how AR technology can enhance the viewing experience and provide new avenues for artistic expression.

- Chapter 3: provides an overview of the existing research and methodologies of the literature on fine-art painting style classification, covering both classical and deep learning approaches. It discusses the commonly used datasets for painting classification, highlighting their characteristics and challenges. Additionally, the chapter presents the evaluation metrics in image classification, explaining their relevance in assessing model performance.

- Chapter 4: focuses on our approach to recognizing the artistic style of fine art paintings using EfficientNet models and transfer learning. We describe in detail the methodology, outline the steps taken in preprocessing the data, and explain how transfer learning with EfficientNet models is applied to the task of style recognition. The experiments conducted throughout this chapter are specifically based on the art classification dataset Painting-91.

- Chapter 5: investigates the effect of optimizers on CNN architectures for art style classification. The focus is on exploring the impact of various optimization algorithms, including SGD, RMSprop, and Adam, with different learning rates (1e-2 and 1e-4), on the performance of six pre-trained CNN models in the task of recognizing and classifying the artistic style of paintings. The experiments are conducted on two datasets, Pandora18k and Painting-91, to thoroughly evaluate the accuracy and effectiveness of each optimizer when combined with different pre-trained CNN architectures.

- Chapter 6: concludes the thesis and presents some future work research direction.

Chapter 2

Fine-art Paintings: A Comprehensive Exploration of Artistic Styles and Challenges in Painting

2.1 Introduction

This chapter provides an introduction to fine-art paintings, including an overview of the artistic style and an exploration of the methods used by experts to identify the unique style of a painting, alongside the challenges they face in this process. Furthermore, it explores the difficulties researchers encounter in working with fine art paintings and the integration of augmented reality in the art world. Finally, we conclude the chapter with a conclusion.

2.2 Preview

2.2.1 Fine art paintings

Fine art paintings encompass a diverse range of visual artworks created on various surfaces with the intention of aesthetic expression, utilizing different styles, subject matters, mediums, painting techniques, and materials. They are a form of artistic expression that prioritizes creativity, aesthetics, and the exploration of visual language.

The world of fine art painting encompasses a vast range of styles, genres, and subjects. Artists employ various techniques, such as brushwork, layering, blending, and texture creation, to bring their artistic visions to life. Through the use of color, composition, light, shadow, and other visual elements, artists convey emotions, tell stories, explore symbolism, and challenge societal norms.

Fine-art paintings are treasured for their ability to evoke emotions, inspire contemplation, and provide a unique perspective on the world. They serve as a means of artistic expression and cultural heritage, exhibited in galleries, museums, and private collections worldwide. These artworks continue to captivate and inspire viewers with their beauty, creativity, and the profound depths of human imagination.

2.2.2 Artistic style

The artistic style of a fine art painting can vary widely depending on the artist and the specific painting. Fine art encompasses a broad range of styles and techniques, each with its own distinct aesthetic qualities. In visual art, style is defined as "...a distinctive manner which permits the grouping of works into related categories" (1).

In other words, the artistic style refers to the distinctive manner in which an artist portrays subjects, conveys ideas, and applies techniques. It encompasses the artist's personal interpretation, preferences, and recurring elements that define their visual vocabulary. Artistic style can vary widely, from realistic and representational to abstract and experimental. It may be influenced by historical movements, cultural contexts, and personal experiences, reflecting the artist's worldview and intentions.
Figure 2.1 (2) presents the timeline of artistic styles as different styles appeared over history in different time periods. Examples of these styles include Abstract Expressionism, Romanticism, Baroque, Rococo, Realism, and Symbolism.

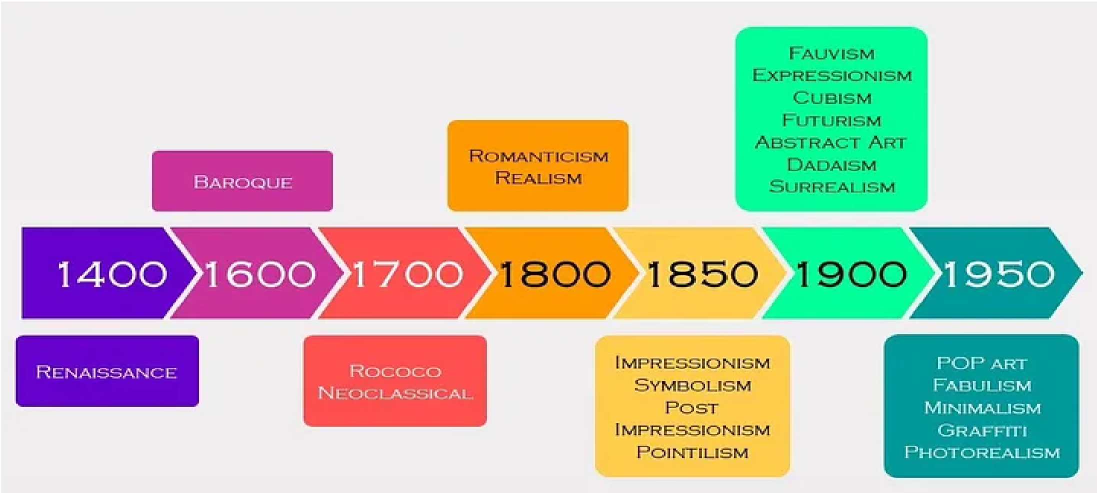


Figure 2.1: Timeline of artistic styles

The Baroque style (1600-1750) emerged in the early 16th century in Rome, then spread rapidly to the rest of Europe. It is renowned for its dramatic expression, ornate detailing, and use of dark colors, contrasting light and shadow to create a sense of depth and movement. Figure 2.2 shows different examples of fine art paintings with the Baroque style.

The Realism style (1848-1900) came during the French revolution. It is characterized by a highly detailed and precise representation of reality, often with an emphasis on natural light and a neutral color palette. Realist painters sought to portray the world as it was, focusing on everyday life and ordinary people. Figure 2.3 presents different examples of fine art paintings with the Realism style

Figure 2.2: Examples of paintings with the Baroque style.

Figure 2.3: Examples of paintings with the Realism style

2.2.3 The role of artistic style and techniques

Artistic styles and techniques are not fixed entities but evolve over time. Artists may start with a particular style and then explore new avenues, experimenting with techniques and pushing the boundaries of their artistic practice. Artistic evolution can be influenced by personal growth, exposure to different artistic movements, or societal and cultural shifts, resulting in the development of distinct individual styles or even the establishment of new artistic movements.

Artistic style and techniques play a pivotal role in shaping fine art paintings's visual language and expressive qualities. They contribute to the artwork's unique identity and aesthetic character, reflecting the artist's creative choices, cultural influences, and individual artistic vision. Understanding the significance and impact of artistic style and techniques is essential for appreciating and analyzing fine art paintings.

2.3 Style classification by art experts

In this section, we describe the methods used by experts to recognize the style of a painting and the challenges they face in this process.

2.3.1 Methods

Art experts recognize the artistic style of a painting through careful observation and analysis of its various elements, such as composition, colors, brushstrokes, and subject matter. They use their knowledge of art history, as well as their experience and expertise, to identify the unique characteristics of different styles. Some of the key factors that art experts consider when analyzing a painting's style include:

- Historical Context: This involves examining the time period in which the painting was created and the artistic styles that were prevalent during that time. For example, a painting created in the Early renaissance period may have distinct characteristics associated with that time period, such as the use of linear perspective and an interest in classical mythology(3).
- Technique: This includes studying the brushwork, the use of color, the handling of light and shade, and other technical aspects of the painting to identify any particular style or technique used by the artist. For instance, an impressionist

painting is characterized by loose brushstrokes, the use of complementary colors, and the depiction of fleeting moments of light(4).

- Subject Matter: The expert analyzes the subject matter and how it is represented in the painting, as a subject matter may be associated with particular artistic styles.
- Color Palette: The expert looks at the colors used in the painting and how they are used, as certain color schemes can be associated with different styles.
- Provenance: The expert considers the painting's history and any known information about its creation and previous ownership to identify any connections to particular artists or artistic movements.

In addition to these characteristics, an artist's style may be influenced by their cultural background, social context, and personal experiences. By carefully examining these features, art experts can identify the unique qualities that define an artist's style, connect their work to a specific artistic movement or historical period style, and provide insights into the artist's intentions and message.

By recognizing the significance of artistic style and techniques, viewers and art scholars can analyze, appreciate, and contextualize fine art paintings within the broader spectrum of artistic expression. Understanding the interplay between style, techniques, and artistic intention enriches the interpretation and appreciation of artworks, allowing for a deeper engagement with the artist's creative process and the cultural significance of the artwork.

2.3.2 Challenges and difficulties

Identifying and categorizing the unique style of a painting poses significant challenges for experts in the field of art. These challenges arise from the subjective nature of the artistic style, variations and hybridity within styles, contextual considerations, limitations of visual analysis, incomplete or scarce information, and the complexities of evolving contemporary styles.

The subjective and interpretive nature of artistic style makes achieving consensus among experts difficult. In contrast, the diverse range of styles and the incorporation of multiple influences by artists further complicate the task. Different experts may have varying opinions and criteria for categorizing styles, leading to a lack of a general agreement. The fluidity and evolution of artistic styles further complicate the task, as artists often experiment with new techniques and borrow influences from multiple sources, making it challenging to assign rigid classifications.

Contextual factors such as historical, social, and cultural contexts must be considered to assign artworks to specific stylistic categories accurately. Additionally, limitations in visual analysis, incomplete information about artists and art movements, and the evolving nature of contemporary art styles present additional hurdles for experts.

According to Tan et al. (5), the classification of paintings is more challenging than normal classification tasks, such as the recognition of scenes, object, and architecture in natural images. This is despite the fact that the classification of paintings has a variety of applications, such as art authentication, art collection management, digital archiving and art recommendation systems.

Figure 2.4 (6) presents a collection of representative examples illustrating the challenge of classifying paintings by different artists. Each row in the figure contains a distinctive array of samples from different well-known artists, including Vincent van Gogh, Paul Cezanne, and Maurice Prendergast. The diverse colors displayed beneath the pictures represent the varying genres and styles employed by these artists in each painting. Notably, a single artist may create a variety of artworks in multiple different styles and genres, making it challenging for a computer to categorize the paintings accurately.

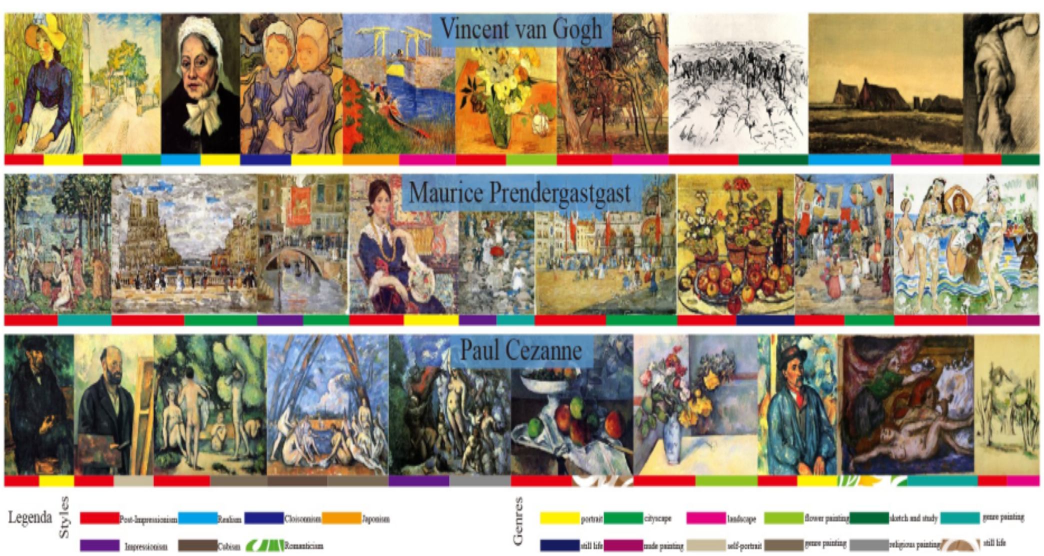


Figure 2.4: Samples of paintings from different artists. Styles and genres are included based on the color coding

One of the most significant obstacles in the study of artworks is the ability of machines to comprehend abstraction and visualization in painting. However, classifying paintings by style requires a combination of technical expertise and attention to detail because it is confusing to distinguish between classes. For example, Figure 2.5 (7) shows a variety of images of buildings in various artistic styles, and Figure 2.6 (8) displays eight examples of forest scenes painted in various fine art styles. Although all of these images have the same content (building or forest), they belong to various stylistic categories. Thus, fine-art painting image categorization is more challenging than natural image classification.

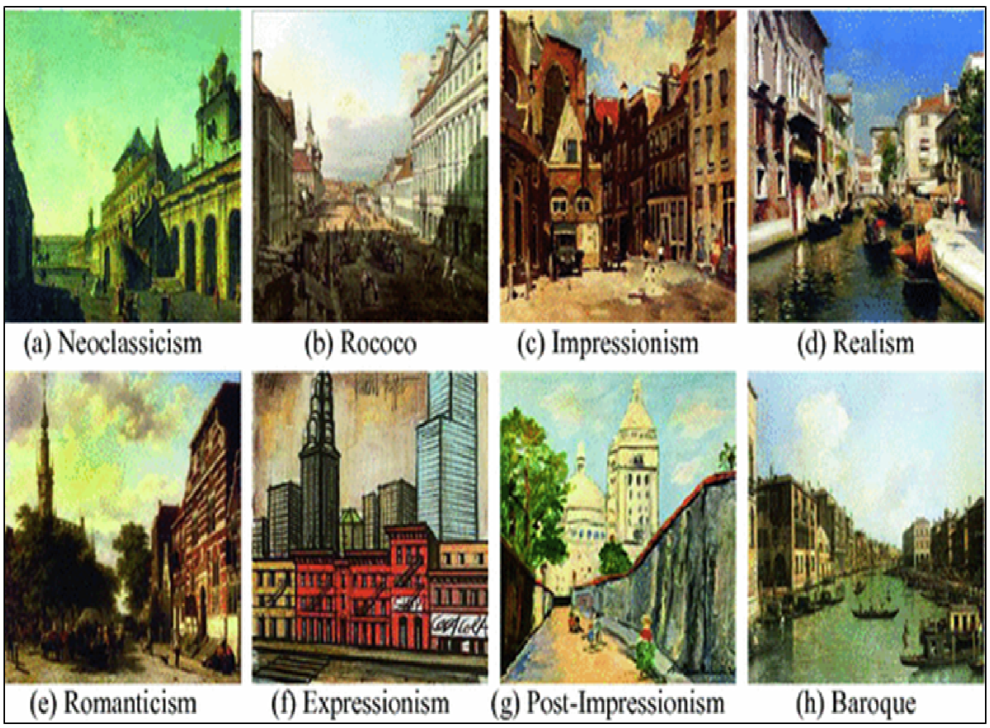


Figure 2.5: Portraying buildings with different styles.

Working with digital images of paintings can introduce additional challenges, mainly when accurately representing the artwork's colors, texture, and overall appearance. The conversion from physical to digital form may involve issues such as color accuracy, lighting conditions, and capturing fine details. Furthermore, the size of the paintings can pose difficulties in capturing the entire piece in a single image without compromising on detail or resolution. Large paintings may require high-resolution imaging techniques or stitching multiple images together to ensure that all aspects of the artwork are properly captured.

Figure 2.6: Images of forest with different styles.

2.4 Fine art paintings and augmented reality

In this section, we describe augmented reality and some of its applications. Furthermore, we present the integration of augmented reality in the art world.

2.4.1 Augmented reality

Augmented Reality (AR) is a technology that enhances the real-world environment by overlaying computer-generated perceptual information onto it, thereby augmenting the user's sensory experience (9). AR integrates virtual elements, such as images, videos, 3D models, or data, into the user's real-world view in real-time, allowing for an interactive and immersive blend of the physical and digital worlds.

The key feature of AR is its ability to align the virtual content with the real-world context, allowing users to interact with and manipulate the augmented elements in real time. This interaction can involve gestures, voice commands, or other input methods, enabling users to engage with and manipulate the virtual objects or access additional information overlaid on physical objects.

AR has various applications across industries, including gaming(10), education(11), healthcare(12), architecture(13), retail(14), and entertainment(15). It offers opportunities for immersive learning experiences, enhanced visualization, remote collaboration, guided navigation, virtual try-on experiences, and more.

2.4.2 Integration of augmented reality in art

The integration of augmented reality (AR) technology within the art world has emerged as a dynamic and evolving field, revolutionizing the way art is experienced, interpreted, and interacted with.

Augmented Reality (AR) technology and computer vision techniques have evolved to offer exciting possibilities to accurately recognize and track fine art paintings, enhancing artistic interpretation and deepening our engagement with artworks. By overlaying digital information and virtual elements onto the real-world view, AR enriches our understanding of artistic works, provides contextual information, and offers immersive experiences that augment the traditional art viewing process.

One way AR enhances artistic interpretation is by providing access to additional layers of information about the artwork. Viewers can use their smartphones or AR-enabled devices to scan a painting and instantly access to details about the artist, the historical context, the techniques used, or the symbolism embedded within the artwork. This additional information empowers viewers to delve deeper into the artistic intention and provides a richer context for interpretation.

The current landscape of AR in the art world showcases a growing adoption of this technology across various domains; for example, museums and galleries have embraced AR as a means to enhance traditional exhibition spaces. AR applications can provide additional layers of information, context, and multimedia content, enriching the visitor's understanding and engagement with artworks. They also can be used in art education, offering innovative ways to teach and learn about art history, techniques, and styles. Augmented reality enabled applications and platforms to provide interactive lessons, virtual studio experiences, and immersive art history tours.

Figure 2.7 (16) presents an example of AR applications that use image recognition to identify scanned artworks and provide people with additional information about them.

Figure 2.7: Exemple of augmented reality applications in an art gallery.

2.5 Conclusion

This chapter comprehensively introduced fine-art paintings, their instinctive artistic styles, and their role in art. Additionally, it presented methods employed by art experts to identify the unique style of a painting. It navigated the challenges these experts face in this task and the obstacles researchers encounter when working with digital images of fine art paintings. Furthermore, the chapter explored the integration of augmented reality (AR) within the art world, illuminating the innovative applications and transformative potential of this technology in enhancing the viewer's experience and expanding the boundaries of traditional art.

In conclusion, the need for automatic tools to recognize the style of paintings is evident in the field of fine art. Automatic tools, such as machine learning algorithms and computer vision techniques, offer promising solutions to assist experts in recognizing and analyzing artistic styles. These tools have the potential to enhance the efficiency and accuracy of style recognition, enabling a deeper understanding of artistic choices and contributing to the preservation, research, and accessibility of artworks.

The next chapter presents the literature review of artistic style classification, the most famous painting datasets and the most used evaluation metrics in image classification.

Chapter 3

Style Classification : Literature Review

3.1 Introduction

In this chapter, we present a comprehensive overview of the current research and methodologies in the field of fine-art painting style classification, encompassing both classical and deep learning approaches. Additionally, we examine the various approaches adopted in the literature and explore the most relevant painting datasets commonly utilised for painting classification, highlighting their unique characteristics and the challenges they pose. Furthermore, we describe the evaluation metrics employed in image classification, explaining their significance in effectively evaluating the performance of style classification models.

3.2 Approaches for style classification

Throughout the past several years, the subject of classifying paintings of fine art according to their styles has been investigated in a variety of studies, different approaches have been applied they can be divided into two main approaches based on their feature selection and classification algorithms. The first approach, known as the classical approaches, utilizes handcrafted feature extraction techniques in combination with machine learning methods. In contrast, the second approach utilises deep learning techniques, where the neural networks themselves automatically extract features. Within the realm of deep learning, there are two main subcategories: supervised learning approaches and unsupervised approaches. The supervised learning approaches further encompass four subcategories: convolutional neural network approaches, transfer learning approaches, multi-task learning approaches, and hierarchical approaches. Figure 3.1 visually represents these different approaches employed in style classification.

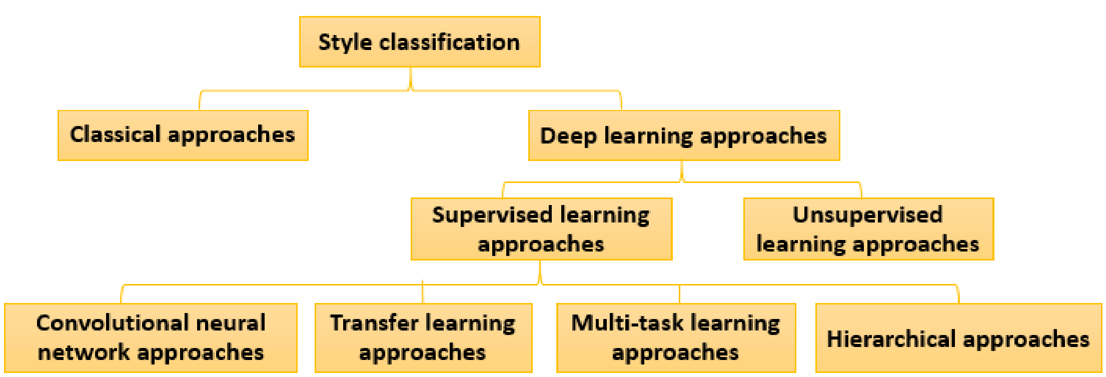


Figure 3.1: Approaches for style classification

3.2.1 Classical approaches for style classification

Given that the artistic style of a painting may be understood in terms of texture, colors, or shapes, early studies proposed to automatically identify the unique style of painting by applying the classical approaches for image classification, which involve extracting low-level and global features, as they are two important categories of visual descriptors used to extract information from images. Once the features are extracted, a machine-learning algorithm is trained to recognize the patterns in the feature space and classify new images based on those patterns. Figure 3.2 illustrates the framework of this process.

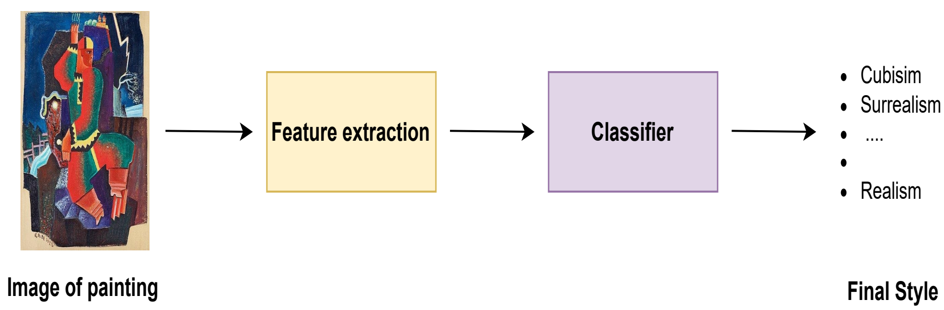


Figure 3.2: Diagram of classical approaches for style classification

Low-level features refer to the basic and primitive visual characteristics of an image, that can be extracted directly from the pixel values, such as edges, corners, texture patterns, and color histograms. These features are often computed using simple mathematical operations or filters applied to the image. Low-level features capture local information and provide a representation of the image at a pixel level. On the other hand, global features capture high-level and holistic information about the entire image. Instead of focusing on local details, global features consider the overall structure, composition, and spatial distribution of visual elements in an image. These features provide a global representation of the image and are typically computed by aggregating or summarizing the low-level features across the entire image. Table 3.1 represents some of the most commonly used low-level and global features in image classification along with their brief definitions:

Table 3.1:: Examples of low-level and global features with brief definitions

Feature Type	Example	Brief Definition
Low-level	Histogram of Oriented Gradients (HOG) (17)	Counts the occurrences of gradient orientations in localized regions of an image to capture the shape and appearance of an object.
	Scale-Invariant Feature Transform (SIFT)(18)	Detects and describes keypoint features that are invariant to scale, rotation, and affine transformations.
	Local Binary Patterns (LBP) (19)	Extracts texture information by comparing the intensity of a central pixel with its neighbours.
	Gray-Level Co-occurrence Matrix (GLCM) (20)	Measures the frequency of pixel pairs at different spatial relationships to capture textural patterns.
Global	GIST (Global Image Structure) (21)	Summarizes the distribution of simple visual features, such as color and texture, to capture the holistic structure of an image.
	Color Histograms (22)	Quantizes and counts the frequency of colors present in an image, providing a global color distribution.
	Bag-of-Words Models(23)	Represents an image as a histogram of visual words, where words are learned from a collection of local features.

One of the first studies to automatically recognize the style of a painting was presented by Shamir et al. (24) They proposed to classify 513 paintings into three distinct styles: Expressionism, Impressionism, and Surrealism by analyzing various low-level visual features of the paintings, such as color, texture, and shape; then used the machine learning approach Weighted Nearest Neighbor (WNN) as a classifier to assign paintings to their respective styles. Although the proposed approach achieved significant accuracy in style classification, it was limited to a small number of paintings and art styles. Another example of a relevant study based on the extraction of low-level features method was proposed by Arora and Elgammal (25). They extracted low-level features such as opponent-SIFT (O-SIFT) and color scale-invariant feature transform (CSIFT) from paintings. These features were then classified using a Support Vector Machine (SVM) classifier. The study demonstrated the effectiveness of these features in categorizing paintings into different art styles. However, the dataset used in the study was relatively small, containing only seven art styles with 70 paintings.

Agarwal et al.(26) conducted a study on painting classification based on genre and style, demonstrating the successful utilization of various feature extraction methods, which included color histograms, texture features, and shape features, to capture the visual characteristics of the paintings. Additionally, the authors employed the libsvm classifier with an X^2 kernel for the classification task. The results showed the effectiveness of this approach in accurately classifying paintings based on their genre and style. However, the study has some limitations, such as the dependency on the quality and quantity of input data and the limited dataset used for evaluation.

In addition to proposing the Painting-91 dataset of 4,266 painting images from 91 different artists belonging to 13 art styles, Khan et al. (27) evaluated the performance of different global and local features descriptors for the tasks of artist and style classification using an SVM classifier with an X^2 kernel. They found that using only the low-level features was insufficient and combining multiple features, including local features such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), and Pyramid of Histograms of Orientation Gradients (PHOG), and global features such as Global Image Structure (GIST) and bag-of-words framework, significantly improves the performance of the classifiers. However, the results were yet limited.

Different combinations of features and classifiers were explored by Falomir et al. (28). They proposed a method called QArt-Learn for categorizing paintings. The study combined color similarity, qualitative color descriptors, and quantitative global features, which provided a rich representation of painting characteristics with machine learning techniques. The results indicated that the combination of these features, when used with the K-Nearest Neighbor (K-NN) and Support Vector Machine (SVM) classifiers, achieved high accuracy in classifying paintings into three art styles: Post-Impressionism, Impressionism, and Baroque.

Florea et al. (29) proposed constructed a new dataset called Pandora18K, which consists of 18,628 high-resolution images of paintings labeled into 18 different artistic movements. In addition, they proposed a novel approach for recognizing different artistic movements in visual art using a combination of color structure and topographic description features to capture diverse aspects of artistic movements. The proposed approach utilized boosted ensembles of SVMs, which combined the two features to improve recognition accuracy. As their results were limited, later on, in (30), they proposed to improve their results by introducing an expert committee with soft voting. The final decision was made based on a majority vote of the classification results obtained by evaluating the entire painting and various randomly selected subregions. However, in all cases, no significant improvements were achieved, and the first breakthrough for style classification was due to the application of deep learning techniques.

3.2.2 Deep learning approaches for style classification

Deep learning approaches have gained significant popularity in the field of style classification for fine-art paintings and made the first breakthrough. These approaches benefit from the power of artificial neural networks to automatically extract features from images and learn complex patterns associated with different artistic styles. The following sections briefly overview the several standard deep learning models and techniques used in style classification. These studies are organized based on the learning approach employed, supervised or unsupervised. Each section reviews the relevant studies within the respective classification framework.

3.2.2.1 Supervised learning approaches for style classification

In the field of image classification, supervised learning plays a crucial role in establishing the relationship between images and their corresponding labels or categories. This approach allows the machine to learn from labeled examples and make predictions on unseen data. During training, a supervised learning algorithm utilizes a known dataset with labeled inputs and outputs to train a model that can generate accurate predictions for new data instances.

The training process involves minimizing the error between the predicted outputs and the established responses in the training dataset. By iteratively adjusting its internal parameters, the algorithm strives to improve its prediction accuracy. The quality and diversity of the training data significantly impact the performance and generalization ability of the trained model. A well-curated and diverse training dataset enables the model to learn meaningful patterns and make reliable predictions on unseen data.

In the specific domain of fine-art style categorization, supervised learning approaches have been widely adopted. These approaches involve training machine learning models using labeled data, where each instance is associated with a particular artistic style.

Researchers have explored various techniques within supervised learning, focusing on convolutional neural networks (CNNs) as the primary architecture. Transfer learning has been commonly employed, where pre-trained CNN models are fine-tuned on fine-art datasets to leverage their learned representations from large-scale image datasets.

Furthermore, researchers have utilized multi-task learning, where the model is trained not only for style classification but also for related tasks such as artist identification or genre classification. This approach benefits from shared representations across tasks, leading to improved performance. Hierarchical learning has also been explored, allowing models to recognize styles at different levels of granularity, capturing both high-level style characteristics and subtle variations within specific sub-styles or artists. Figure 3.3 illustrates a summary of these supervised approaches for style classification. The studies in the literature of each approach are presented in the following sections.

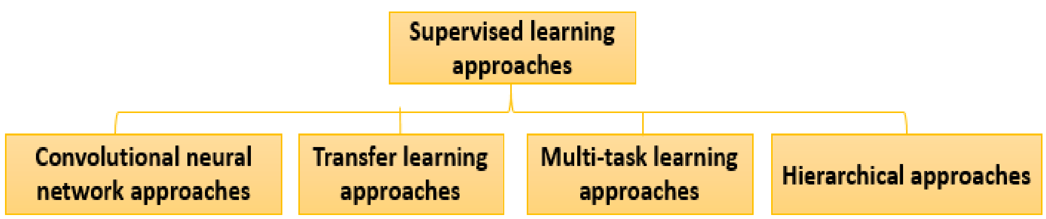


Figure 3.3: Diagram of supervised approaches for style classification

3.2.2.1.1 Convolutional Neural Networks(CNN) The recent advancements in deep convolutional neural networks (CNNs) have revolutionized the field of image recognition and classification. These networks have demonstrated remarkable capabilities in automatically learning and extracting relevant features directly from raw image data, without the need for explicit feature engineering.

This progress has motivated researchers to explore the application of CNNs for style classification tasks. By leveraging the powerful learning abilities of CNNs, researchers aim to develop models that can accurately recognize and classify artistic styles based on the visual characteristics of artworks. Figure 3.4 presents the basic architecture of CNN, which is based on two main parts: feature extraction and classification.

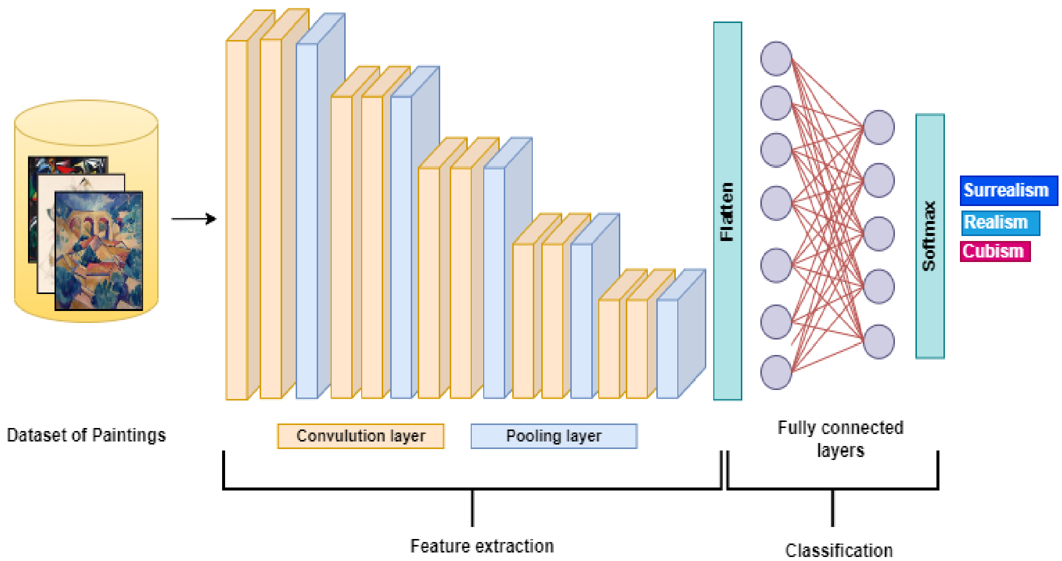


Figure 3.4: Diagram of convolutional neural networks for style classification

The first part consists of convolutional layers, which apply a set of learnable filters to the input image. Each filter produces a feature map, which highlights specific patterns or features present in the input image. The filters are learned through a process of backpropagation, where the network adjusts the filter weights to minimize the error between the predicted output and the true output. In addition to convolutional layers, it also includes pooling layers, which reduce the spatial size of the feature maps by taking the maximum or average value of a local region. This helps to make the network more computationally efficient and makes the learned features more robust to small shifts and distortions in the input image.

The second part has one or more fully connected layers at the end of the network, which are used to make predictions based on the learned features. These layers take the high-level features extracted by the convolutional and pooling layers and map them to a fixed number of output classes. The last layer has Softmax as an activation function that is used to convert the outputs of the last layer of the network into a probability distribution over the possible classes. The class with the highest probability defines the label of the input image.

Over the years, several convolutional neural network (CNN) architectures have been proposed. Each architecture has its own unique characteristics, including its resolution, number of layers, and number of parameters. The ideal CNN model for image classification would indeed possess high accuracy, a low computation cost, and a short inference time. However, in reality, achieving the perfect balance between these attributes is often challenging, and sacrifices must be considered based on the specific requirements of the application. Table 3.2 presents the most popular CNN architectures.

Table 3.2:: Summary of CNN Architectures

Model	**Year**	**Input Image Size**	**Depth**	**Size (MB)**	**Parameters (Million)**
AlexNet (31)	2012	224x224	8	233	60M
ZFNet (32)	2013	224x224	8	192	59M
VGG16 (33)	2014	224x224	16	528	138M
VGG19 (33)	2014	224x224	19	549	144M
GoogLeNet (34)	2015	224x224	22	27	7M
Inception-V3 (35)	2016	299x299	48	89	23.9M
ResNet-34 (36)	2016	224x224	34	46	21M
ResNet-50 (36)	2015	224x224	50	96	25.6M
SqueezeNet (37)	2016	224x224	18	5	1.2M
Xception (38)	2017	299x299	81	88	22.9M
InceptionResNetV2 (39)	2017	299x299	164	213.41	56M
DenseNet-121 (40)	2017	224x224	121	33	7.6M
DenseNet201 (40)	2017	224x224	201	80	20M
MobileNet (41)	2017	224x224	28	17	4M
EfficientNet-B0 (42)	2019	224x224	240	41	5.3M

In recent years, there has been a growing interest in using CNN architectures for the task of image classification in the domain of paintings. For example, Karayev et al. (43) used the CNN architecture AlexNet to extract features from images and then compared these features with various sets of low-level features for style classification. Features were fed into the SVM classifier to predict the style of the image. They demonstrated that CNNs are more effective than classical models for classifying artworks. In addition, they introduced the first painting dataset Wikiart with 85K paintings annotated with 25 style labels. Bar et al. (44) also utilized CNN models as feature extractors to classify painting styles. They trained a deep neural network on a large dataset of paintings and extracted features from the network's layers. They then binarized the features to reduce their dimensionality and used them to train the k-nearest Neighbors (kNN) classifier for distinguishing between different artistic styles. While the method successfully captured the nuanced characteristics of artistic styles, it encountered challenges in accurately distinguishing closely related styles. As a result, the approach achieved a relatively low accuracy in the classification task.

Inspired by the observation that features maps within CNNs can effectively describe image texture, Peng and Chen (45) introduced a cascading approach of modified AlexNet that allowed the extraction of features from multiple layers, which are then combined to form the cross-layer CNN features. They showed that the cross-layer CNN features outperform traditional CNN architectures in classifying artistic style, architectural style, and artist. The incorporation of correlations between feature maps from different layers enhanced the ability to capture texture-related information, leading to improved classification accuracy. Further, Chu et al. (46) transformed the correlations between feature maps into style vectors using a Gram matrix based on a VGG19 model. These style vectors are then utilized as inputs for an SVM classifier to classify images based on their artistic styles. The proposed method demonstrated the efficacy of leveraging deep correlation features for image-style classification tasks.

3.2.2.1.2 Transfer learning Many studies on style classification that utilize deep learning techniques have employed the transfer learning technique, which involves using a pre-trained convolutional neural network (CNN) model initially trained on a comprehensive dataset of natural images and then, a brief fine-tuning phase takes place, where the model is trained on a relatively smaller image dataset representing distinct artistic categories.

Figure 3.5 provides an illustrative representation of the transfer learning process employed from natural image classification to style classification of paintings. The pre-trained CNN model is already trained on the large-scale natural image classification task ImageNet dataset (47). It consists of several layers, including convolutional layers for feature extraction and fully connected layers for classification. These layers have learned to extract general features from natural images.

For style classification of paintings, the last three layers of the pre-trained CNN model are customized by replacing the last fully connected layer, the SoftMax layer, and the classification output layer. The new last fully connected layer is adapted to match the number of different artistic styles in the dataset. The SoftMax layer produces a vector of probabilities for each artistic style, indicating the likelihood of an image belonging to each style. Finally, the classification output Llayer assigns the image to a specific stylistic category based on the highest probability. The feature extraction block of the pre-trained CNN model remains unchanged, as it has learned to extract relevant visual features from images. This block captures the inherent characteristics and patterns that can be useful for style classification.

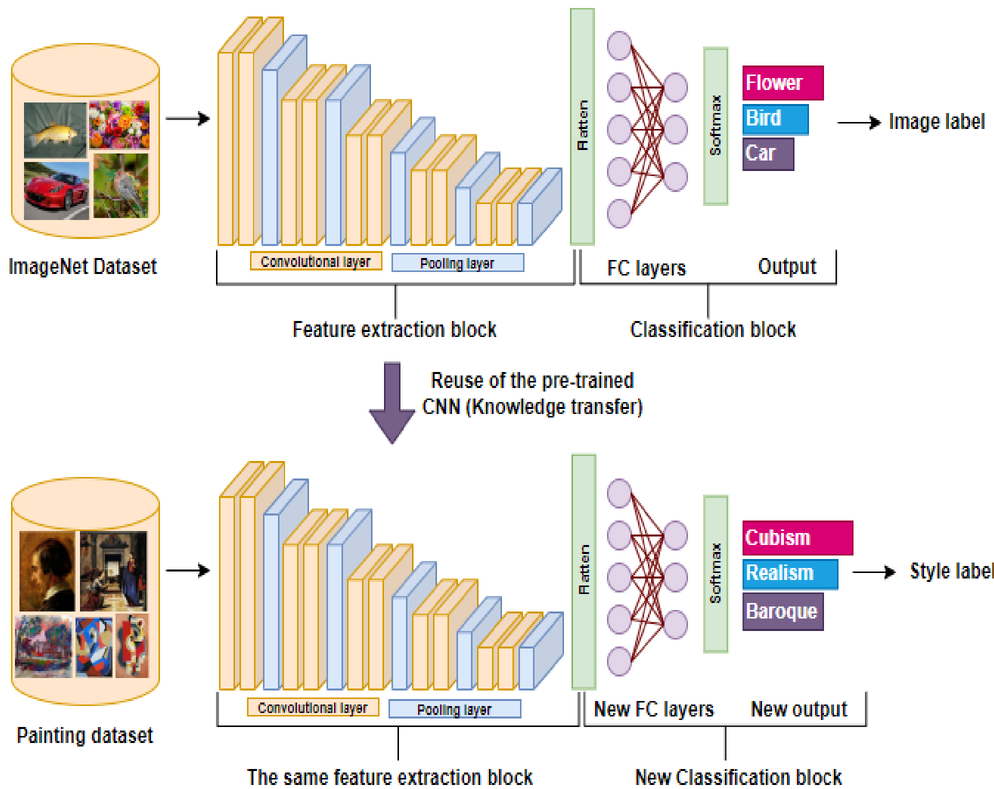


Figure 3.5: Transfer learning from natural images classification to style classification of paintings

By fine-tuning this modified network, the pre-trained CNN model can be retrained specifically for the style classification task using the initial weight values obtained from the pre-training phase. This process enables the model to leverage the knowledge learned from natural image classification and adapt it to the task of style classification in paintings.

Tan et al. (5) is one of the first studies to investigate the effectiveness of pre-trained convolutional neural networks (CNNs) as feature extractors for fine art painting classification. The authors investigated whether pre-trained CNNs could outperform handcrafted descriptors in classifying paintings based on style, genre, and artist. They utilized the Wikiart dataset and trained an end-to-end CNN model. They compared different options for their network, including fine-tuning a pre-trained CNN model that was originally trained on the ImageNet dataset for object recognition, training a CNN model from scratch, and testing support vector machine (SVM) classifiers on deep features extracted from the CNN models. Interestingly, the results of their experiments demonstrated that the fine-tuned model, which involved retraining a pre-trained CNN on the Wikiart dataset, achieved the best performance across all tasks.

Lecoutre et al. (48) employed a pre-trained deep residual neural network (ResNet) that was initially trained on the ImageNet dataset and fine-tuned on the WikiArt dataset to distinguish between 25 different artistic styles. The key finding of their study was that deeper retraining of the network contributed to improving the accuracy of artistic style classification. However, the study also suggested that using larger datasets for training deep networks could potentially lead to further improvements in accuracy. Kedia (49) focused on analyzing the performance of fine-tuning the VGG network with weighted cross-entropy for the task of large-scale style, genre, and artist classification. Yu et al. (50) explored the effectiveness of hybrid VGG and inception architectures for classifying paintings based on their respective artistic styles. The utilization of transfer learning to fine-tune the InceptionV3 architecture yielded the best results in terms of accuracy.

Elgammal et al. (51) also explored the application of pre-trained convolutional neural network (CNN) models for style classification in art. The researchers investigated how these models' learned representations correlate with the chronology of paintings, using concepts derived from art history. Their study involved training several CNN models on a dataset of digitized artworks spanning various historical periods. These models learned to extract visual features from the paintings, then examined how the learned representations captured the characteristics associated with different art styles.

Bianconi et al. (52) compared handcrafted descriptors with pre-trained convolutional networks for painting categorization. They considered nine handcrafted descriptors and applied three pre-processing schemes: no pre-processing, pyramid decomposition, and image split. Additionally, they utilized five pre-trained convolutional networks. Their experimental evaluation on the Pandora dataset showed that the pre-trained convolutional networks outperformed the combined handcrafted descriptors with different pre-processing schemes, indicating the superiority of deep learning approaches in painting categorization.

Sandoval et al. (53) proposed a novel approach for fine-art style classification that incorporated transfer learning, patch-based classification, and a weighted aggregation scheme. The paintings are divided into sub-regions or patches, which serve as the units for classification. Each patch is individually classified using the features extracted from the deep neural network. The classification outcomes of the patches are then combined using a weighted sum, where the weights reflect the importance or significance of each patch in determining the overall stylistic label of the painting. They demonstrated enhancement in the accuracy of classifying artistic styles by considering the individual patches and their contributions to the overall stylistic label of the painting. Building upon their previous work, in (54), they extend their approach to fine-art style classification by proposing a two-stage classification system that involved individual-patch classification using a deep CNN, followed by applying a shallow neural network to the outcome probability vectors for the final classification decision. This approach improved the classification accuracy by effectively capturing both local variations and global stylistic patterns within the paintings in the classification process.

Afterwards, Sandoval et al. (55) focused on classifying fine-art paintings with simulated partial damages. They conducted experiments using a dataset comprising both damaged and non-damaged paintings and trained their models using a fine-tuned CNN (ResNet50) for feature extraction. They further employed a shallow Neural Network to classify the paintings into 20 different styles. The study found that training on a dataset that includes both damaged and non-damaged paintings resulted in a highly accurate classification of non-damaged artwork.

Menis et al. (56) employed transfer learning by training different convolutional neural networks (CNNs) and fine-tuning the hyperparameters on single-architecture models for style recognition. They then utilized ensemble learning, a technique combining the knowledge of multiple models using a meta-classifier. Furthermore, the authors investigated the impact of various data augmentation techniques. Their proposed ensemble approach led to improved results for style recognition

Pérez and Cozman (57) employed Generative Adversarial Networks (GAN) (58) for data augmentation in the context of painting style classification to explore the potential of using synthetic data to augment existing datasets to address the challenge of limited labelled datasets and class imbalance. Specifically, they focused on the performance of the pre-trained EfficientNet B0 model. They found that using synthetic paintings as additional training data helped enhance the model's ability to classify different artistic styles accurately.

Recently, Zhao et al. (6) compared ResNet with six modified versions of its architecture (RegNet, ResNeXt, Res2Net, ResNeSt, and EfficientNet B3) to classify paintings based on their style, artist, and genre. Their experiment on three different painting datasets showed that the pre-trained models on ImageNet produced the best results for art classification. Additionally, they showed the effectiveness of transfer learning in improving the classification accuracy of models.

3.2.2.1.3 Multi-task learning approaches: To address the limitations of traditional image classification models to capture the complex relationships between the different elements of a painting, such as the genre, style, and artist. More recent studies have explored the use of multitasking learning approaches in analysing and classifying paintings. The aim was to predict more than one category at the same time.

In conventional image classification tasks, each image is typically assigned a single label or category. However, in multi-task learning (MTL) context, researchers train models to classify images into multiple categories or labels simultaneously. This enables a single image to be associated with multiple labels or categories (59). Figure 3.6 illustrates the diagram of the multitask learning approach for paintings according to genre, style, and artist.

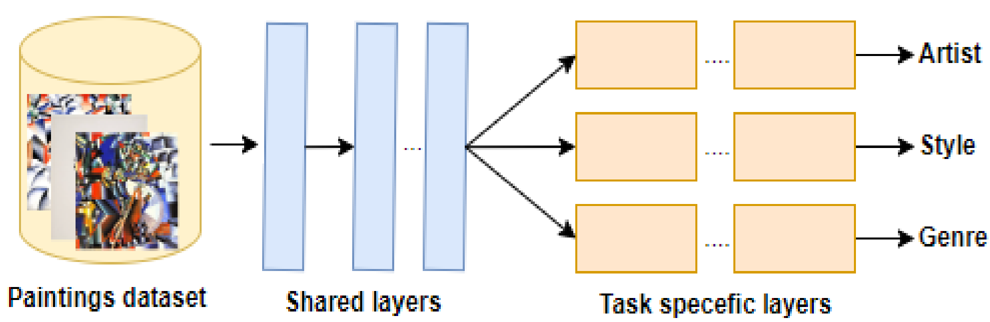


Figure 3.6: Diagram of multitask learning approach for paintings classification

Bianco et al. (60) presented a novel approach to categorizing paintings using a deep multibranch neural network by artist and style. They took various crops from the images at different scales and fed them into a multi-branch deep neural network. The network consisted of multiple branches, each specialized in extracting features from different image regions. The features extracted by each branch are then concatenated and fed into a fully connected layer for final classification. Later on, they proposed another approach (61), in which a multi-task classification module with an injection of the HOG feature is fed three patches (one from the down-sampled original image and two from the original version) to make predictions about the genre, artist, and style of paintings. The results on the Painting-91 dataset show that the proposed multi-task deep multibranch neural network outperformed the single-task deep multibranch neural network. In addition, they presented a new dataset called MultitaskPainting100k which consisted of 100,000 high-resolution digital images of fine art paintings. Each painting in the dataset is labeled with its corresponding style, genre, and artist.

Recently, Efthymiou et al. (62) proposed a novel multimodal architecture called ArtSAGENet, which is based on a graph neural network (GNN) combined with a convolutional neural network (CNN) to learn both visual and semantic-based representations of fine art paintings. They also utilized the benefits of multi-task learning in the analysis of fine art paintings by training the model on multiple tasks simultaneously, such as creation period estimation, style classification, artist attribution, and tag prediction; the model can learn shared representations that capture the relationships between these different tasks.

3.2.2.1.4 Hierarchical approaches To explore the application of deep neural networks for the hierarchical classification of fine-art paintings, Mohammadi et al. (63) introduced a novel hierarchical approach to categorize related artistic styles by developing a hierarchical system. Which aimed to group similar styles into super styles referred to as parents. They designed a parent classifier and multiple child classifiers to identify both the super style and the individual style. Figure 3.7 illustrates the proposed clustering system. The experimental evaluation of their approach using the WikiArt dataset demonstrated an improvement in the average F1 score of the DenseNet121 network.

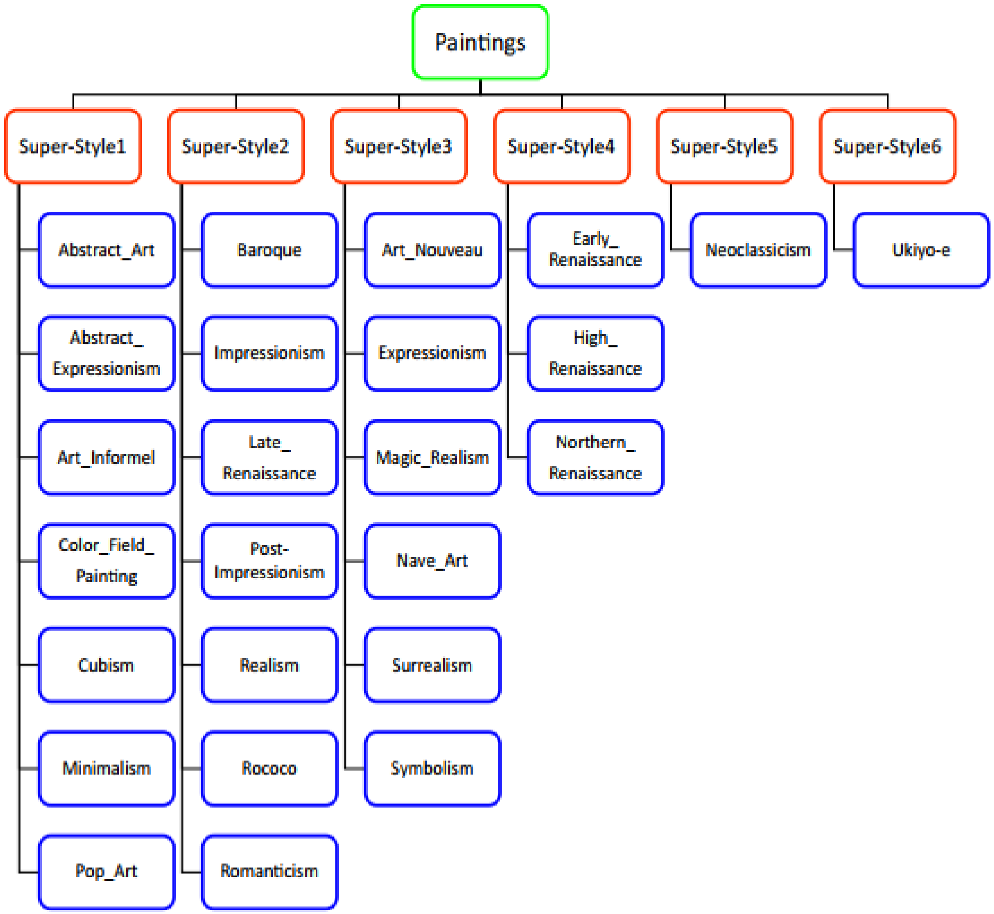


Figure 3.7: Diagram of hierarchical approach for clustering 25 styles into 6 super-styles

3.2.2.2 Unsupervised learning approaches for style classification

In image classification, unsupervised learning approaches are used to extract useful patterns from data without the need for labeled data or prior knowledge of the categories or classes to which the data belongs. These approaches involve training machine learning models where labeled data is not available or limited. Figure 3.8 illustrates the diagram of the unsupervised learning approaches for style classification which has been applied in some recent studies.

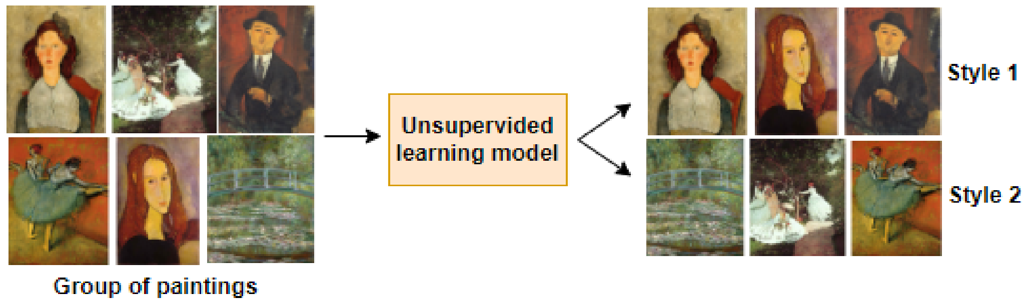


Figure 3.8: Diagram of unsupervised learning approach for style classification

Lee et al. (64) was one of the first studies that introduced an unsupervised learning method for style classification in art. They utilized a dataset of 1633 paintings and extracted composition-based local features through object segmentation and global features through color-based statistical computation. These features were then employed to train a self-organizing map (SOM), an unsupervised learning technique, to cluster the paintings into four distinct styles based on their similarity. The approach demonstrated the potential of unsupervised learning for style classification, providing insights into the clustering and organization of paintings based on their visual characteristics. However, the study did not explore the performance of the approach on larger or more diverse datasets, and the reliance on handcrafted features may limit its generalizability.

Later on, Gultepe et al. (65) presented another unsupervised approach for classifying digital images of paintings by artistic style. They proposed to extract features using the nonlinear unsupervised feature learning with the K-means (UFLK) technique from a dataset of 6776 paintings images, and after that, they used these extracted features as an input to an SVM classifier to classify the paintings into eight different stylistic categories. In addition, they employed spectral clustering to categorize the paintings by style. The experimental results demonstrated the effectiveness of their approach by achieving high accuracy in predicting and grouping paintings by style. However, the study did not explore the generalizability of the approach to other datasets or consider other factors such as artist attribution or genre classification.

A deep clustering model was proposed by Castellano and Vessio (66) based on the Deep Convolutional Embedding Clustering (DCEC) framework introduced in (67) and unsupervised learning techniques for clustering 8,446 digitized paintings from nine artistic styles based on their visual similarity. They used a pre-trained convolutional auto-encoder to extract features from digitized paintings and applied t-Distributed

Stochastic Neighbor Embedding (t-SNE) (68) to decrease the dimensions of these features. Then, they applied k-means clustering to the reduced feature vectors to group similar paintings together. They also evaluated their approach on a subset of 439 Pablo Picasso paintings. The experiments demonstrated the effectiveness of their proposed approach.

Lately, Sandoval et al. (69) proposed a novel unsupervised approach to labeling fine art paintings that uses adversarial training. The suggested approach combines supervised classification with unsupervised clustering using an optimization algorithm that iteratively improves the clustering procedure in accordance with predetermined goal criteria. The authors evaluated the proposed method on three different datasets of paintings and compared it to several other unsupervised labeling methods. The results show that their approach outperformed the other methods in terms of accuracy and efficiency.

Although unsupervised learning approaches can be useful for classifying fine-art paintings by style and potentially saving time and resources in the labeling process, they have been less explored because they require domain expertise in fine art to interpret the results.

3.3 Summary of literature review for style classification

Table 3.3 summarizes the principal results of fine art painting classification by style from literature in order of appearance. The table specifies the methods used, the dataset(s), the number of styles used in the evaluation, and the classification accuracy. For each work, we mention only the best result for style classification.

Table 3.3:: Papers on paintings style classifications with their related method and best performance

Article	Method	# styles	Dataset	Accuracy
Shamir et al. (2010)(24)	Handcrafted features, WNN as a classifier.	3 styles	513 images	91%
Arora and Elgammal (2012)(25)	Handcrafted features, SVM as a classifier.	7 styles	490 images	65.4%

Karayev et al (2013)(43)	Handcrafted features, Use AlexNet as a feature extractor	25 styles	WikiArt 85k images	47.3%
Khan et al (2014)(27)	Combine low-level features, SVM classifier with an X^2 kernel	13 styles 91 artists	Paintnig-91 2338 images	62.2%
Bar et al (2014)(44)	Binarized Features derived from a deep neural network, K-NN as a classifier as classifier	27 styles	WikiArt 47,724images	43.0%
Agarwal et al (2015)(26)	Handcrafted features, libsvm with X^2 kernel classifier	10 styles 6 genres	WikiArt 1800 in genre 3000 in style	62.73%
Saleh and Elgammal (2015)(70)	Combining (GIST, Classeme, Picodes, and Alexnet as a feature extractor) , compressing them using (PCA), SVM classifiers	27 styles 10 genres 23 artists	WikiArt 78,449 images	45.97%
Peng et Chen (2015)(45)	Cross-layer features from a cascade of six AlexNet models, SVM as a classifier.	13 styles	Painting-91 2338 images	69.2%
Tan et al. (2016)(5)	Fine-tuned CNN (AlexNet)	27 styles 10 genres 23 artists	WikiArt 78.449 images	54.50%
Lee et al (2016) (64)	Global and local features, Self-organizing map (SOM)	4 styles	1633 images	/
Chu et al. (2016)(46)	Gram matrix of feature maps (in VGG-19), SVM as the classifier.	25 styles	WikiArt	58.19%
Lecoutre et al (2017)(48)	Fine-tuned CNN (ResNet50)	25 styles	WikiArt 80.000 images	62.8%
Kedia (2017)(49)	Fine-tuned CNN (VGG-19)	27 styles	WikiArt 81,449 images	65.4%

Bianco et al (2017) (60)	Multibranch deep neural network	13 styles 91 artists	Painting-91 2338 images	84.4%
Florea et al. (2017)(29)	Color histograms and topographical features, SVM as a classifier.	18 styles	Pandora18k 18,040 images	50.1%
Yu et al (2017)(50)	Fine-tuned CNN (InceptionV3)	18 styles	Pandora18k 18,040 images	56.6%
Florea et al (2018)(30)	Color histograms and topographical features Boosted SVM classifiers, soft voting by an expert committee.	18 styles	Pandora18k 18,040 images	63.5%
		25 styles	WikiArt 85,000 images	46.2%
Bianconi et al(2018) (52)	Fine-tuned CNN (ResNet50)	12 styles	Pandora7k 7724 images	67%
Elgammal et al (2018) (51)	Fine-tuned CNN (ResNet152)	20 styles	WikiArt 76,921 images	63.7%
Gultepe et al. (2018)(65)	Unsupervised feature learning with K-means (UFLK), SVM classifier	8 styles	6776 images	/
Falomir et al. (2018) (28)	color similarity, qualitative color descriptors and quantitative global features, K-NN as a classifier	3 styles	Painting-91 252 images	65%
Sandoval et al. (2018) (53)	Fine-tuned CNN (InceptionV3), patch-based classification, weighted aggregation	6 styles	WikiArt 30,870 images	59.4 %
		19 styles	Pandora18k 19,320 images	70.2 %
Sandoval et al. (2019) (54)	Fine-tuned CNN (InceptionV3), patch-based classification, shallow neural network.	6 styles	WikiArt 30870 images	67.16 %

		22 styles	WikiArt 26,400 images	66.71 %
		19 styles	Pandora18k 19,320 images	77.53%
Bianco et al (2019) (61)	Use ResNet-18 + HOG features, patch-based classification	125 styles	Multitask-Painting100k	57.20%
Sandoval et al. (2020) (55)	Train on a dataset with damaged and non-damaged painting, Fine-tuned CNN (ResNet50), A shallow Neural Network	20 styles	Pandora	66.78%
Menis et al. (2020) (56)	Fine-tune (Inception-V3), stacking ensemble method	18 styles	Pandora18k 18.038 images	72.47%
		21 styles	Wikiart 80.039 images	68,55%
Mohammadi et al (2021) (63)	Hierarchical classification, use DenseNet121 network	25 styles	WikiArt 82,000 images	59.10%
Castellano and Vessio (2021) (66)	Pre-trained convolutional auto-encoder, (t-SNE) to decrease the dimensions of these features, k-means clustering	9 styles	8,446 images	/
Pérez and Cozman(2021) (57)	Generative Adversarial Networks (GAN), Fine-tuned (EfficientNet B0)	15 styles	WikiArt 63,659 images	74.40%
Zhao et al (2021) (6)	Fine-tuned CNN (EfficientNet)	13 styles	Painting-91 2,338 images	79.23%
		27 styles	WikiArt 81,444 images	69.97%
		125 styles	Multitask Painting100k 99,816 images	63.15%

Efthymiou et al (2021)(62)	Fine-tuned CNN (ResNet-34), Graph neural network	20 styles	WikiArt 75,921 images	77.6%

Table 3.3 highlighted the various methodologies that have been explored for style classification in the field of painting analysis. However, a significant challenge in comparing classification results arises from the variations in the number of images and style categories across different studies, even when the same dataset is used for evaluation. As a result, it becomes difficult to directly compare the performance of different approaches due to the lack of standardized evaluation criteria.

3.4 Paintings datasets:

Several fine-art painting datasets are available for researchers and practitioners interested in studying and analyzing fine art. Each dataset has its own characteristics, size, and labeling methodology.

Table 3.4 displays the most used datasets in fine art style classification and the corresponding amount of painting images and their labels. Each dataset is presented bellow. From the table, we can notice the variance between the datasets in terms of the number of images, the number of classes in each label, and the complexity of label categories they provide.

Table 3.4:: Datasets of paintings with their amount number of images and labels

Name	#Images	Labels
Painting-91 (27)	4,266	91 artists 13 styles
Pandora (29)	7,724	12 styles
Pandora18k (29)	18,038	18 styles
Wikiart (43)	85k	25 styles
MultitaskPainting100k (61)	100k	1508 artists 125 styles 41 genres

3.4.1 Wikiart dataset

The WikiArt dataset (43), also known as WikiPaintings, is a widely used collection of art images sourced from museums, universities, town halls, and other institutions. It contains over 250,000 artworks, including paintings, sculptures, drawings, posters, and sketches, contributed and labelled by volunteers. The paintings subset of the dataset is most frequently utilized and includes over 85,000 high-resolution images of artwork pieces from various artists classified into 27 different artistic styles. While the dataset's annotations are not highly accurate due to their collaborative nature, the dataset remains a valuable resource for automated fine-art classification studies. Researchers leverage this dataset for tasks such as image classification, style recognition, artist identification, and genre classification. Figure 3.9 (71) presents samples of paintings in the WikiArt dataset.

Figure 3.9: Samples of the styles in the WikiArt dataset

Figure 3.10 presents the distribution of the images in the dataset. We notice that the number of samples representing different styles is highly imbalanced as the numbers of images for different styles vary between 98 and 12000.

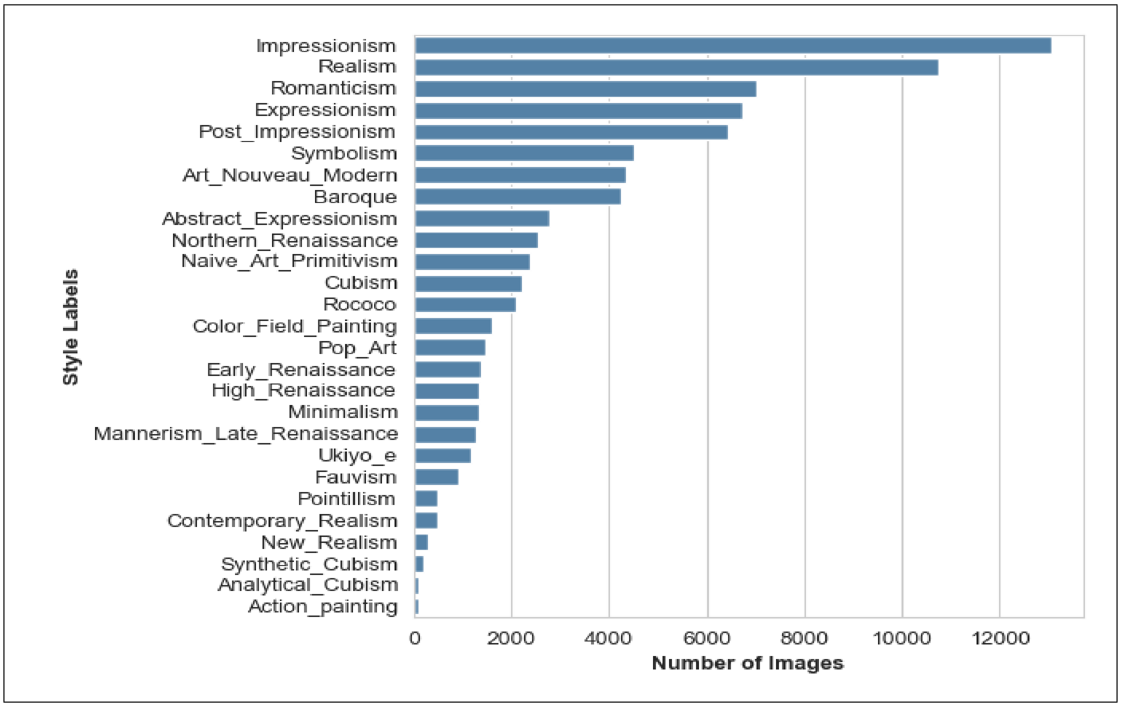
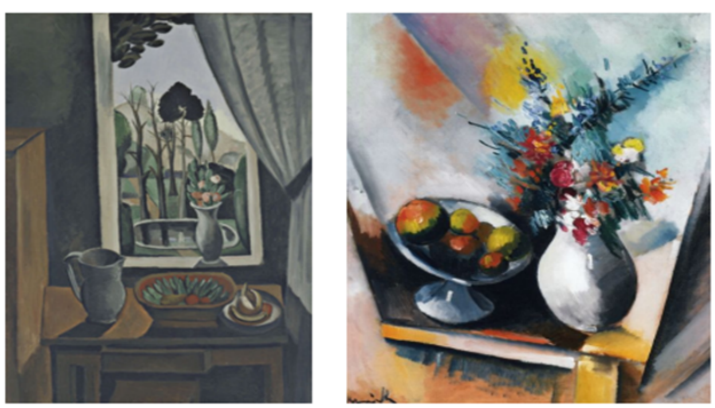


Figure 3.10: Distribution of images in the WikiArt dataset

Although WikiArt is one of the largest datasets with paintings labeled by style, it contains some misclassified paintings because public volunteers on the Wikiart.org website (72) made their labels. Figure 3.11 (73) shows a misclassified example from the class cubism. With this misclassification in the dataset, the classifier will not be able correctly to identify the corresponding style to the painting.



(a) A cubism image, misclassified as realism. "Window at Vers" by André Derain (1912).

(b) A cubism image, misclassified as art-nouveau. "Fruit And Flowers" by Maurice de Vlaminck (1910).

Figure 3.11: Two randomly selected cubism images that are misclassified in the Wikiart dataset

3.4.2 Paintings-91 dataset

The Painting-91 dataset includes a total of 4,266 painting images created by 91 different artists. They are classified according to the artist and the style. There are a total of 2,338 paintings that were created by a total of 50 different artists. These paintings have been categorized according to one of 13 different artistic styles namely: cubism, abstract expressionism, baroque, constructivism, pop art, impressionism, neoclassical, postimpressionism, renaissance, romanticism, realism, symbolism, and surrealism. 1250 of them were utilized for training, while 1088 of them were used for testing. This dataset, created by Khan et al.(27), is one of the most often utilized datasets for classifying artists and styles. Figure 3.12 shows a few examples from the dataset; each picture has its corresponding style and artist.

Figure 3.12: Examples of paintings from Painting-91. Each image has two labels: artist and style.

Figure 3.13 presents the distribution of the images in the dataset, where the total number of images for each style ranges from 80 to 280.

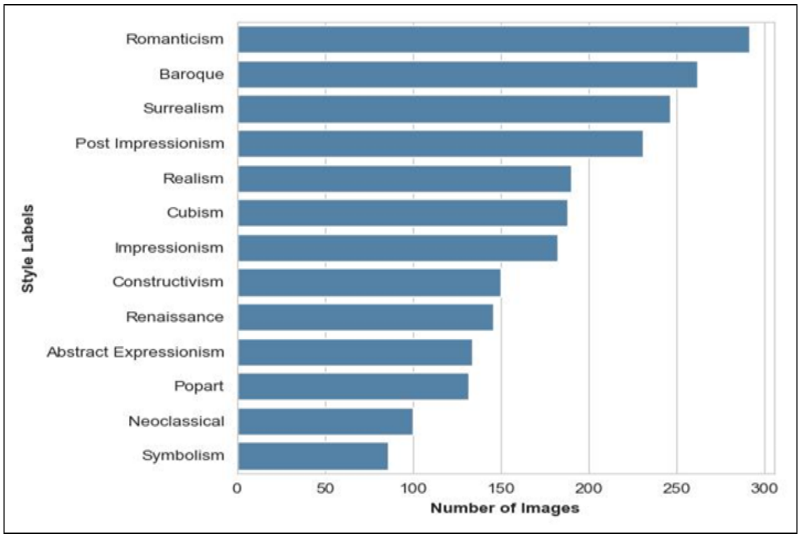


Figure 3.13: Distribution of images in the Painting-91 dataset

3.4.3 Pandora dataset

Florea et al. (74) introduced the PANDORA (Paintings Dataset for Recognising the Art movement) dataset to address the need for a more balanced fine-art painting dataset. The initial version, known as PANDORA, comprises approximately 7,724 images representing 12 distinct artistic styles. Figure 3.14 presents samples of the 12 different artistic movements in the dataset.

Figure 3.14: Samples of the 12 classes in the Pandora7k dataset

Building upon the success of the PANDORA dataset, a subsequent version called PANDORA18K was introduced (30; 75). This expanded dataset contains a more extensive collection of 18,038 paintings distributed across 18 unique artistic styles. By increasing the number of images and styles, PANDORA18K provides an even more comprehensive resource for studying and exploring the various art movements and their characteristics. Figure 3.15 presents samples of the 18 different artistic movements in the dataset.

Figure 3.15: Samples of the 18 classes in the Pandora18k dataset

Table 3.5 shows the historical eras of the artistic styles included in the Pandora18K dataset. The list begins with the ancient movement of Byzantine Iconography, goes through three significant phases of art, namely Renaissance, Baroque, Realism, and ends with the modern art movements(76).

Table 3.5:: List of the artistic movements in the Pandora18K dataset

Artistic Movement	Historical period
Abstract Art	1910–now
Baroque	1590–1725
Byzantine Iconography	500–1400

Cubism	1907–1920
Early Renaissance	1280–1450
Expressionism	1905–1925
Fauvism	1905–1908
High Renaissance	1490–1527
Impressionism	1860–1950
Naïve Art	1890–1950
Northern Renaissance	1497–1550
Pop Art	1950–1969
Post Impressionism	1860–1925
Realism	1880–1880
Rococo	1650–1850
Romanticism	1770–1880
Surrealism	1920–1940
Symbolism	1850–1900

Figure 3.16 presents the distribution of images in the dataset, where the total number of images for each style ranges from 700 to 1200.

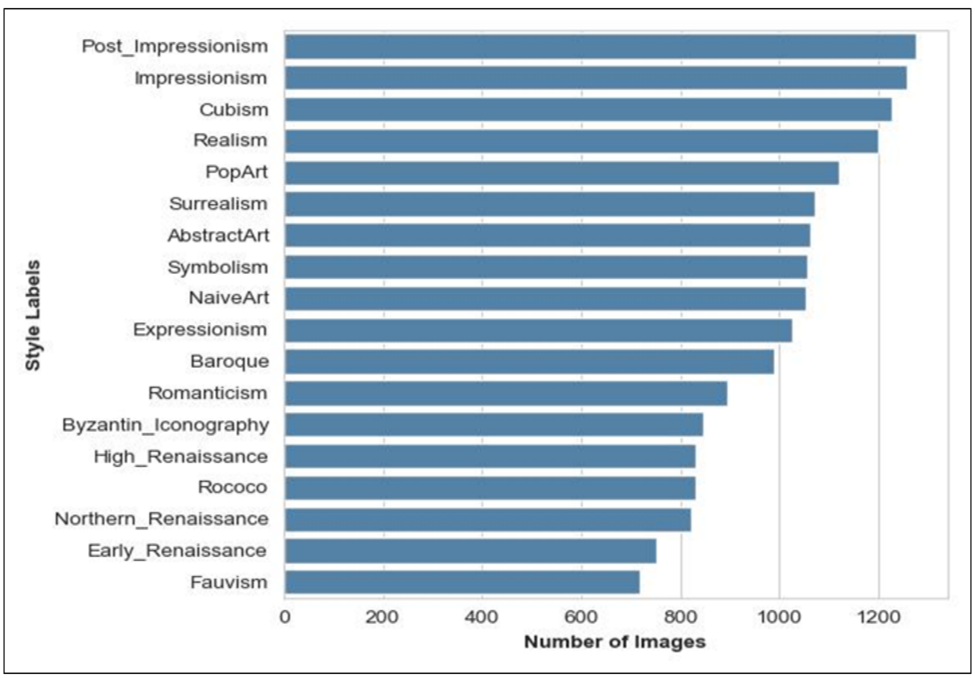


Figure 3.16: Distribution of images in the Pandora18k dataset

The high validity of the Pandora dataset can be attributed to the fact that it was subjected to two very stringent review processes: a visual inspection check, in which images of poor quality were eliminated, and a fine-art expert review, in which annotations and labels were refined and verified. Experts in their respective fields performed both of these processes.

3.4.4 MultitaskPainting100k

The "MultitaskPainting100k" dataset was originally collected by Bianco et al. (61) and subsequently adapted for multitask learning involving artist, style, and genre classification. The dataset consists of approximately 100,000 paintings, sourced primarily from WikiArt.org (72). It includes artwork from 1508 different artists, covering 125 distinct styles and 41 genres. Figure 3.17 presents samples of the 125 different styles in the MultitaskPainting100k dataset.

Figure 3.17: Samples of the styles in the MultitaskPainting100k dataset

3.5 Evaluation metrics in image classification

In image classification, various evaluation metrics are used to assess the performance of classification models. These metrics provide insights into the accuracy, precision, recall, and overall effectiveness of the classification process. Some commonly used evaluation metrics in image classification include:

- **Accuracy**: It measures the overall correctness of the classification model by calculating the ratio of correctly classified images to the total number of images (77). It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

  True Positive(TP): the number of positive class samples the model predicted correctly.
  True Negative(TN): the number of negative class samples the model predicted correctly.
  False Positive(FP): the number of negative class samples the model predicted incorrectly.
  False Negative(FN): the number of positive class samples the model predicted incorrectly.

- **Precision**: It measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). It indicates the model's ability to avoid false positives (77). It can be defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

- **Recall (Sensitivity or True Positive Rate)**: It measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It indicates the model's ability to identify all positive instances (77). It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

- **F1 Score**: It is the harmonic mean of precision and recall, providing a balance between the two metrics. The F1 score combines precision and recall into a single value, which is useful when the class distribution is imbalanced (77).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

- **Receiver Operating Characteristic (ROC) Curve**: It is a graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) at various classification thresholds. The area under the ROC curve (AUC) is a commonly used metric to evaluate the performance of a classification model(77).

- **Precision-Recall Curve**: It is a graphical representation of the trade-off between precision and recall at various classification thresholds. It helps evaluate the model's performance when dealing with imbalanced datasets (77).

- **Confusion Matrix**: It is a table that provides a detailed breakdown of the model's performance by showing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. It helps in understanding the types of errors made by the model. Figure 3.18 presents the confusing matrix in testing a predictor. All the testing samples are divided into four categories according to the real labels and the prediction results.(77)

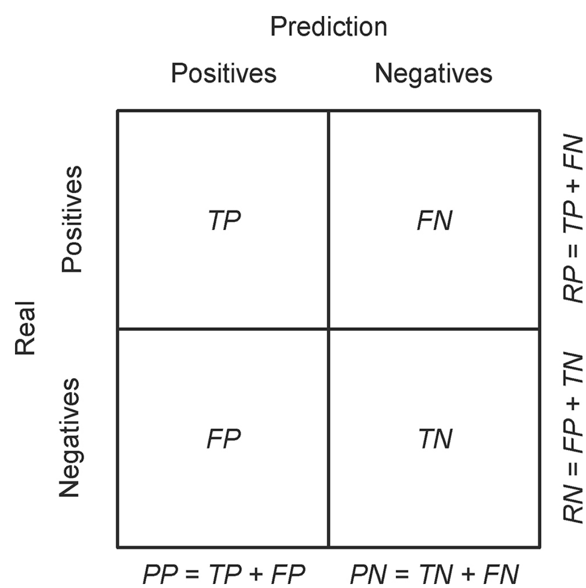


Figure 3.18: Confusion matrix in testing a predictor

These evaluation metrics provide valuable insights into the performance of image classification models and assist in comparing different approaches or fine-tuning the models to achieve better results. The choice of metrics depends on the specific requirements of the task and the class distribution of the dataset.

3.6 Conclusion

The literature review presented in this chapter showed that the field of painting style classification has witnessed significant advancements through the application of various methodologies. Researchers have explored different feature extraction techniques, from handcrafted features to deep neural networks.
One prevalent technique employed in recent approaches is transfer learning, which involves adapting pre-trained models from object classification to artistic categorization. This technique provided advantages in the training process and has demonstrated good performance.
Moreover, the analysis extends beyond single-label classification, with studies incorporating multi-task learning to predict multiple categories simultaneously. This allows for a more comprehensive understanding of art pieces and their complex relationships. However, the results across the studies indicate that there is still room for improvement in achieving higher accuracy rates.
Another significant observation is that most existing fine-art classification studies, including classical and deep learning approaches, rely on supervised techniques that necessitate high-quality art expert annotations. On the other hand, exploring categorization systems from an unsupervised perspective, without the need for human annotations, has been relatively limited.

In the next chapter, we present our first contribution to recognizing the artistic style of fine art painting and its experimental results.

Chapter 4

Recognizing the Style of a Fine-art Painting with EfficientNet and Transfer Learning

4.1 Introduction

This chapter presents the description of our first research contribution to the field of recognizing the artistic style of fine art paintings and its experimental results.

In this contribution, we focus on investigating the effectiveness of the EfficientNet family in combination with transfer learning. By using the transfer-learning technique, we aim to benefit from the knowledge learned from a large dataset of general natural images and adapt it to the specific task of style classification. The methods are evaluated using the Painting-91 standard fine art classification dataset.

This chapter is structured into six sections. An introduction and a summary of the related studies are presented in Section 4.2. The following sections provide details of the proposed methods described in Section 4.3. Section 4.4 covers the experimental validation, including information on the datasets used, data preprocessing techniques, a presentation of the architecture employed, and an overview of the experimental setup details. Section 4.5 presents and discusses in detail the results obtained from the experiments. Finally, the chapter concludes with Section 4.6, where the key findings of the study are summarized.

The first contribution was presented at the 7th International Conference on Image and Signal Processing and their Applications (ISPA 2022). The published paper is titled “Recognizing the style of a fine-art painting with EfficientNet and Transfer learning” and can be accessed at https://ieeexplore.ieee.org/document/9786371/

4.2 Background

The digitization of painting databases has grown rapidly in recent years, presenting challenges in manual content manipulation. This has led to open a new research area focused on providing automated tools to assist the artistic community in analyzing, classifying, and gaining a deeper understanding of paintings.

The utilization of transfer learning in fine-tuning pre-trained Convolutional Neural Networks (CNN) has demonstrated remarkable efficacy in automatically recognizing the artistic style of fine art paintings (5; 48; 49; 50; 51; 54). However, these models often require a fixed input size, which can lead to information loss when resizing images. To address this concern and to investigate the impact of the input size, the network depth, and the deep retraining on model performance, we suggest evaluating the EfficientNet models for artistic style classification. We chose these architectures because they offer a scaling model that balances the network's depth, width, and resolution.

While EfficientNet models have shown exceptional performance in natural image classification tasks, their application in painting classification has received limited attention. Previous studies focused mainly on EfficientNetB0 in(57) and EfficientNetB3 in (6), but our research expanded the investigation to include seven models from the EfficientNet family, ranging from B0 to B6, allowing for a comprehensive comparison in style classification. Additionally, we enhanced the base architecture of the EfficientNet models by adding additional layers to create our custom models. Furthermore, we explored the impact of deep retraining the last layers of a pre-trained model on the accuracy of style recognition.

4.3 Proposed methodologies

In this section, we present our three proposed methodologies in this study for recognizing the artistic style of fine art paintings.

4.3.1 Pre-trained EfficientNet models for style classification

In order to investigate the efficacy of the pre-trained EfficientNet models for style classification, we proposed a fine-tuning strategy to adapt these models (ranging from B0 to B6) to our specific task. This involved replacing the last fully connected layers of each model with a softmax layer, with the number of classes in our dataset determining the dimensionality of the output. To address potential overfitting concerns during the training, we added batch normalization and dropout layers. The output of the models provided the probabilities for each possible artistic style category, indicating the probability of a painting image belonging to a particular style.
For fine-tuning, we used the EfficientNet pre-trained models on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset (47) that contains 1.2 million natural images of objects with 1000 categories as a starting point, initially freezing all layers except for the last fully connected layer, which was trained from scratch. Figure 4.1 presented the framework of the proposed pre-trained EfficientNet models for style classification

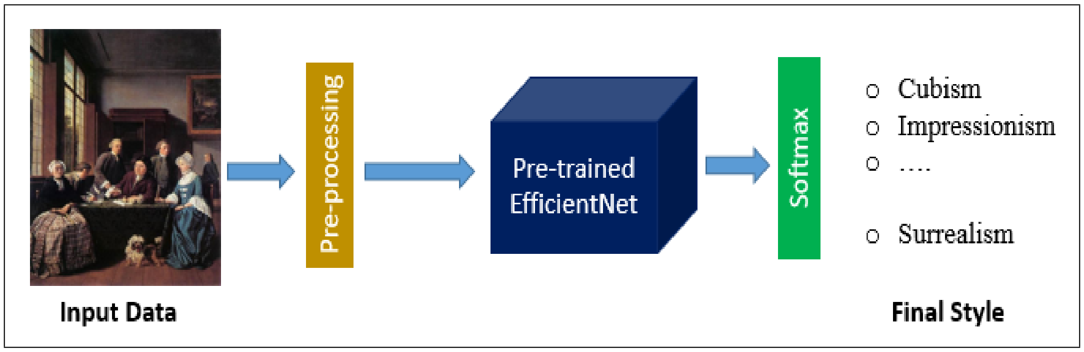


Figure 4.1: General framework of the pre-trained EfficientNet models for style classification

4.3.2 Custom pre-trained EfficientNet models for style classification

In order to enhance the capability of pre-trained EfficientNet models for style classification, we introduced custom models specifically designed for this task. The modification involved replacing the last fully connected layers of each pre-trained model with two dense layers, which had a ReLU (Rectified Linear Unit) activation function (78) to introduce non-linearity and improve the model's capacity to learn complex patterns. Additionally, batch normalization and dropout layers were added after the dense layers to improve training stability and reduce overfitting. Finally, we included a softmax layer at the end of the model, with the corresponding number of classes in the dataset. These layers are randomly initialized. Figure 4.2 presented the overall framework of our custom pre-trained EfficientNet models for style classification.

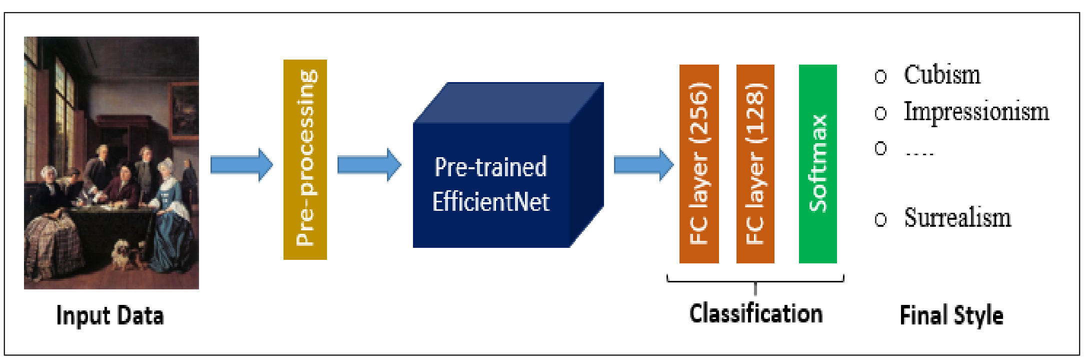


Figure 4.2: General framework of our custom pre-trained EfficientNet models for style classification

4.3.3 Deep retraining of the custom pre-trained EfficientNet models for style classification

To investigate the effects of deep retraining, we focused on unfreezing the last eight blocks of our custom EfficientNet models, ranging from B0 to B6. By unfreezing these blocks, we allowed their weights to be updated during training. Additionally, we included the last fully connected layers in this retraining process, initializing them randomly. This approach enabled us to examine the influence of deep retraining on the performances and capabilities of our models. Figure 4.3 illustrates the framework of deep retraining of the custom pre-trained EfficientNet models for style classification.

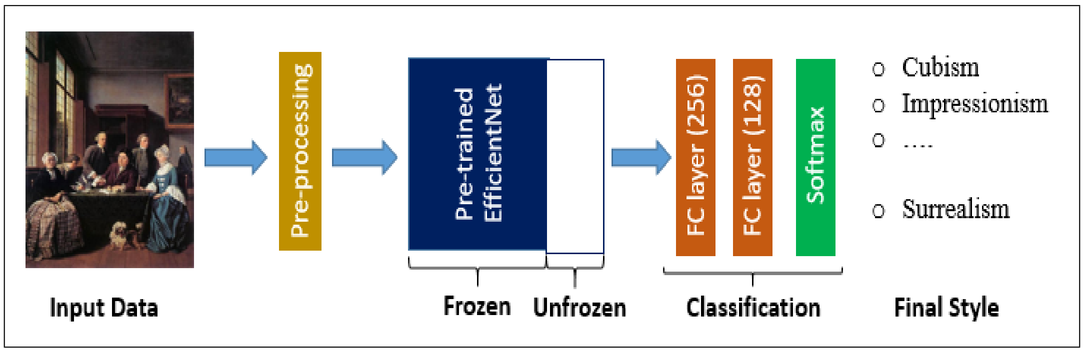


Figure 4.3: General framework of deep retraining of the custom pre-trained EfficientNet models for style classification

4.4 Experimental validation

We evaluated the proposed methodologies with the seven pre-trained EfficientNet models from B0 to B6 for style classification using the fine-art painting classification Painting-91 dataset. In this section, we present the used dataset, the data preprocessing steps, the EfficientNet architecture, and the training setup used in our evaluation.

4.4.1 Exprimental dataset: Painting-91

To evaluate the proposed methodologies, we used the Painting-91 dataset (27). It consists of a total of 2,338 paintings categorized according to one of 13 different artistic styles, namely: cubism, abstract expressionism, baroque, constructivism, pop art, impressionism, neoclassical, postimpressionism, renaissance, romanticism, realism, symbolism, and surrealism. 1250 of them were utilized for training, while 1088 of them were used for testing. Figure 4.4 shows the percentage of different styles in the Painting-91 dataset.

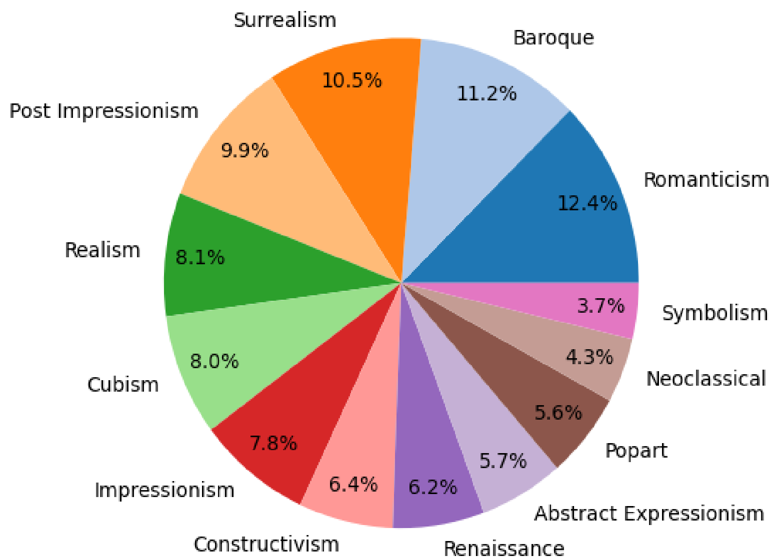


Figure 4.4: Percentage of different styles in the Painting-91 dataset

4.4.2 Data pre-processing

In the initial stage of our process, we performed data pre-processing to prepare the input images for the models. This involved resizing all the images to match the specific input size required by each model. Each pre-trained model has its own input size, so we ensured that all images were adjusted accordingly.

We applied data augmentation techniques to enhance the diversity and robustness of the training data. These techniques introduce variations to the input images, thereby increasing the model's ability to generalize. Some of the augmentation techniques we employed included rotation within a range of 5 degrees, adjusting the width and height within a range of 0.1, horizontal flipping to create mirror images and small-scale zooming within a range of 0.2.

To prevent overfitting, we utilized EfficientNet's preprocessing input. EfficientNet is a powerful convolutional neural network architecture that incorporates its own preprocessing steps. By employing this preprocessing input, we aimed to further enhance the model's performance and generalization capabilities.

For the test data, we followed a similar procedure. We resized all the test images to match the specific input size required by the model under evaluation. Additionally, we normalized the images using the mean and standard deviation of the dataset. Normalization is a common practice in machine learning, as it helps to scale the pixel values and make them more suitable for the model to process effectively.

By implementing these pre-processing steps, including resizing, data augmentation, and normalization, we aimed to ensure consistency in the input data and improve the model's ability to learn and make accurate predictions.

4.4.3 EfficientNet architecture

EfficientNet is a family of convolutional neural network models that were proposed by researchers at Google (42); it consists of eight models, labeled B0 through B7. The models use a compound scaling method, which is presented in Figure 4.5. It scales up or down the width(more channels), depth(more layers), and resolution (image size) of the baseline model EfficientNet B0 based on a set of fixed scaling coefficient Φ in a principled way:

$$Depth : d = \alpha^{\phi} \tag{4.1}$$

$$Width :: w = \beta^{\phi} \tag{4.2}$$

$$Resolution : r = \gamma^{\phi} \tag{4.3}$$

$$such\ that\ \ \alpha.\beta^2.\gamma^2 \simeq 2\ given\ all\ \alpha, \beta, \gamma \geq 1$$

φ controls all the desired dimensions and scales them together but not equally. α, β, γ tell us how to distribute the additional resources to the network.

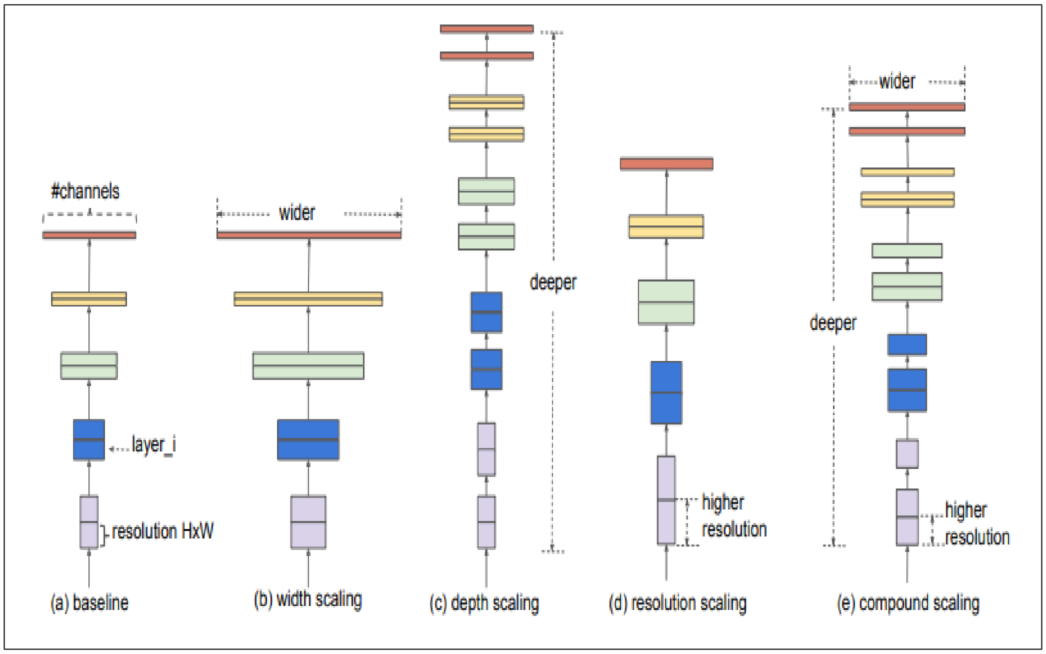


Figure 4.5: Scaling models.

Taking B0 as a baseline model, the authors developed a full family of EfficientNet models from B1 to B7, which achieved state-of-the-art accuracy on ImageNet dataset (47) while being very efficient compared to its competitors. Figure 4.6 presents the architecture of the EfficientNet B0 model, which is also summarized in Table 4.1. The architecture uses blocks of mobile inverted bottleneck convolution (MBConv) (79; 80) that is also called inverted residual block with an additional SE (Squeeze and Excitation) block (81). These two blocks are explained bellow.

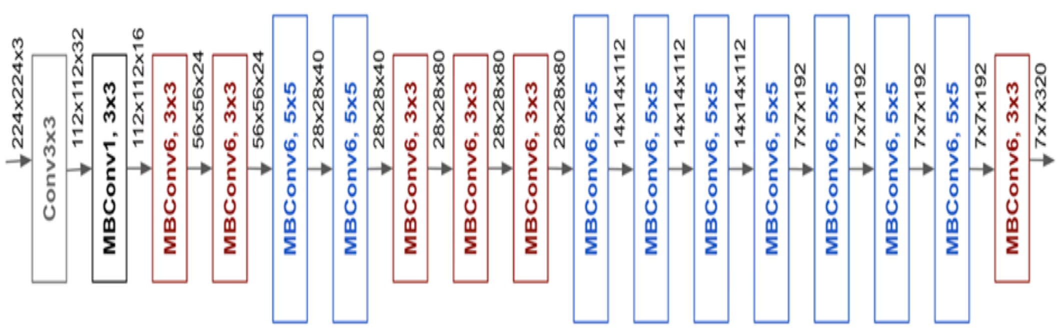


Figure 4.6: The EfficientNet-B0 architecture.

Table 4.1:: The composition of the EfficientNet B0.

Stages	**Operators**	**Resolution**	**#Channels**	**#Layers**
1	Conv3 × 3	224 × 224	32	1
2	MBConv1, k3 × 3	112 × 112	16	1
3	MBConv6, k3 × 3	112 × 112	24	2
4	MBConv6, k5 × 5	56 × 56	40	2
5	MBConv6, k3 × 3	28 × 28	80	3
6	MBConv6, k5 × 5	14 × 14	112	3
7	MBConv6, k5 × 5	14 × 14	192	4
8	MBConv6, k3 × 3	7 × 7	320	1
9	Conv1 × 1 & Pooling & FC	7 × 7	1280	1

- **MBConv block**: Mobile Inverted Residual Bottleneck Convolution (MBConv) block uses Depth-wise Separable Convolution(38), first, the channels will be widened by a point-wise convolution (conv 1x1) then uses a 3x3 depth-wise convolution that reduces significantly the number of parameters and finally it use a 1x1 convolution to reduce the number of channels so the beginning and the end of the block can be added (82). Figure 4.7 (80) presents the structure of the MBConv block.

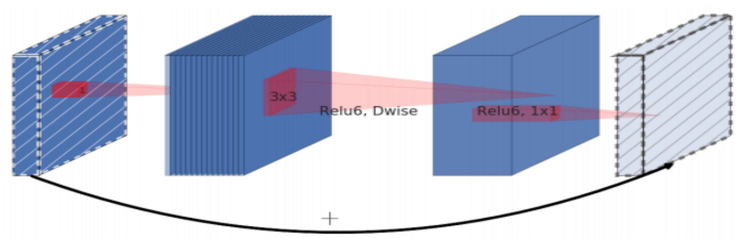


Figure 4.7: The structure of an inverted residual block (MBconv)

- **Squeeze and Excitation (SE) block**: SE is a building block for CNNs to improve the interdependencies between the channels by performing dynamic feature channel-wise recalibration; Rather than assigning equal weights to all channels, the SE Block dynamically assigns higher weights to the most important channels. This adaptive recalibration mechanism helps the network to focus on the most informative channels. Figure 4.8 (82) shows the components of this block.

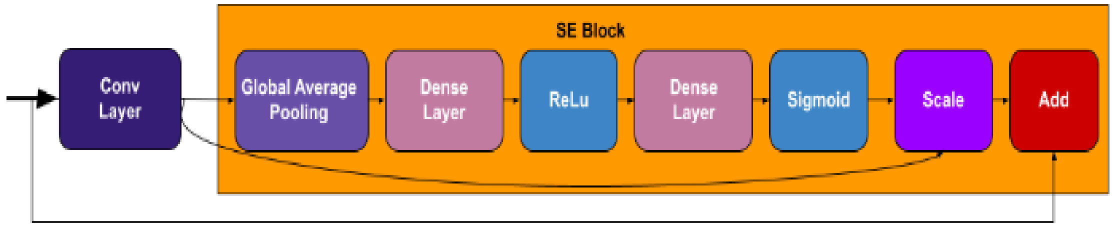


Figure 4.8: The squeeze and excitation (SE) block architecture

EfficientNet applies the SE block along the way with the MBConv block, resulting in the following structure illustrated in Figure 4.9 (82).

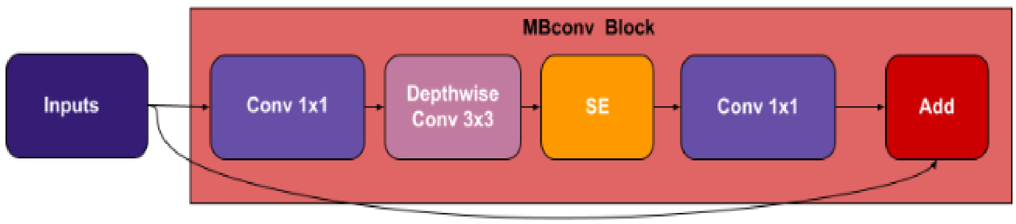


Figure 4.9: The MBConv block with SE block in the EfficientNet architecture

Table 4.2 presents the resolution and the number of parameters of each EfficientNet model. The resolution varies from 224 to 600, while the parameters range from 5.3 million to 66 million.

Table 4.2:: The resolution and number of parameters of EfficientNet models from B0 to B7.

Base model	**Resolution**	**#Parameters**
EfficientNetB0	224	5.3 M
EfficientNetB1	240	7.8 M
EfficientNetB2	260	9.2 M
EfficientNetB3	300	12 M
EfficientNetB4	380	19 M
EfficientNetB5	456	30 M
EfficientNetB6	528	43 M
EfficientNetB7	600	66 M

4.4.4 Experimental setup

In our experimental setup, we implemented several techniques to mitigate the risk of overfitting during the training process. One of the strategies we employed was the incorporation of Batch Normalization and Dropout layers before the softmax layer.

Batch Normalization is a technique that normalizes the inputs of each mini-batch during training. It helps stabilise and accelerate the training process by reducing the internal covariate shift, which is the change in the distribution of network activations as the parameters are updated. By normalizing the inputs, Batch Normalization enables smoother and more stable gradient propagation, leading to faster convergence and improved generalization (83).

Dropout, on the other hand, is a regularization technique that randomly deactivates a proportion of the neurons during training; it prevents the network from relying too heavily on any single neuron by randomly dropping out neurons and encourages the network to learn more robust and generalizable features (84). This regularization technique helps in reducing overfitting by effectively adding noise to the training process and forcing the network to learn redundant representations.

To optimize the models, we utilized the Adam optimizer(85), which is an adaptive learning rate optimization algorithm. It computes adaptive learning rates for each parameter based on the estimates of first and second-order moments of the gradients. The initial learning rate for Adam was set to 1e-2, and this value was fine-tuned during the training process.

In order to further enhance the learning process, we employed the ReduceLROnPlateau function (86). This function dynamically reduces the learning rate if the accuracy of the model does not improve over a certain number of epochs. In our case, if the accuracy did not improve after three consecutive epochs, we decreased the learning rate. This technique helps in fine-tuning the learning rate and finding a better local minimum in the optimization landscape.

Lastly, our experiments were conducted with a batch size of 64, which represents the number of samples propagated through the network before the weights are updated. A suitable batch size can significantly affect the training dynamics and generalization of the models.

By implementing these strategies, we aimed to prevent overfitting, optimize the learning process, and improve the performance and generalization of our models in style classification tasks.

4.5 Results and discussion

The evaluation metric used was accuracy, which is defined as the percentage of successfully identified examples relative to the total number of examples. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

Where true positive and true negative classification predictions are denoted by TP and TN, respectively, while false positive and false negative classification predictions are denoted by FP and FN, respectively (77).

4.5.1 Style classification accuracy for all experiments

Table 4.3 presents a comprehensive overview of the average accuracy results obtained from various classification tests. The table compares the performance of different models based on the EfficientNet architecture, with versions ranging from B0 to B6. The models are evaluated under three different configurations: base pre-trained, custom pre-trained, and custom pre-trained with deep retraining of the last eight layers.

Table 4.3:: The results of style classification

Model	**Base Pre-trained**	**Custom pre-trained**	**Custom pre-trained (8 layers)**
EfficientNetB0	68,47%	70.31%	71,32%
EfficientNetB1	68.56%	70.68%	72,06%
EfficientNetB2	69.12%	70.86%	72.89%
EfficientNetB3	69.30%	71,42%	73,53%
EfficientNetB4	70.04%	72.52%	73.90%
EfficientNetB5	70.68%	73.13%	74,54%
EfficientNetB6	**70.96%**	**73.47%**	**75.55%**

The base pre-trained models demonstrate decent accuracy, with values ranging from 68.47% (EfficientNetB0) to 70.68% (EfficientNetB5). However, the accuracy significantly improves when employing the custom pre-training approach. The custom pre-trained models achieve higher accuracy across the board, with values ranging from 70.31% (EfficientNetB0) to 73.13% (EfficientNetB5). Further enhancing the models by retraining the last eight layers leads to even better performance. The custom pre-trained models with deep retraining of the last eight layers yield accuracy values ranging from 71.32% (EfficientNetB0) to an impressive 75.55% (EfficientNetB6).

Among all the models, EfficientNetB6 stands out as the top performer, achieving the highest accuracy in all cases: the base pre-trained model, the custom pre-trained model, and the custom pre-trained with retraining the last eight layers configurations. With accuracy values of 70.96%, 73.47%, and 75.55%, respectively, EfficientNetB6 demonstrates its capability to recognize patterns and make reliable predictions accurately.

In the case of base pre-trained models, there is a noticeable 2% increase in accuracy from EfficientNet B0 to B6. Specifically, EfficientNet B0 achieves an accuracy of 68.47%, while EfficientNet B6 achieves a higher accuracy of 70.96%. This indicates that higher resolution and larger network size positively impact the accuracy of painting style classification. Conversely, resizing images to a smaller size can lead to geometric distortions, loss of details, and, subsequently, lower accuracy.

The performance of the custom pre-trained models surpasses that of the base pre-trained models. This highlights the effectiveness of the network size on classification accuracy. Despite having the same input size, the custom pre-trained EfficientNet models consistently outperform their base pre-trained counterparts. For instance, while the base pre-trained EfficientNet B5 achieves an accuracy of 70.68%, our custom pre-trained EfficientNet B5 achieves a higher accuracy of 73.12%.

Furthermore, deep retraining of the last eight layers in the custom models leads to further improvement in accuracy. This finding demonstrates the importance of fine-tuning the model's parameters to optimize its performance. Notably, EfficientNet B6 exhibits the best results among all configurations, achieving an impressive accuracy of 75.55%.

Overall, the results emphasize the impact of network size, resolution, and fine-tuning on the accuracy of style classification in paintings. The larger and more complex models consistently yield higher accuracy, highlighting the significance of capturing intricate visual details in fine art analysis. These findings contribute to the advancement of automated image classification techniques, specifically in the domain of style recognition in paintings.

4.5.2 Classification results for style classification

The results presented in Figure 4.10 provide valuable insights into the impact of additional layers and deep retraining on the accuracy of different models within the EfficientNet family, ranging from B0 to B6.

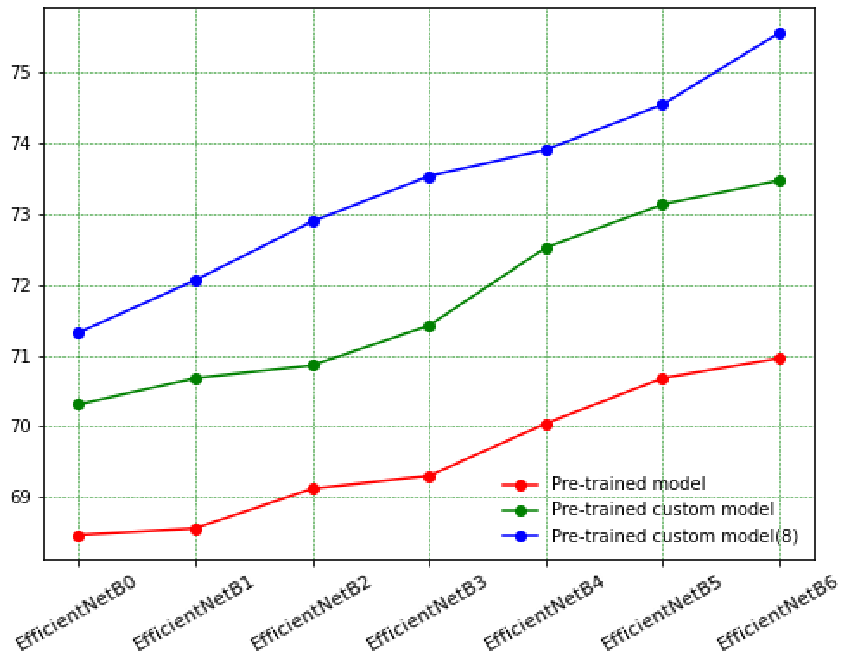


Figure 4.10: Classification results of the base pre-trained EfficientNet, our custom model, and the results of retraining the last 8 layers

One notable observation is that each model demonstrates improved accuracy when extra layers are added to its architecture and deep retraining is applied. This finding suggests that adding extra layers enables the models to capture more complex and hidden features, leading to improving the classification performance. The incremental improvement in accuracy across the EfficientNet models highlights the effectiveness of this approach in boosting the models' capabilities.

Interestingly, the results also reveal that deep retraining of the custom EfficientNet B2 model yields higher accuracy compared to the custom EfficientNet B3 and EfficientNet B4 models, despite the latter two having higher input resolution and larger model size. This finding suggests that the deep retraining technique plays a crucial role in improving the performance of the network, surpassing the influence of increased resolution and network size alone. It highlights the significance of fine-tuning the model's parameters to optimize its performance, even when faced with resolution and model size variations.

4.5.3 Confusion matrices

Figure 4.11 and Figure 4.12 present the confusion matrix of the pre-trained EfficientNet B6 base model and our re-trained custom EfficientNet B6 model, respectively, for style identification on the Paintings-91 dataset. The confusion matrix provides a comprehensive view of the classification accuracy for each individual style, where each number corresponds to a specific style label and 1(Ab-Expr) refers to Abstract Expressionism, 2: Baroque, 3(Construc): Constructivism, 4: Cubism, 5(Impress): Impressionism, 6(N-class): Neo-Classical, 7: Popart, 8(P-Impr): Post Impressionism, 9: Realism, 10(Renaiss): Renaissance, 11(Roman): Romanticism, 12(Surreal): Surrealism and 13(Symbol): Symbolism.

Analyzing the confusion matrices, we observe notable improvements in the accuracy of certain style classes when comparing the base model to our re-trained custom model. Specifically, the re-trained custom EfficientNet B6 model exhibits significant enhancements in the accuracy of Post-Impressionism and Romanticism, with improvements of 13% and 10%, respectively. This can be attributed to the efficacy of retraining the last layers of the model, which capture high-level features crucial for discriminating between these styles. Moreover, Abstract Expressionism and Baroque also demonstrate improved accuracy, with gains of 5% and 8%, respectively.

However, it is worth noting that the accuracy improvement for Surrealism is relatively small, with only a 1% increase. Additionally, both models struggle with accurately classifying Renaissance style, achieving a relatively low accuracy of 50% and frequently confusing it with Baroque and Neo-Classical styles. This challenge arises from the close proximity and overlapping characteristics of these styles, making it difficult to distinguish them. Furthermore, the model encounters confusion between Symbolism and Constructivism, achieving an accuracy of 67% for Symbolism.

Despite these challenges, our best accuracy is achieved in the Post Impressionism style, reaching an impressive 91%. This result highlights the model's proficiency in recognizing the distinct features and patterns associated with this particular style.

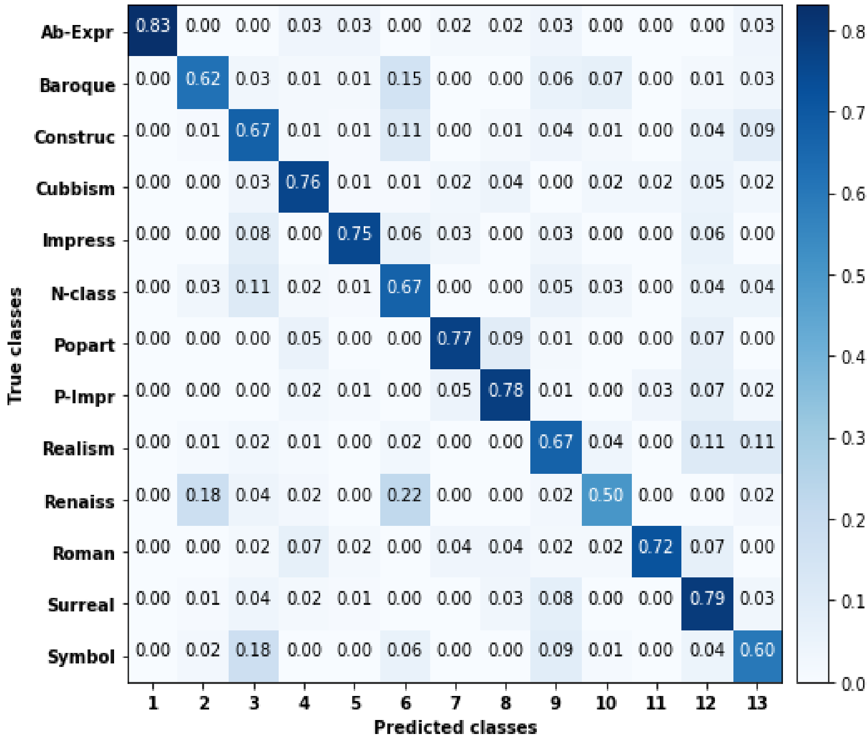


Figure 4.11: Confusion matrix of the pre-trained EfficientNet B6.

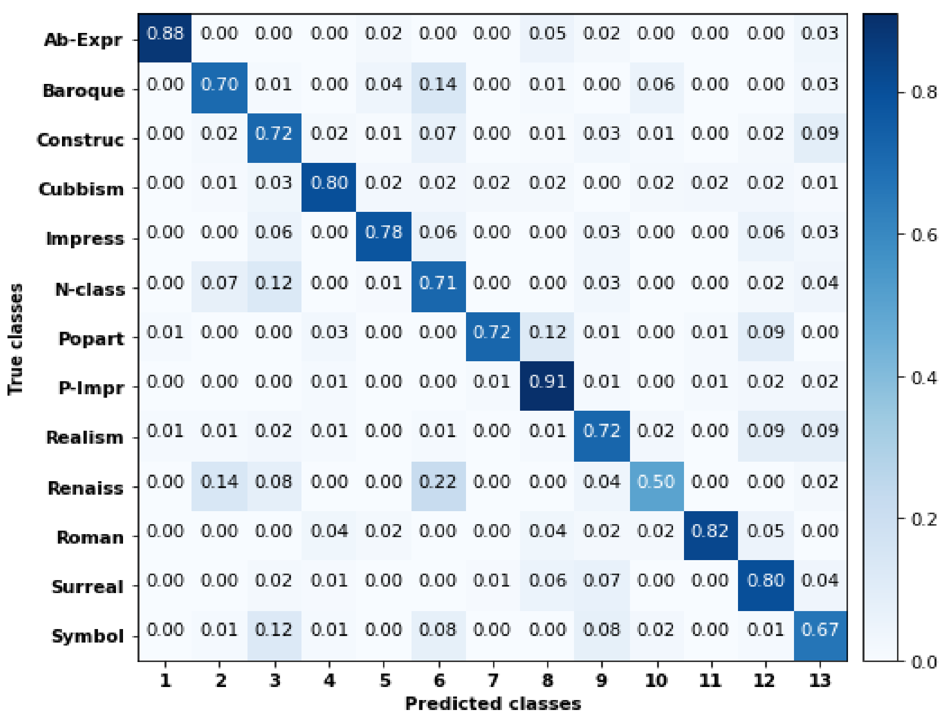


Figure 4.12: Confusion matrix of re-trained custom EfficientNet B6.

4.6 Conclusion

In this chapter, we presented our first contribution to the subject of recognizing the artistic style of a fire art painting. It consisted of three methodologies: Pre-trained EfficientNet models, custom pre-trained EfficientNet models, and deep re-training of the custom pre-trained EfficientNet models.

The results of the evaluation on the Painting-91 dataset showed that the style classification of the custom pre-trained EfficientNet models performed better. They achieved higher accuracy than the results of the pre-trained EfficientNet models in all cases from EfficientNet B0 to EfficientNet B6.

Furthermore, applying the deep retraining technique to the last layers of the custom pre-trained EfficientNet models yielded further improvements in performance. This technique enhanced the capabilities of the models, leading to even better classification results.

The classification results also showed that EfficientNet B6, with the deep retraining technique applied to the last layers of the custom pre-trained model, achieved the highest level of performance. It outperformed all other models within the EfficientNet B0 to B5 range, regardless of whether they were pre-trained models, custom pre-trained models, or deep re-trained custom pre-trained models.

The deep retraining of the custom EfficientNet B2 achieved higher accuracy than the custom EfficientNet B3 and EfficientNet B4 models, even though the two models have higher input resolution and model size. This finding shows that the effectiveness of deep retraining in improving network performance surpasses the impact of increasing the network's resolution size.

These findings highlighted the significant impact of the resolution of the input size, model complexity and deep retraining on the style classification accuracy, as the deep retraining of the deeper and more complex architectures consistently performs better.

In the next chapter, we present our second contribution to style classification, which focuses on studying the effect of different optimizers on the performance of various pre-trained CNN architectures.

Chapter 5

The Effect of Optimizers on CNN Architectures for Art Style Classification

5.1 Introduction

This chapter presents the description of our second research contribution to the field of recognizing the artistic style of fine art paintings and its experimental results.

The second contribution of our research involves a comprehensive study on the influence of optimizers on pre-trained CNN architectures. We compare different CNN architectures specifically designed for style classification and analyze how various optimization algorithms affect their performance. This analysis enables us to identify the most suitable CNN architecture for achieving accurate and robust style recognition. We used in our experiments two art classification datasets, Pandora18k and Painting-91.

This chapter is structured into six sections, starting with an introduction and summary of related studies in Section 5.2. Section 5.3 presents the detailed methodology of the proposed approach, while Section 5.4 covers the experimental validation, including dataset information, data preprocessing, CNN architectures, optimization algorithms, and experimental setup details. The results obtained from the experiments are discussed in Section 5.5, and the chapter concludes in Section 5.6 with a summary of the key findings.

The results of this study were published in the peer-reviewed journal "The International Journal of Computing and Digital Systems" (IJCDS) under the title: "The Effect of Optimizers on CNN Architectures for Art Style Classification". The published paper can be accessed at: https://journal.uob.edu.bh/handle/123456789/4747

5.2 Background

Previous studies in the field of art classification have primarily focused on exploring various methodologies and approaches using different CNN models. However, the investigation of different optimizers, which play a crucial role in model performance, has received limited attention in this domain. While optimization algorithms have been extensively studied in other domains, their application to art classification remains relatively unexplored.

For instance, Agarwal et al. (87) conducted experiments to compare the performance of convolutional neural networks using different optimization algorithms on handwritten datasets such as MNIST (88) and CIFAR 10 (89). Their study aimed to verify the impact of different optimizers on the accuracy and convergence of the models.

Similarly, Verma et al. (90) proposed a comparison of two different optimizers implemented on CNN architectures for classifying COVID-19 X-Ray images. By evaluating the performance of the models with different optimization algorithms, they aimed to identify the optimizer that yields superior results in terms of accuracy and generalization.

In our study, we leverage the technique of transfer learning to assess the effectiveness of six pre-trained convolutional neural networks in identifying the artistic style of paintings. By utilizing pre-trained models, we benefit from the knowledge learned from large-scale datasets and adapt it to the style classification task.

Moreover, we thoroughly investigate the impact of various hyperparameters, including optimizers and learning rates, on the performance of each model. By systematically exploring different combinations of hyperparameters, we aim to identify the optimal configuration that yields the best results for each pre-trained model.

This study fills the research gap in the field of art classification by shedding light on the significance of hyperparameter selection, particularly the choice of optimizers and learning rates. Through our comprehensive analysis, we aim to provide valuable insights into the most effective hyperparameter settings for each pre-trained model, enabling researchers and practitioners to achieve superior performance in style classification tasks.

5.3 Proposed methodology

In this work, we aim to concentrate on two points. The first is to propose a framework for the style classification of a fine art painting, which is illustrated in Figure 5.1. Our framework consists of two essential parts: the first is the data pre-processing, and the second is feature extraction with the use of transfer learning and classification.

To leverage transfer learning, we adopted pre-trained ImageNet models as the foundation for our CNN architectures, avoiding the need to train them from scratch. The last fully connected layers of these architectures, originally designed for 1,000 classes, were replaced with two new dense layers. These layers, initialized randomly, had activation functions using Swish (91) with dimensions of 256 and 128, respectively, followed by a softmax layer representing the number of artistic styles in the dataset. We incorporated batch normalization and dropout layers after each layer to prevent overfitting. The output of each model was a probability vector indicating the potential art-style classes corresponding to the artwork image.

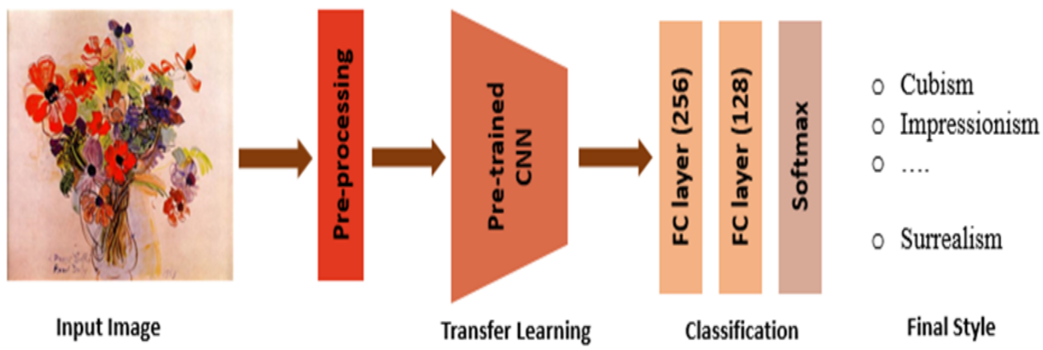


Figure 5.1: The proposed framework for style recognition

The second point is to study and compare the effect of different optimizers with various learning rates (1e-2 and 1e-4) on the performances of six pre-trained CNN architectures on the ImageNet dataset (47), which has 1.2 million natural images and 1000 classes.

5.4 Exprimental validation

In this section, we present the used datasets, the data preprocessing steps, the used CNN architecture, the optimizers and the training setup used in our evaluation.

5.4.1 Exprimental datasets

In our experiments, we used two standard datasets of fine art paintings collected from free accessible fine-art paintings collections.

5.4.1.1 Dataset 1: Panting-91

The Painting-91 dataset (27) consists of a total of 2,338 paintings categorized according to one of 13 different artistic styles, namely: cubism, abstract expressionism, baroque, constructivism, pop art, impressionism, neoclassical, postimpressionism, renaissance, romanticism, realism, symbolism, and surrealism. 1250 of them were utilized for training, while 1088 of them were used for testing. Figure 4.4 shows the percentage of different styles in the Painting-91 dataset.

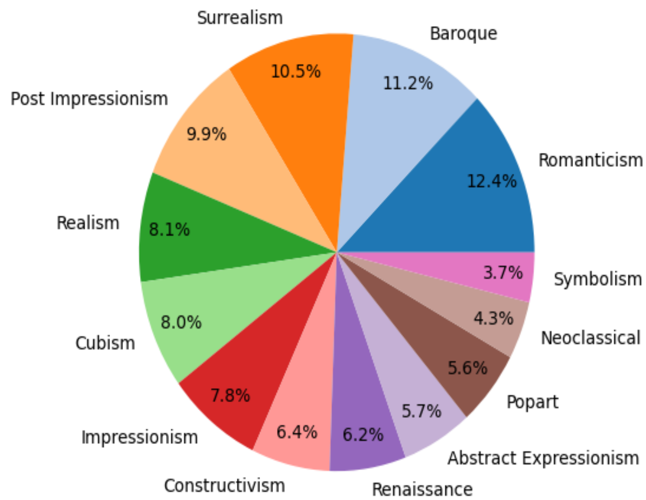


Figure 5.2: Percentage of different styles in the Painting-91 dataset

5.4.1.2 Dataset 2: Pandora18k

The second dataset used in this study was Pandora18k, the Painting Database for the Art Movement Recognition (30; 75). It is one of the most high-quality datasets available for fine art classification tasks, and it consists of 18,038 images of paintings representing 18 distinct artistic styles. Figure 5.3 illustrates the distribution of the dataset images across styles, indicating a relatively balanced distribution with minimal class imbalance.

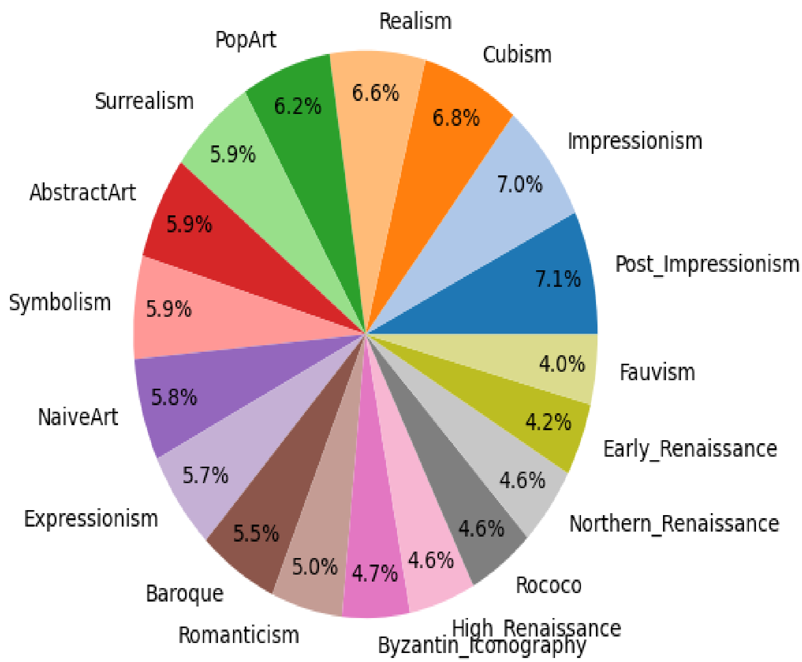


Figure 5.3: Percentage of different styles in the Pandora18k dataset

5.4.2 Data pre-processing:

Before training the models on the Pandora18k and Painting-91 datasets, we performed several preprocessing steps to ensure consistent and reliable results. One of the primary steps was resizing all the images in both the training and test sets to a standardized resolution of 480x480 pixels. This resizing helped to maintain uniformity and facilitate the training process.

Additionally, we applied normalization to the images, which involves adjusting the pixel values to a common scale. Normalization aids in reducing the impact of variations in pixel intensity and ensures that the models are not biased towards specific image characteristics.

To further enhance the diversity and robustness of the training data, we employed data augmentation techniques. These techniques introduce controlled variations to the existing images, effectively expanding the training set. Some of the augmentation techniques utilized include horizontal flipping, where images are mirrored horizontally, and random shifting of width and height. This shifting involves slight translations of the image to different positions, enabling the model to learn from different perspectives. Furthermore, we incorporated image rotation and slight zooming to provide additional variations in the training data.

In order to prevent overfitting, we employed the pre-processing input of each model. This input includes techniques such as dropout or regularization, which help to regularize the model during training and prevent it from memorizing the training data excessively. By applying these techniques, we aim to enhance the model's generalization capabilities and its ability to perform well on unseen data.

Figure 5.4 presents samples of the data augmentation techniques applied to a single image. This visual representation provides a glimpse into how the augmented data differs from the original image, highlighting the variations introduced during the preprocessing stage. These augmented images contribute to a more diverse and representative training dataset, enabling the models to learn and generalize better.

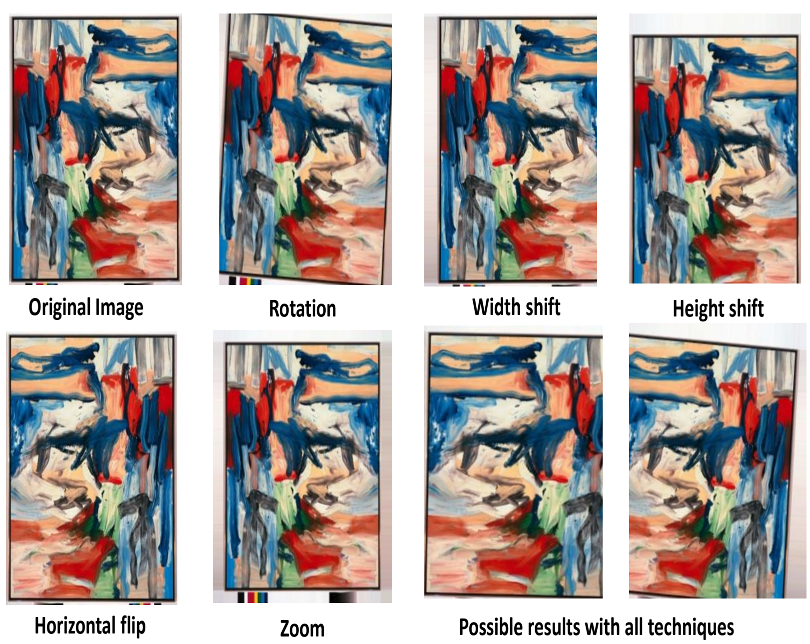


Figure 5.4: Data pre-processing

5.4.3 Convolutional neural network architectures

In this study, we chose six widely used CNN architectures, namely Xception (38), ResNet50 (36), InceptionV3 (35), InceptionResNetV2 (39), DenseNet121 (40), and EfficientNet B3 (42). These architectures have been shown to be powerful and have achieved state-of-the-art performance in various image classification tasks.
Table 5.1 presents the most important characteristics of each CNN architecture in terms of the input size, depth, the size of the model, and the number of parameters. InceptionResNetV2 is the largest and deepest model we tested in our study.

Table 5.1:: The characteristics of CNN architectures

Model	**Input Image Size**	**Depth**	**Size (MB)**	**Parameters (Millions)**
Xception (38)	299 x 299 x 3	81	88	22.9
ResNet-50 (36)	224 x 224 x 3	50	96	25.6
InceptionV3 (35)	229 x 229 x 3	48	89	23.9
InceptionResNetV2 (39)	229 x 229 x 3	164	213.41	56
DenseNet121 (40)	224 x 224 x 3	121	33	7,6
EfficientNet B3 (42)	300 x 300 x 3	210	48	12.3

5.4.3.1 InceptionV3

InceptionV3 is a CNN architecture that is part of the Inception family of models developed by Google researchers (35). The key innovation of the Inception architecture is the use of the "Inception module" which consists of multiple filters of different sizes (1x1, 3x3, and 5x5) in parallel to capture different levels of detail in the input image. InceptionV3 uses a combination of convolutional layers, pooling layers, and fully connected layers to classify images. Figure 5.5 presents the architecture of InceptionV3.

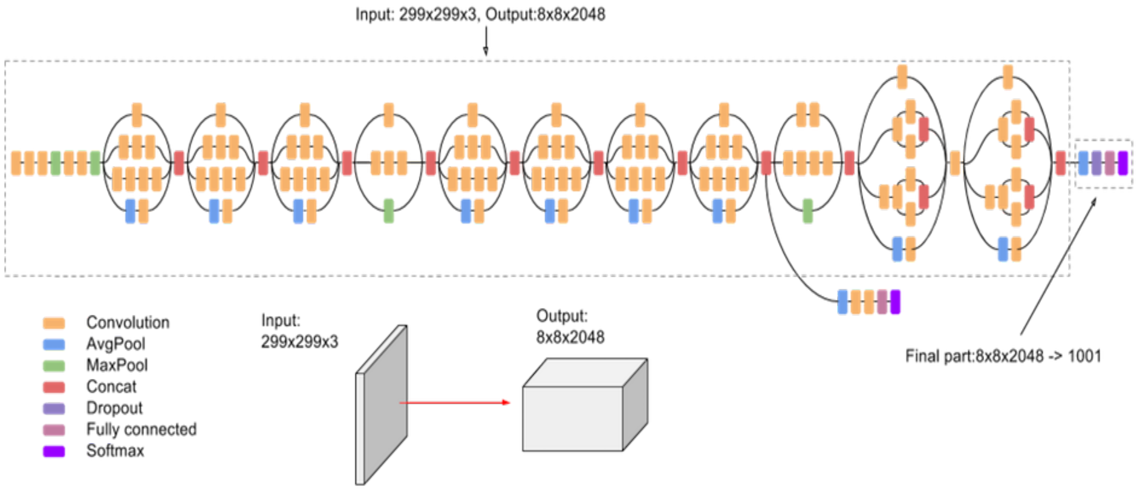


Figure 5.5: Architecture of InceptionV3

5.4.3.2 Xception

Xception is a CNN architecture that is based on the depthwise separable convolution operation. It was introduced in 2016 by François Chollet to improve the Inception architecture's computational efficiency(38). It achieved this by replacing the Inception module with a "depth-wise separable convolution" module, which factorizes each convolutional layer into two separate operations, depth-wise convolution, and point-wise convolution (Figure 5.6) which can provide the same level of accuracy while requiring fewer computations.

Figure 5.7 presents the Xception architecture where the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. The architecture consists of a series of blocks, each of which contains several convolutional layers. The first block contains a 3x3 convolutional layer followed by a batch normalization layer and a ReLU activation function. This block is then followed by a series of depthwise separable convolutional layers. The output of these layers is then passed through another block that combines the output of the previous block with a residual connection.

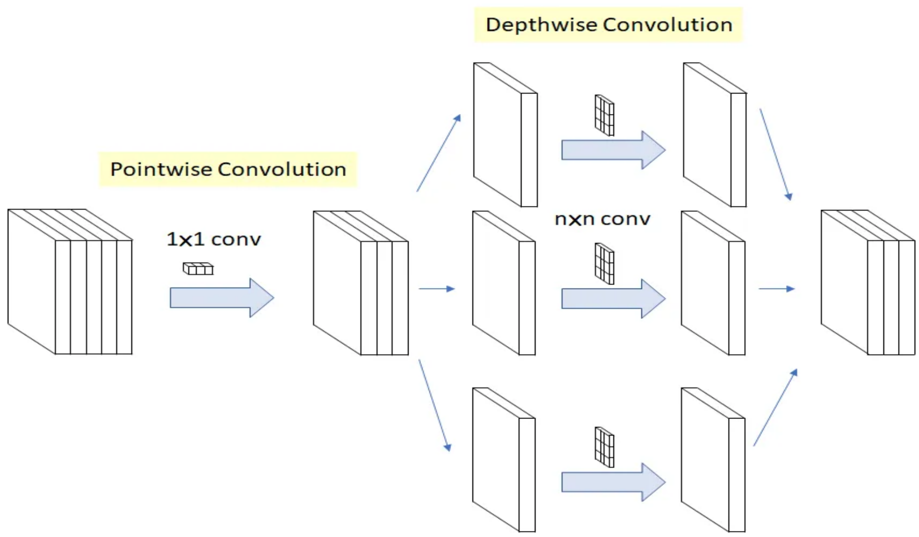


Figure 5.6: The depth-wise and point-wise convolution modules in the architecture of Xception

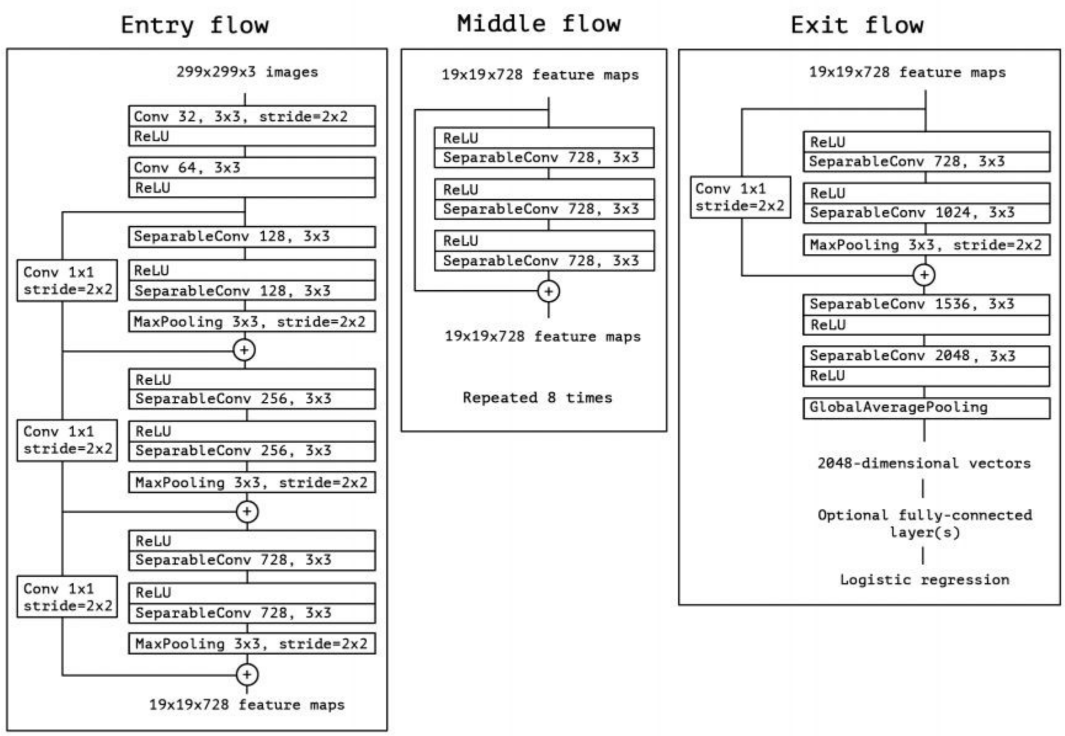


Figure 5.7: Architecture of Xception

5.4.3.3 ResNet50

ResNet50 is a variant of the ResNet (Residual Network) architecture (36), which is composed of 50 layers and it was introduced to address the vanishing gradient problem in deep neural networks. The key innovation of ResNet is the introduction of "residual connections" which is presented in Figure 5.8. It enables the network to learn residual functions instead of directly learning the desired mapping. The residual connections enable the gradient to flow more easily through the network, which helps to improve the accuracy of the model.

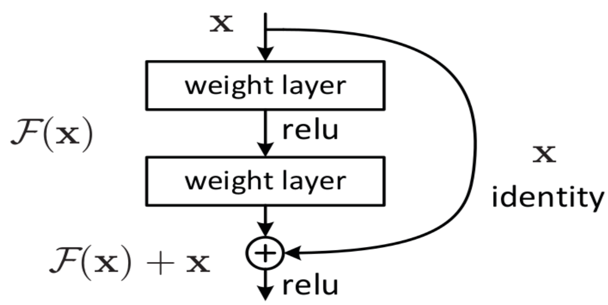


Figure 5.8: Residual connections

The architecture of ResNet50 is illustrated in Figure 5.9,it consists of five stages, each containing multiple residual blocks. Each residual block contains two or three convolutional layers, as well as batch normalization and ReLU activation functions (78). The architecture also uses max pooling and average pooling layers to down-sample the feature maps. At the end of the last stage, there is a global average pooling layer that averages the feature maps across the spatial dimensions, followed by a fully connected layer with a softmax activation function that produces the final classification output.

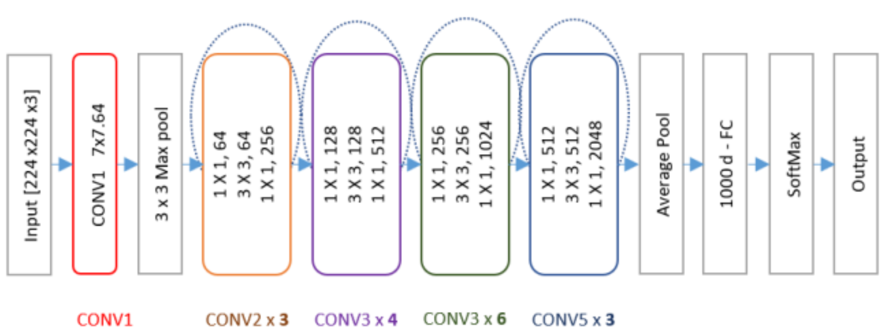


Figure 5.9: Architecture of Resnet-50

5.4.3.4 InceptionResNetV2

InceptionResNetV2 is a hybrid of InceptionV3 and ResNet architectures that combines the benefits of both architectures; it was introduced in 2017 by Szegedy et al. (39) as an extension of the Inception family of models. InceptionResNetV2 improves upon InceptionV3 by adding residual connections, which help alleviate the problem of vanishing gradients during training. InceptionResNetV2 includes residual connections and factorized convolutions, which reduces the number of parameters and computational complexity.

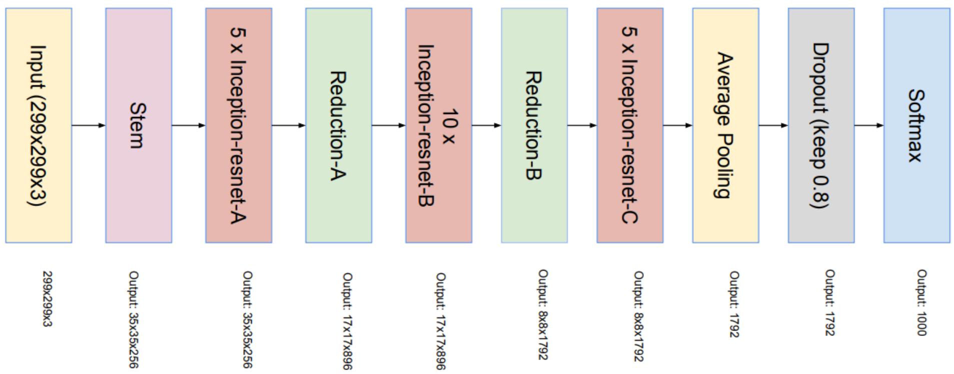


Figure 5.10: Architecture of InceptionResNetV2

Figure 5.10 is the InceptionResNetV2 architectural details, while Figures 5.11, 5.12 and 5.13 present the Stem of the architecture, the Inception modules A, B, C and the Reduction blocks A and B respectively.

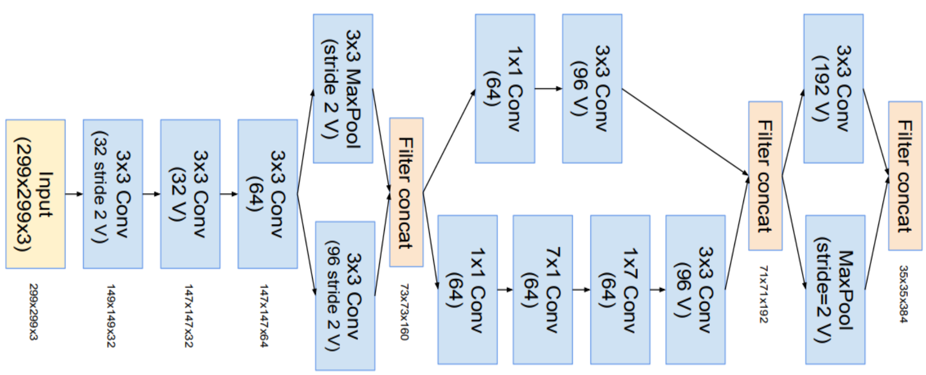


Figure 5.11: Stem of InceptionResNetV2

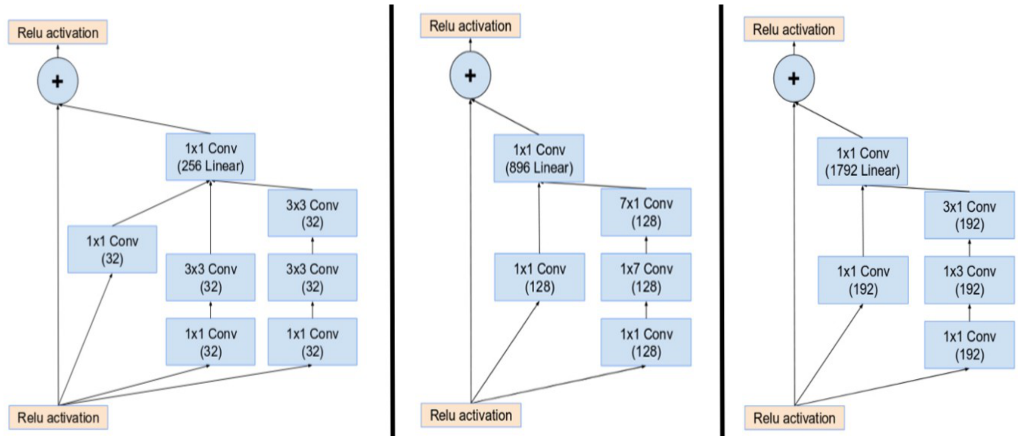


Figure 5.12: Inception modules A, B, C of InceptionResNetV2

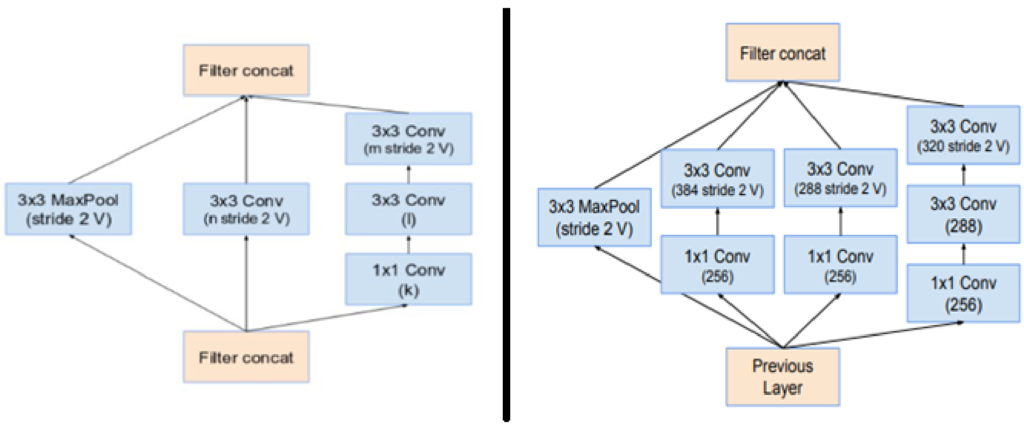


Figure 5.13: Reduction block A and B of InceptionResNetV2

5.4.3.5 DenseNet121

DenseNet121 is a CNN architecture that was introduced in 2016 by Huang et al. (40). It is a variant of the DenseNet family of neural networks, which aims to address the vanishing gradient problem in deep neural networks by introducing dense connectivity between layers. DenseNet121 is designed to have 121 layers, and it is composed of several dense blocks, where each dense block is made up of multiple convolutional layers that are densely connected. Each dense block takes as input the feature maps from all previous dense blocks and produces output feature maps that are fed into the next dense block. The dense connectivity results in a significant reduction in the number of parameters required to train the network, while also improving its accuracy. This architecture encourages feature reuse and enables the training of very deep neural networks with fewer parameters. Figure 5.14 presents the architecture of DenseNet with 5 blocks, while Figure 5.15 presents the architecture of DenseNet121.

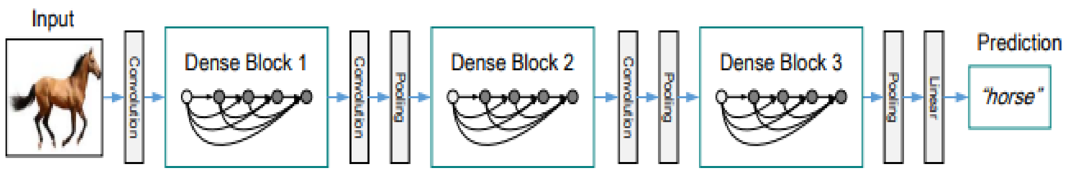


Figure 5.14: Architecture of DenseNet with 5 blocks

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 5.15: Architecture of DenseNet121

5.4.3.6 EfficientNet B3

EfficientNet B3 is a CNN architecture that belongs to the EfficientNet models family (42), which are designed to be efficient in terms of both computation and memory. It uses a compound scaling method to optimize the network's depth, width, and resolution simultaneously. EfficientNet B3 layers use a combination of depth-wise separable convolutions, which separate the spatial and channel-wise convolutions, and inverted residual blocks, which use shortcut connections to reduce the number of computations. Figure 5.16 (92) presents the architecture of EfficientNet B3.

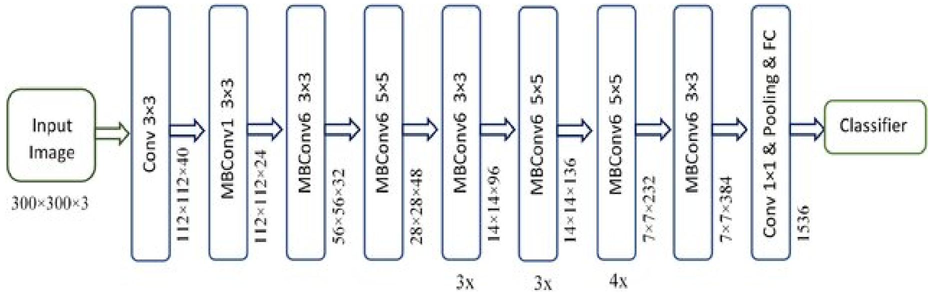


Figure 5.16: Architecture of EfficientNet B3

5.4.4 Optimizers

In deep learning, an optimizer is an algorithm used to adjust the weights and biases of the network during training in order to minimize the loss function, which measures the difference between the predicted outputs of the network and the true outputs. In other words, the loss function measures how well the network is performing on the task it is being trained to solve. The goal of the optimizer is to find the values of the network's parameters that minimize the loss function of the neural network.

Optimizers work by computing the gradients of the loss function with respect to the parameters of the network and then updating the parameters in a way that reduces the loss. Different optimizers use different methods to compute the gradients and update the parameters. Figure 5.17 illustrates the details of our training process showing the optimizer.

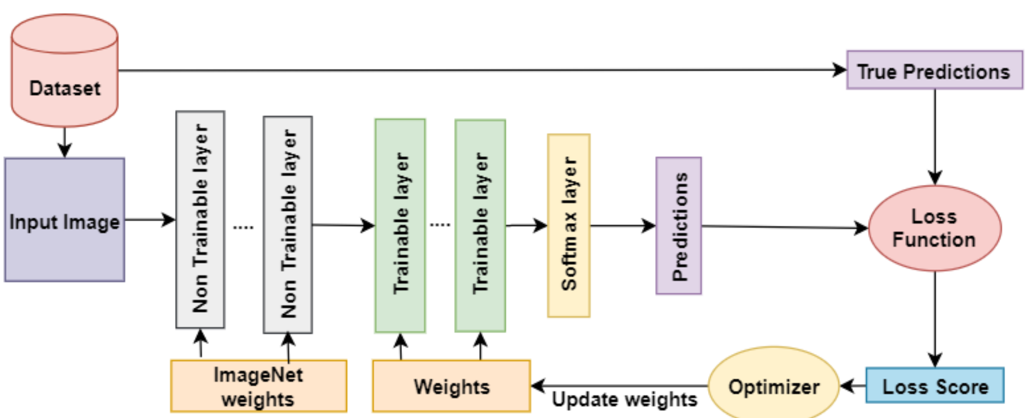


Figure 5.17: A schematic illustration of the training process

In this study, we used there different optimizers are:

5.4.4.1 SGD optimizer

SGD (Stochastic Gradient Descent) (93) is a widely used optimization algorithm in deep learning for minimizing the cost function or loss function. It is a type of gradient descent algorithm that updates the weights of a neural network after each batch of training examples. it works by calculating the gradient of the cost function with respect to the neural network weights for a mini-batch of training examples. It then updates the weights in the opposite direction of the gradient to minimize the cost function. The equation for the weight update using SGD optimizer can be written as:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t, \mathbf{x}_t, y_t)$$

where:

θ_t is the parameter at time t
η is the learning rate
$\nabla J(\theta_t, \mathbf{x}_t, y_t)$ is the gradient of the loss function J with respect to the parameters θ_t, evaluated on the training example $(\mathbf{x}_t, y_t)$.

5.4.4.2 RMSprop optimizer

RMSprop (Root Mean Square Propagation) (94) is an adaptive learning rate optimization algorithm that calculates an exponential weighted moving average of the squared gradient for each weight and divides the gradient by the root mean square of the exponential moving average. The algorithm uses this normalized gradient to update the weights. RMSprop also introduces a decay factor that reduces the influence of past gradients over time. This helps prevent the optimizer from getting stuck in local minima or diverging. It modifies the learning rate of the stochastic gradient descent (SGD) algorithm to improve its convergence speed and stability. The equation for the weight update using the RMSprop optimizer can be written as:

$$g_t = \nabla_\theta J(\theta_{t-1})$$
$$E[g^2]t = \alpha E[g^2]t - 1 + (1 - \alpha) g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

where:

g_t is the gradient of the loss function J with respect to the parameters θ at time t
$E[g^2]_t$ is the exponentially decaying average of squared gradients up to time t

α is the decay rate for the moving average (typically set to 0.9)
η is the learning rate
ϵ is a small constant added for numerical stability

5.4.4.3 Adam optimizer

Adam (Adaptive Moment Estimation) (85) an extension of the stochastic gradient descent with the momentum algorithm, which combines the gradient descent update rule with a momentum term to accelerate convergence. It takes into account the first and second moments of the gradients to adaptively adjust the learning rate during training. Specifically, it computes an estimate of the mean and variance of the gradient, which are then used to update the parameters in each iteration.
the equation for the weight update using the Adam optimizer can be written as:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon} \hat{m}_t
\end{aligned}
$$

where:

m_t and v_t are the first and second moments estimates, respectively, at time t
$\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments, respectively
β_1 and β_2 are the exponential decay rates for the first and second moments, respectively g_t is the gradient at time t
θ_t is the parameter at time t
η is the learning rate
ϵ is a small constant added for numerical stability.

5.4.4.4 Advantages and limitations of SGD, RMSprop, and Adam optimizers

The following table compares the advantages and limitations of SGD, RMSprop, and Adam optimizers:

Table 5.2:: Advantages and limitations of SGD, RMSprop, and Adam optimizers

Optimizer	Advantages	Limitations
SGD (93)	- Simplicity - computational efficiency - ease of implementation	- It can easily get stuck in local minima - takes a long time to converge to the global minimum
RMSprop (94)	-Robust to noisy data -Automatically adapts the learn ing rate - requires only a small amount of memory	-May converge to a suboptimal solution if the learning rate is too high
Adam (85)	-Robust to noisy data -Computes adaptive learning rates for each parameter - Efficiently combines the advan tages of RMSprop and momentum	- Several hyper-parameters that need to be tuned - Can converge to a suboptimal solution - Computationally expensive compared to other optimizers

5.4.5 Exprimental setup

During training, we performed fine-tuning by unfreezing the last four layers of each model and re-training them in addition to the last fully-connected layers. After 40 iterations (epochs) of training with a batch size of 64, we considered the maximum accuracy achieved as the final result. Our experiments were conducted using Tensorflow 2.3.0 (95) on a system running Windows 11 with a Geforce GTX 1660 Super and an Intel i9 10900k processor. The pre-trained models utilized in this study were obtained from the Keras library (96).

5.5 Results and discussion

The evaluation metric used in our experiments was accuracy, which is defined as the percentage of successfully identified examples relative to the total number of examples. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Where true positive and true negative classification predictions are denoted by TP and TN, respectively, while false positive and false negative classification predictions are denoted by FP and FN, respectively (77).

5.5.1 Results of style classification on the Painting-91 dataset

Table 5.3 presents the results of style classification on the Painting-91 dataset using different models and optimizers. The models include Xception, Resnet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNetB3. The optimizers used are SGD, RMSprop, and Adam, with varying learning rates of 1e-2 and 1e-4. Based on the results presented in the table, several observations can be made regarding the performance of different pre-trained CNN architectures on the Painting-91 dataset.

Table 5.3:: The results of style classification on the Painting-91 dataset.

Model \ **Optimizer**	**SGD**		**RMSprop**		**Adam**	
	1e-2	**1e-4**	**1e-2**	**1e-4**	**1e-2**	**1e-4**
Xception	69.67	18.75	67.10	71.51	69.85	71.32
Resnet50	72.15	22.43	72.15	73.16	73.07	72.24
InceptionV3	69.58	19.12	71.69	68.20	70.96	68.66
InceptionResNetV2	**73.25**	**24.36**	72.06	**75.00**	72.89	**75.18**
DenseNet121	69.29	15.44	**73.99**	67.46	**73.71**	65.44
EfficientNetB3	68.84	13.24	71.42	70.40	71.78	69.21

The InceptionResNetV2 model consistently outperformed the other tested pre-trained models when using the SGD optimizer with both learning rates (1e-2 and 1e-4). The model achieved the highest accuracy of 73.25% with a learning rate of 1e-2, which is a significant improvement compared to the accuracy of 24.26% obtained with a smaller learning rate of 1e-4.

When using the RMSprop optimizer with a learning rate of 1e-2, the InceptionResNetV2 model achieved the third-highest accuracy of 72.06%, following ResNet50 and DenseNet121, which achieved accuracies of 72.15% and 73.99% respectively. Interestingly, when using the RMSprop optimizer with a lower learning rate of 1e-4, the accuracy of InceptionResNetV2 increased to 75.00%, becoming the highest among all tested models. In contrast, the accuracy of DenseNet121 decreased by 6.53% to reach 67.46%.

Similarly, when using the Adam optimizer with a learning rate of 1e-2, the InceptionResNetV2 model achieved the third-highest accuracy of 72.89%, following ResNet50 and DenseNet121, which achieved accuracies of 73.07% and 73.99% respectively.

Furthermore, when using the Adam optimizer with a smaller learning rate of 1e-4, the accuracy of InceptionResNetV2 increased slightly to 75.18%, once again becoming the highest among all tested models. However, the accuracy of DenseNet121 decreased by 8.5% to reach 65.21%.

From these results, it can be concluded that the choice of optimizer and learning rate significantly impact the performance of each pre-trained model. The InceptionResNetV2 model achieved the best results with the SGD optimizer and a larger learning rate (1e-2). However, for other optimizers such as RMSprop and Adam, a smaller learning rate (1e-4) yielded the highest accuracy for InceptionResNetV2. Therefore, selecting an appropriate optimizer and fine-tuning the learning rate are crucial steps to achieve optimal performance with pre-trained models on the Painting-91 dataset.

From the previous results, we can conclude that the best optimizer for each pre-trained model differs from one to another. Additionally, it is crucial to choose an adequate learning rate as the model may fail to achieve good results if an inadequate learning rate is used.

5.5.2 Results of style classification on the Pandora 18k dataset

The results presented in table 5.4, which showcases the outcomes of our experiments conducted on the larger Pandora18k dataset, exhibit similarities with those obtained from the smaller Painting-91 dataset. This observation leads us to the conclusion that the size of the dataset does not significantly impact the performance of pre-trained models. Remarkably, the best hyperparameters for a pre-trained model remain consistent irrespective of the dataset size. In our investigation, the pre-trained Xception model, introduced for the first time in the context of style recognition, achieved an impressive accuracy of 65.49% on the Pandora18k dataset. This performance surpassed that of the InceptionV3 model, which achieved an accuracy of 61.73%. These results indicate that the Xception model is a promising choice for style recognition tasks.

The InceptionResNetV2 model achieved the highest accuracy in most cases, regard-

Table 5.4:: The results of style classification on the Pandora 18k dataset.

Model \ Optimizer	SGD		RMSprop		Adam	
	1e-2	**1e-4**	**1e-2**	**1e-4**	**1e-2**	**1e-4**
Xception	62.94	33.49	62.44	65.02	62.72	65.49
Resnet50	66.79	39.57	66.90	66.92	66.65	67.56
InceptionV3	59.02	31.17	61.73	60.04	61.53	59.60
InceptionResNetV2	**67.56**	**40.21**	66.79	**68.36**	66.79	**68.45**
DenseNet121	64.19	32.41	**68.03**	66.45	**68.07**	66.34
EfficientNetB3	62.89	28.46	63.91	64.66	63.16	65.18

less of the optimizer used. With the SGD optimizer and a learning rate of 1e-2, the InceptionResNetV2 model achieved an accuracy of 67.56%, surpassing all other models. Similarly, with the RMSprop optimizer and a learning rate of 1e-4, the InceptionResNetV2 model achieved an accuracy of 68.36%, again outperforming the other models. With the Adam optimizer and a learning rate of 1e-4, the InceptionResNetV2 model achieved an accuracy of 68.45%, once again being the top-performing model. The ResNet50 model also performed relatively well across different optimizers and learning rates. It achieved the second-highest accuracy in most cases. With the Adam optimizer and a learning rate 1e-4, the ResNet50 model achieved the highest accuracy of 67.56%.

The DenseNet121 model achieved the highest accuracy in two cases: with the RMSprop optimizer and a learning rate of 1e-2, and with the Adam optimizer and a learning rate of 1e-2. In both cases, it achieved an accuracy of 68.03The Xception, InceptionV3, and EfficientNetB3 models generally achieved lower accuracies compared to the InceptionResNetV2, ResNet50, and DenseNet121 models. The InceptionResNetV2 model consistently performed well, achieving the highest or second-highest accuracies across different optimizers and learning rates. The ResNet50 and DenseNet121 models also demonstrated competitive performance.

The findings from our experiments highlight the significance of selecting appropriate pre-trained models and optimizing hyperparameters. The consistent superiority of the InceptionResNetV2 model reinforces its suitability for style recognition tasks, while the Xception model presents a viable alternative with its commendable performance. Furthermore, the consistency of optimal hyperparameters across different dataset sizes emphasizes the robustness and generalizability of these models.

5.5.3 Analysis of classification results

Figures 5.18, 5.19, and 5.20 provide a visual representation of the results obtained from our experiments, focusing on the performance of the six pre-trained models for style classification using different optimizers: SGD, RMSprop, and Adam. Each figure comprises two subplots, with the top subplot presenting the results for the Painting-91 dataset and the bottom subplot displaying the results for the Pandora18k dataset. In each subplot, every pre-trained model is represented by two bars: a blue bar indicating the accuracy achieved when trained with a learning rate of 1e-2, and an orange bar representing the accuracy attained with a learning rate of 1e-4.

Upon analyzing the figures, it is evident that the pre-trained Xception model outperformed the pre-trained InceptionV3 model on the Pandora18k dataset. This observation suggests that Xception is more effective in capturing the style characteristics of the Pandora18k dataset.

Additionally, it is worth noting that the pre-trained ResNet50 model exhibited higher accuracy compared to the Xception, InceptionV3, and EfficientNet B3 models on both the Painting-91 and Pandora18k datasets. This finding indicates that ResNet50 excels in extracting and classifying artistic styles, performing consistently well across different dataset sizes.

The visual representation of the results in Figures 5.18, 5.19, and 5.20 provides a clear overview of the performance of each pre-trained model and optimizer combination on the two datasets. These figures serve as a valuable reference for understanding the comparative strengths and weaknesses of the models and optimizers, aiding in the selection of the most suitable approach for style classification tasks.

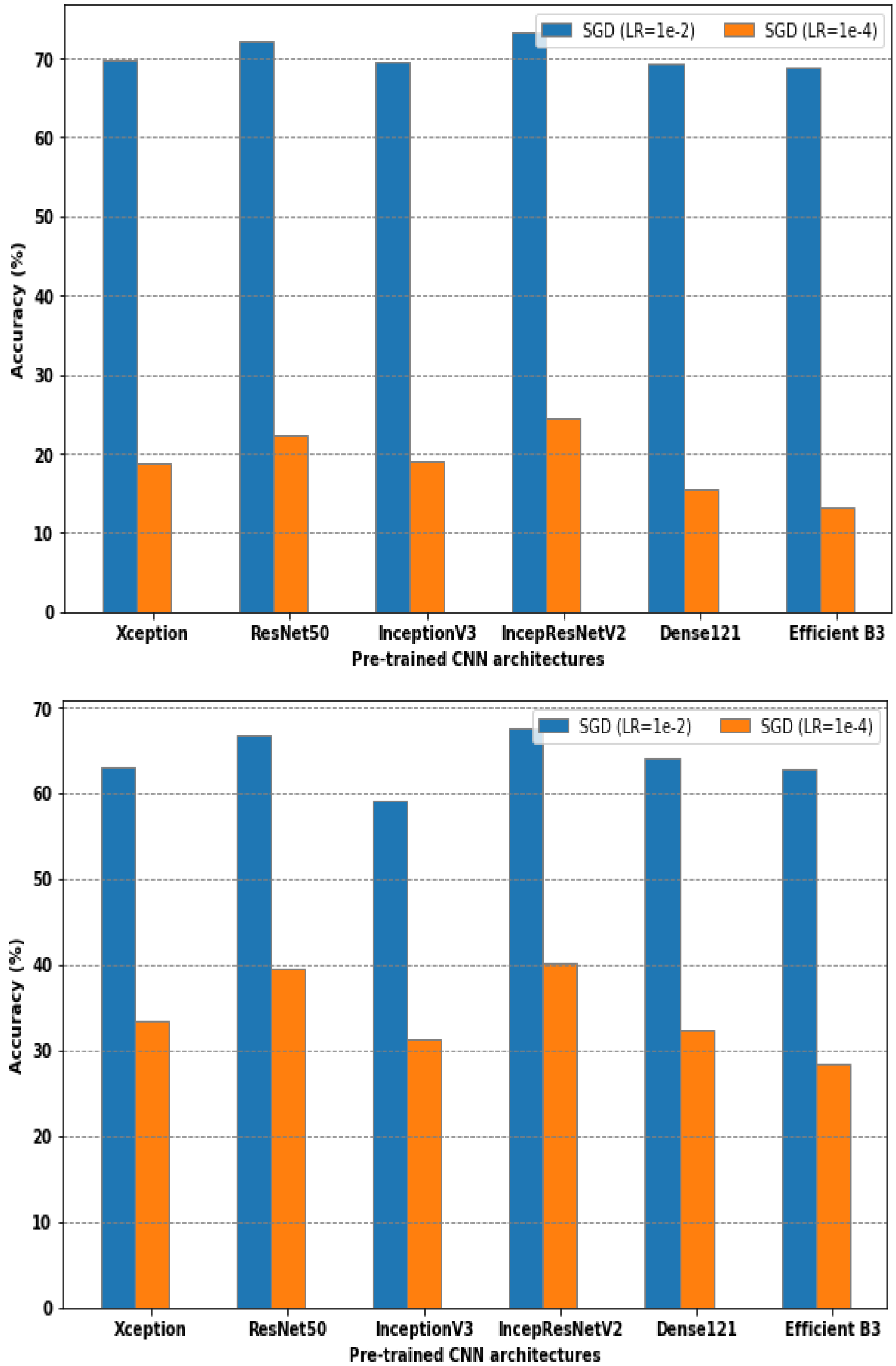


Figure 5.18: The results of style classification with SGD optimizer on the top: Panting-91 dataset and on the bottom: Pandora18k dataset

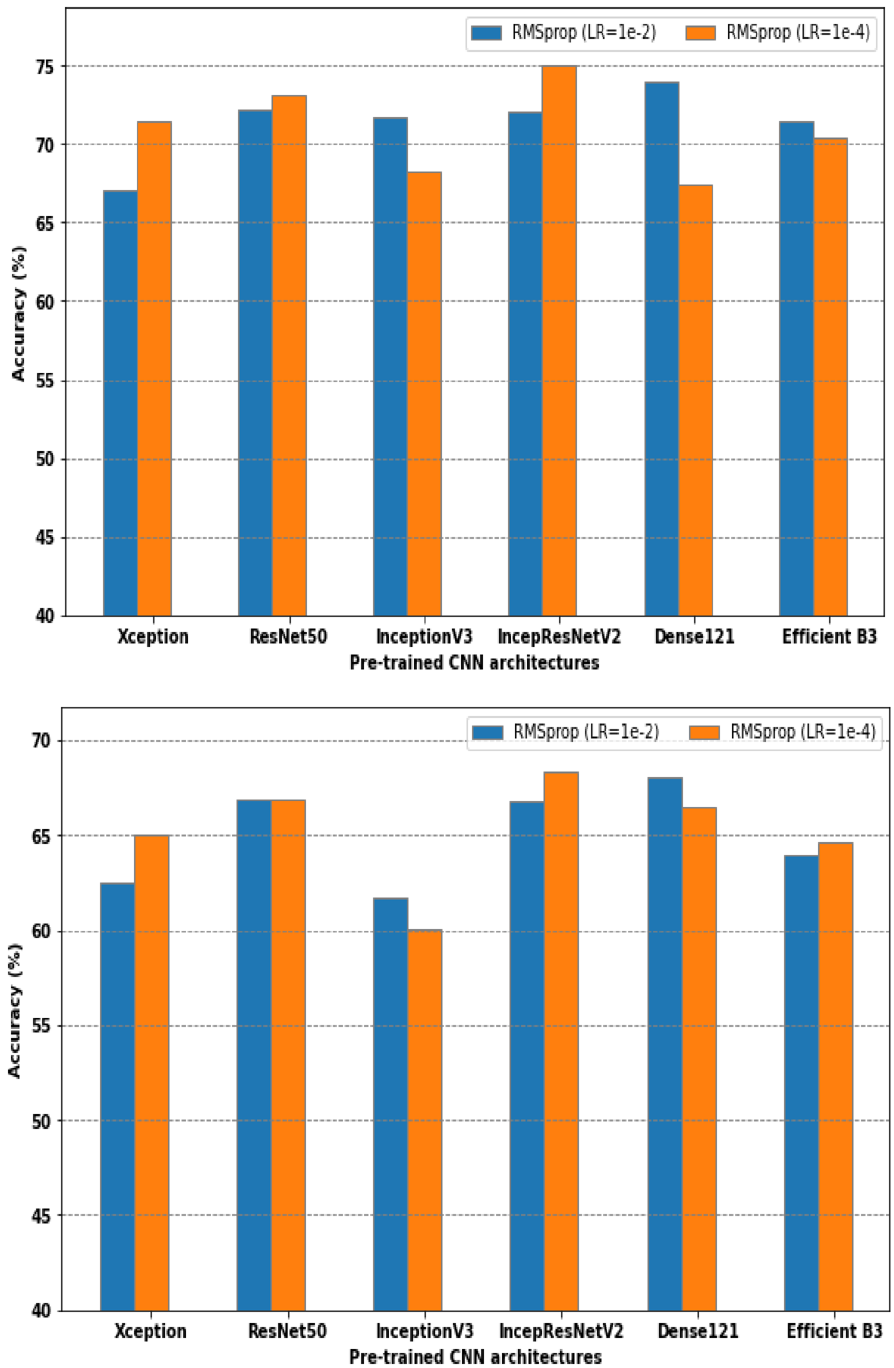


Figure 5.19: The results of style classification with RMSprop optimizer on the top: Panting-91 dataset and on the bottom: Pandora18k dataset

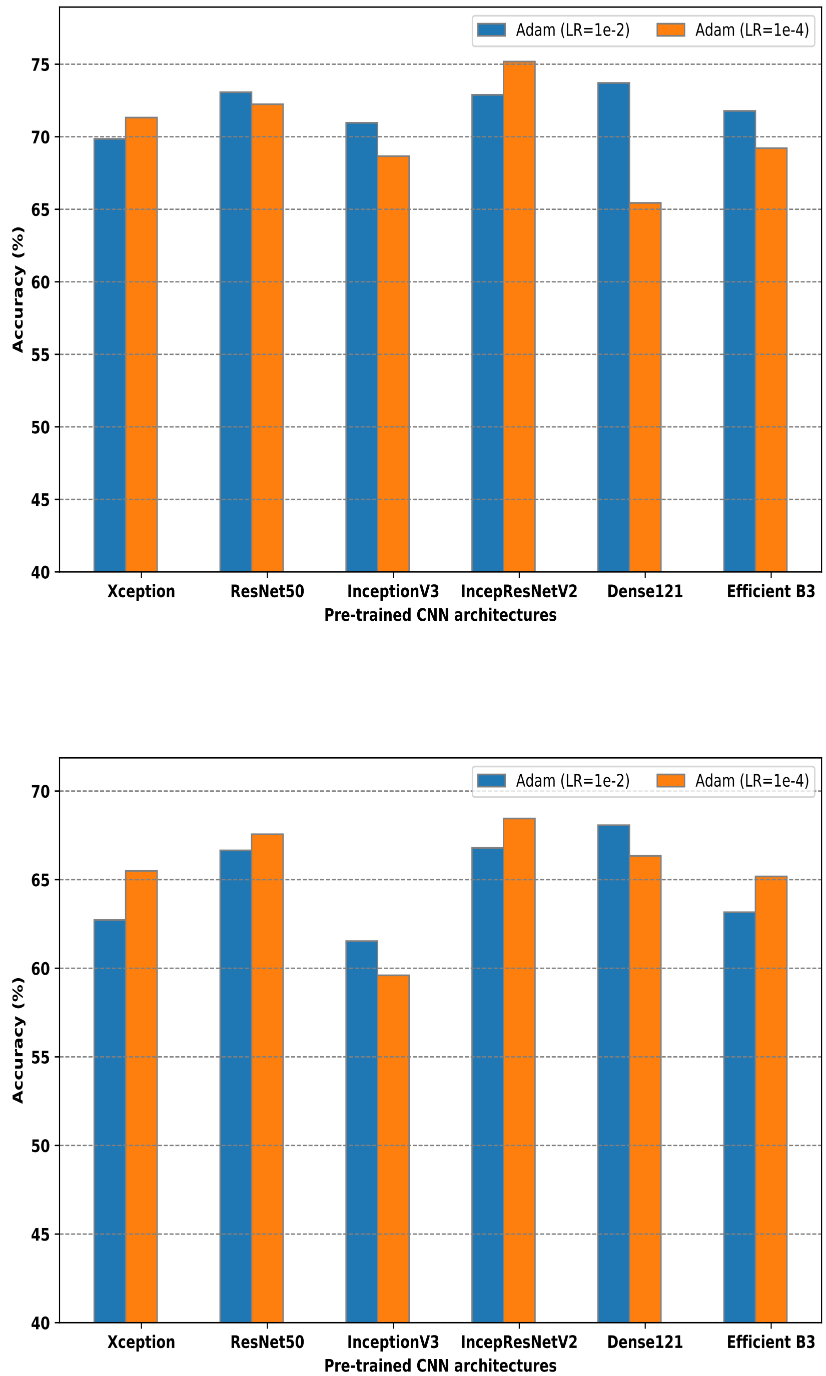


Figure 5.20: The results of style classification with Adam optimizer on the top: Panting-91 dataset and on the bottom: Pandora18k dataset

5.5.4 Confusion Matrices

Figures 5.21 and 5.22 illustrate the confusion matrices for the InceptionResNetV2 model trained with the Adam optimizer and a learning rate of 1e-4 on two different datasets: Dataset 1 (Painting-91) and Dataset 2 (Pandora18k). The diagonal elements of the matrices represent the average accuracy achieved for each individual style.

Figure 5.21, we can see that the Abstract Expressionism, Surrealism, and Cubism styles exhibited the highest accuracy, with recognition rates of 93%, 88%, and 87%, respectively. On the other hand, The paintings belonging to Symbolism were mixed up with other paintings from the Constructivism style. The Renaissance style had the lowest accuracy of 56% and was frequently misclassified, often confused with the Neo-classical style. This suggests that distinguishing between Renaissance and Neo-classical styles proved to be more challenging for the model.

Moving to Figure 5.22 shows that the styles Bayantinizim and Early-Renaissance achieved the highest accuracy, with 97% and 87%, respectively. This indicates that the model successfully recognized these styles with high accuracy. However, Expressionism yielded the lowest accuracy of 37% and exhibited relatively high confusion rates with the Fauvism, and Post-Impressionism styles. Similarly, the Baroque style was often misclassified as Rococo and Romanticism. These instances of confusion can be attributed to the similarities between these styles, as they belong to adjacent periods in art history.

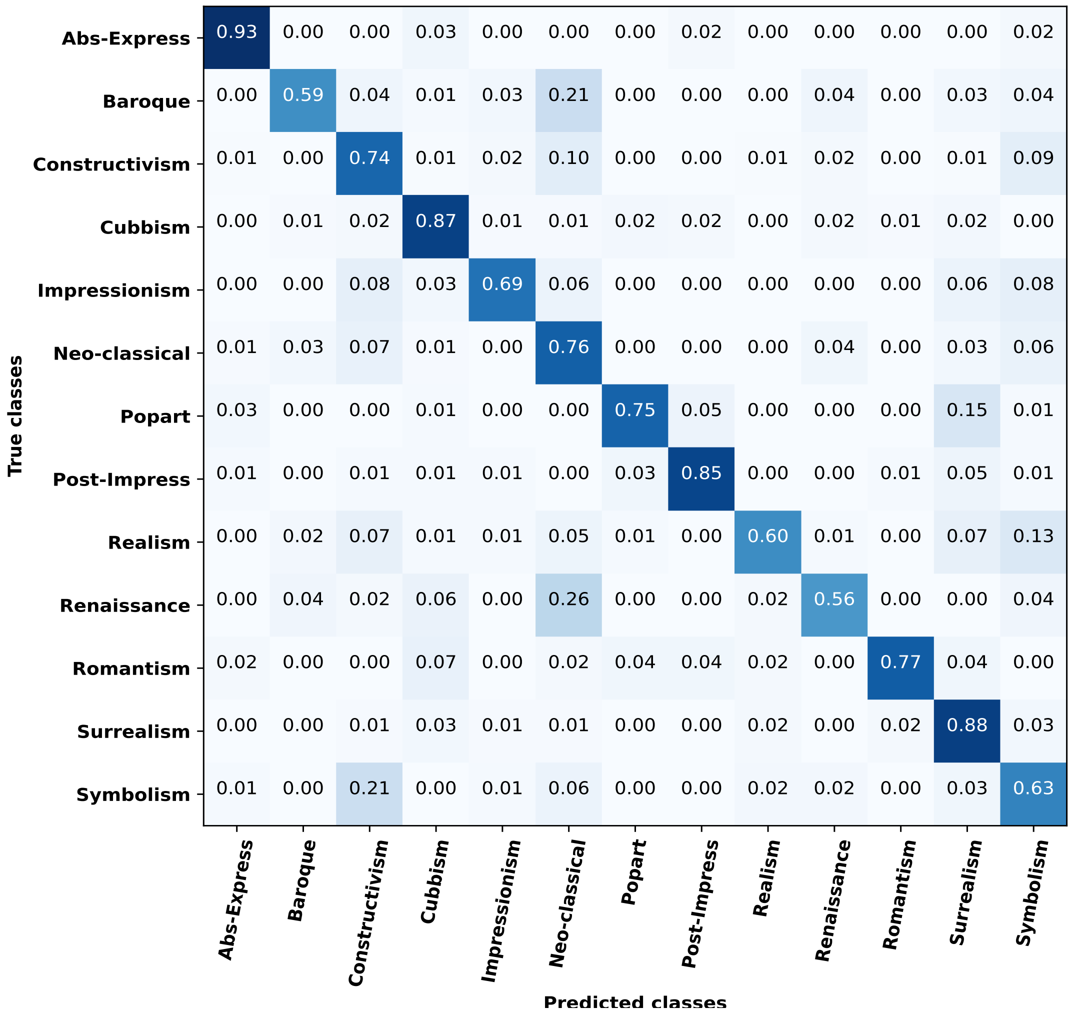


Figure 5.21: Confusion matrix of of InceptionResNetV2 with Adam optimizer and learning rate of 1e-4 on the Painting-91 dataset

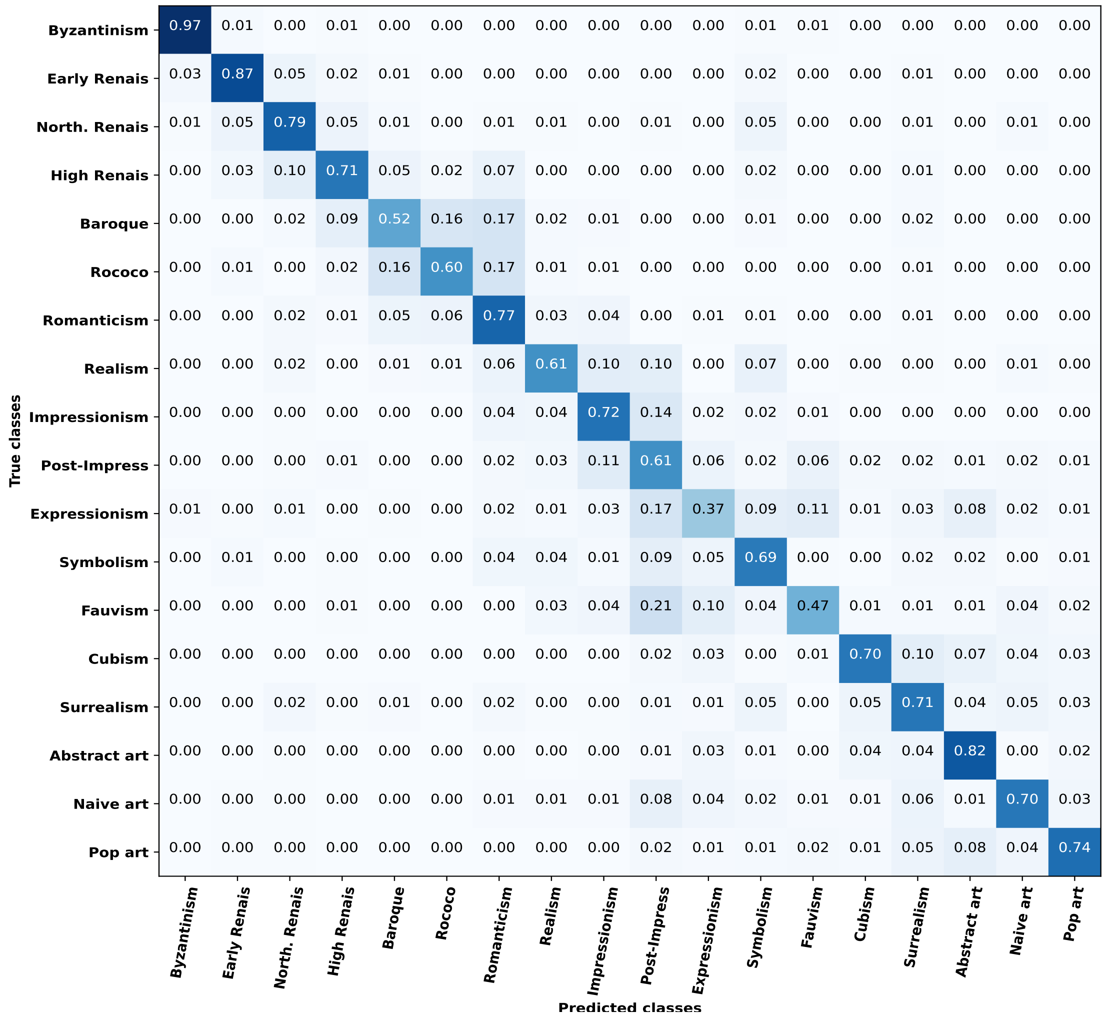


Figure 5.22: Confusion matrix of InceptionResNetV2 with Adam optimizer and learning rate of 1e-4 on the Pandora18k dataset

5.6 Conclusion

This chapter presented our second contribution, focusing on comparing different pre-trained CNN architectures, namely Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3, for style classification tasks, including Xception architecture, which to our knowledge, has never been used for this purpose before. Additionally, we investigated the impact of various optimizers (SGD, RMSprop, and Adam) with different learning rates (1e-2 and 1e-4) on the performance of these architectures.

The results of the evaluation on the Painting-91 dataset and the Pandora18k dataset showed that all the pre-trained models performed poorly with the SGD optimizer and a small learning rate (1e-4). However, significant improvements were observed when using a higher learning rate of 1e-2, indicating the impact of choosing the correct learning rate, as the model may fail to achieve good results with inadequate hyper-parameters.

The results of all the pre-trained CNN models with the RMSprop optimizer and the Adam optimizer show similar results when evaluated with the same learning rate. Both are better than the ones with the SGD optimizer.

Moreover, we found that a small model's best-performing optimizer and learning rate are not always the best hyper-parameters for a more profound and larger model.

The pre-trained Xception model, which was not previously utilized for style classification of fine art paintings, outperformed the pre-trained InceptionV3 model on the Pandora18k dataset. Furthermore, the pre-trained ResNet50 model demonstrated higher accuracy than the pre-trained Xception, InceptionV3, and EfficientNet B3 on both the Painting-91 and Pandora18k datasets. These findings highlight the effectiveness of the pre-trained ResNet50 model in capturing and recognizing the intricate details and stylistic features present in fine art paintings. It also indicates that the choice of the pre-trained model can significantly influence the performance of style recognition tasks.

Among the evaluated models, the pre-trained InceptionResNetV2 architecture demonstrated the highest accuracy for artistic style classification on both datasets when trained with the Adam optimizer and a learning rate of 1e-4.

In conclusion, the research highlights the importance of choosing the appropriate optimizer for CNN architectures in art style classification as the model may fail to achieve good results if an inadequate learning rate is used. Importantly, the best optimizer for each pre-trained model differs from one to another.

Chapter 6

Conclusion and Future Work

In this chapter, we present the thesis's main finding and provide some future work research directions.

6.1 Summary and findings of the thesis

In this thesis, our primary focus was on advancing the field of recognizing fine art paintings' artistic style through applying deep learning approaches. We recognized the potential of supervised learning in this context and specifically explored the use of transfer learning to fine-tune various convolutional neural network (CNN) architectures for style classification.

To contribute to the existing literature, we delved into exploring different CNN architectures that have not been extensively investigated before in the domain of fine art classification. By leveraging the power of transfer learning, we were able to initialize these architectures with pre-trained weights, allowing us to benefit from their learned features and adapt them to the task of style classification.

Although the great performance of EfficientNet models in natural image classification on the ImageNet dataset, these architectures had not previously been used for painting classification. Therefore, Our first contribution focused on investigating the effectiveness of seven pre-trained models from the EfficientNet family, ranging from B0 to B6, allowing for a comprehensive comparison for recognizing the artistic style of a fine art painting. Additionally, we enhanced the base architectures of the EfficientNet models by adding additional layers to create our custom models. Furthermore, we explored the impact of deep retraining the last layers of a pre-trained model on the accuracy of style recognition.

The classification results on the Painting-91 dataset showed that EfficientNet B6, with the deep retraining technique applied to the last layers of the custom pre-trained model, achieved the highest level of performance. It outperformed all other models within the EfficientNet B0 to B5 range, regardless of whether they were pre-trained models, custom pre-trained models, or deep re-trained custom pre-trained models. These findings highlighted the significant impact of the resolution of the input size, model complexity and deep retraining on the style classification accuracy, as the deep retraining of the deeper and more complex architectures consistently performs better.

Furthermore, we investigated how the choice of optimizer and learning rate affects the performance and robustness of the pre-trained convolutional neural network models in the field of fine art paintings.

We compared different pre-trained CNN architectures, namely Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3, for style classification tasks, including Xception architecture, which to our knowledge, has never been used for this purpose before. And we focused on exploring the effectiveness of three popular optimizers: Stochastic Gradient Descent (SGD), Adam, and RMSprop to advance the state-of-the-art of artistic analysis techniques by unraveling the relationships between model architectures, optimizers, and artistic style classification performance..

The evaluation results on the Painting-91 and Pandora18k datasets showed that the size of the dataset does not affect the performance of the pre-trained models, as the best hyper-parameters for a pre-trained model are the same for a small or large dataset.

The study's findings revealed that the choice of optimizer significantly impacts the performance of the pre-trained CNN architectures for art style classification. The results demonstrated variations in accuracy across different optimizers and learning rates, indicating their crucial role in effectively training the models as the model may fail to achieve good results if an inadequate learning rate is used.

In conclusion, this research emphasizes the significance of carefully selecting optimizers and learning rates for pre-trained CNN architectures in art-style classification. It highlights the necessity of conducting thorough experiments and considering the specific characteristics of the models to achieve optimal performance.

6.2 Future work

Further research in style classification is crucial to address the challenges of reducing confusion between specific artistic movements and improving the overall accuracy of classification systems. One potential avenue to explore is the utilization of Generative Adversarial Networks (GANs) for generating synthetic paintings in various artistic styles. Synthetic artworks representing different styles can be generated by training GAN models on a diverse dataset of fine art paintings. These synthesised paintings can serve as valuable training data, augmenting the existing dataset and improving the classification accuracy.

Another promising approach is using an ensemble of different CNN (Convolutional Neural Network) approaches. By combining multiple CNN models, each trained on a specific subset of the data or using different architectural configurations; we can benefit from their collective intelligence to improve the accuracy and robustness of style classification. Ensemble methods have been shown to effectively reduce bias and increase the overall performance of classification tasks.

Additionally, incorporating Long Short-Term Memory (LSTM) networks into the classification pipeline could be explored. LSTM networks can capture temporal dependencies and sequential information, which could be valuable in capturing the evolution and progression of artistic styles over time. A more comprehensive and nuanced understanding of artistic styles can be achieved by combining the visual features extracted by CNNs with the temporal modelling capabilities of LSTM networks.

Furthermore, our research focused on recognizing the style of a painting, but there are exciting opportunities for future research to expand upon our findings. For example, to explore the integration of the style classification approaches into augmented reality (AR) applications. By leveraging AR technology, we can bring the analysis of fine art to a new level of interactivity and engagement.

Integrating style classification algorithms into AR applications would allow users to instantly identify and explore the style of a painting in real time. This could involve overlaying style labels or information directly onto the artwork when viewed through an AR device or providing interactive features allowing users to delve deeper into a particular style's historical context and background.

By pushing the boundaries of fine art analysis by integrating style classification into augmented reality applications and developing advanced automated techniques, we can unlock new possibilities for art enthusiasts, researchers, and even artists themselves. These advancements will contribute to a deeper understanding and exploration of artistic styles, enriching the world of fine art appreciation.

References

[1] E. Fernie, *Art history and its methods: A critical anthology.* Phaidon London, 1995.

[2] Moon, "A brief history of art." `https://medium.com/@cchndn13/a-brief-history-of-art-c3d689fae01f`, Jul 2020. [Accessed 16-September-2023].

[3] M. Doğan and Z. S. Yaşlıca, "Evaluation of gastronomic objects in renaissance painting," *International Journal of Gastronomy and Food Science*, vol. 30, p. 100606, 2022.

[4] L. Booth, "Artistic and Cultural Impacts of Impressionism - Liberty University." `https://digitalcommons.liberty.edu/cgi/viewcontent.cgi?article=2163&context=research_symp`, 05-02-2023. [Accessed 23-Jun-2023].

[5] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, "Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3703–3707, IEEE, 2016.

[6] W. Zhao, D. Zhou, X. Qiu, and W. Jiang, "Compare the performance of the models in art classification," *Plos one*, vol. 16, no. 3, p. e0248414, 2021.

[7] E. Oomen, *Classification of painting style with transfer learning.* PhD thesis, School of Humanities and Digital Science, Tilburg University Tilburg, The . . . , 2018.

[8] S.-h. Zhong, X. Huang, and Z. Xiao, "Fine-art painting classification via two-channel dual path networks," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 137–152, 2020.

[9] B. S. Hantono, L. E. Nugroho, and P. I. Santosa, "Meta-review of augmented reality in education," in *2018 10th international conference on information technology and electrical engineering (ICITEE)*, pp. 312–315, IEEE, 2018.

[10] H. Almurashi, R. Bouaziz, W. Alharthi, M. Al-Sarem, M. Hadwan, and S. Kammoun, "Augmented reality, serious games and picture exchange communication system for people with asd: Systematic literature review and future directions," *Sensors*, vol. 22, no. 3, p. 1250, 2022.

[11] A. M. Al-Ansi, M. Jaboob, A. Garad, and A. Al-Ansi, "Analyzing augmented reality (ar) and virtual reality (vr) recent development in education," *Social Sciences & Humanities Open*, vol. 8, no. 1, p. 100532, 2023.

[12] I. Gupta, S. Dangi, and S. Sharma, "Augmented reality based human-machine interfaces in healthcare environment: Benefits, challenges, and future trends," in *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pp. 251–257, IEEE, 2022.

[13] A. Hajirasouli and S. Banihashemi, "Augmented reality in architecture and construction education: state of the field and opportunities," *International Journal of Educational Technology in Higher Education*, vol. 19, no. 1, p. 39, 2022.

[14] R. Chen, P. Perry, R. Boardman, and H. McCormick, "Augmented reality in retail: a systematic review of research foci and future research agenda," *International Journal of Retail & Distribution Management*, vol. 50, no. 4, pp. 498–518, 2022.

[15] H.-F. Lin, H.-y. S. Tsai, and B. Yeo, "Augmented reality advertising in entertainment programming: An exploration across cultures," *Journal of Creative Communications*, p. 09732586221135062, 2023.

[16] M. Spaziani, "Art Galleries meet Augmented Reality." `https://www.inglobetechnologies.com/art-galleries-meet-augmented-reality/`, Sep 2022. [Accessed 03-September-2023].

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

[18] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[19] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[20] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[21] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, pp. 145–175, 2001.

[22] M. J. Swain and D. H. Ballard, "Indexing via color histograms," in *Active perception and robot vision*, pp. 261–273, Springer, 1992.

[23] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *European Conference on Computer Vision*, vol. Vol. 1, 2004.

[24] L. Shamir, T. Macura, N. Orlov, D. M. Eckley, and I. G. Goldberg, "Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 2, pp. 1–17, 2010.

[25] R. S. Arora and A. Elgammal, "Towards automated classification of fine-art painting style: A comparative study," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3541–3544, IEEE, 2012.

[26] S. Agarwal, H. Karnick, N. Pant, and U. Patel, "Genre and style based painting classification," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 588–594, IEEE, 2015.

[27] F. S. Khan, S. Beigpour, J. Van de Weijer, and M. Felsberg, "Painting-91: a large scale database for computational painting categorization," *Machine vision and applications*, vol. 25, no. 6, pp. 1385–1397, 2014.

[28] Z. Falomir, L. Museros, I. Sanz, and L. Gonzalez-Abril, "Categorizing paintings in art styles based on qualitative color descriptors, quantitative global features and machine learning (qart-learn)," *Expert Systems with Applications*, vol. 97, pp. 83–94, 2018.

[29] C. Florea, C. Toca, and F. Gieseke, "Artistic movement recognition by boosted fusion of color structure and topographic description," in *2017 IEEE Winter*

Conference on Applications of Computer Vision (WACV), pp. 569–577, IEEE, 2017.

[30] C. Florea and F. Gieseke, "Artistic movement recognition by consensus of boosted svm based experts," *Journal of Visual Communication and Image Representation*, vol. 56, pp. 220–233, 2018.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, no. 1106-1114, p. 1, 2012.

[32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833, Springer, 2014.

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

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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

[37] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[38] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

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

[42] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[43] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, "Recognizing image style," *arXiv preprint arXiv:1311.3715*, 2013.

[44] Y. Bar, N. Levy, and L. Wolf, "Classification of artistic styles using binarized features derived from a deep neural network," in *ECCV Workshops (1)*, vol. 8925 of *Lecture Notes in Computer Science*, pp. 71–84, Springer, 2014.

[45] K.-C. Peng and T. Chen, "Cross-layer features in convolutional neural networks for generic classification tasks," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3057–3061, IEEE, 2015.

[46] W.-T. Chu and Y.-L. Wu, "Deep correlation features for image style classification," in *Proceedings of the 24th ACM international conference on Multimedia*, pp. 402–406, 2016.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] A. Lecoutre, B. Negrevergne, and F. Yger, "Recognizing art style automatically in painting with deep learning," in *Asian conference on machine learning*, pp. 327–342, 2017.

[49] M. Kedia, *Fine-grained painting classification.* Thesis, University of Illinois at Urbana-Champaign, 2017.

[50] Y. Yu, O. Jin, and D. Hsu, "Artistic movement recognition using deep cnns," tech. rep., Stanford University, 2017.

[51] A. Elgammal, B. Liu, D. Kim, M. Elhoseiny, and M. Mazzone, "The shape of art history in the eyes of the machine," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[52] F. Bianconi and R. Bello-Cerezo, "Evaluation of visual descriptors for painting categorisation," *IOP Conference Series: Materials Science and Engineering*, vol. 364, p. 012037, jun 2018.

[53] C. Sandoval, M. Lech, and E. Pirogova, "Classification of style in fine-art paintings using transfer learning and weighted image patches," in *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–7, IEEE, 2018.

[54] C. Sandoval, E. Pirogova, and M. Lech, "Two-stage deep learning approach to the classification of fine-art paintings," *IEEE Access*, 2019.

[55] C. Sandoval, E. Pirogova, and M. Lech, "Classification of fine-art paintings with simulated partial damages," in *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–8, IEEE, 2020.

[56] O. Menis-Mastromichalakis, N. Sofou, and G. Stamou, "Deep ensemble art style recognition," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[57] S. P. Pérez and F. G. Cozman, "How to generate synthetic paintings to improve art style classification," in *Brazilian Conference on Intelligent Systems*, pp. 238–253, Springer, 2021.

[58] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[59] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.

[60] S. Bianco, D. Mazzini, and R. Schettini, "Deep multibranch neural network for painting categorization," in *Image Analysis and Processing - ICIAP 2017*, pp. 414–423, Springer International Publishing, 2017.

[61] S. Bianco, D. Mazzini, P. Napoletano, and R. Schettini, "Multitask painting categorization by deep multibranch neural network," *Expert Systems with Applications*, vol. 135, pp. 90–101, 2019.

[62] A. Efthymiou, S. Rudinac, M. Kackovic, M. Worring, and N. Wijnberg, "Graph neural networks for knowledge enhanced visual representation of paintings," *arXiv e-prints*, pp. arXiv–2105, 2021.

[63] M. R. Mohammadi and F. Rustaee, "Hierarchical classification of fine-art paintings using deep neural networks," *Iran Journal of Computer Science*, vol. 4, no. 1, pp. 59–66, 2021.

[64] S.-G. Lee and E. Cha, "Style classification and visualization of art paintings genre using self-organizing maps," *Human-centric Computing and Information Sciences*, vol. 6, pp. 1–11, 2016.

[65] E. Gultepe, T. E. Conturo, and M. Makrehchi, "Predicting and grouping digitized paintings by style using unsupervised feature learning," *Journal of cultural heritage*, vol. 31, pp. 13–23, 2018.

[66] G. Castellano and G. Vessio, "Deep convolutional embedding for digitized painting clustering," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2708–2715, IEEE, 2021.

[67] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24*, pp. 373–382, Springer, 2017.

[68] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[69] C. Sandoval, E. Pirogova, and M. Lech, "Adversarial learning approach to unsupervised labeling of fine art paintings," *IEEE Access*, 2021.

[70] B. Saleh and A. Elgammal, "Large-scale classification of fine-art paintings: Learning the right metric on the right feature," *arXiv preprint arXiv:1505.00855*, 2015.

[71] G. Castellano and G. Vessio, "A brief overview of deep learning approaches to pattern extraction and recognition in paintings and drawings," in *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III*, pp. 487–501, Springer, 2021.

[72] WikiArt, "Visual art encyclopedia." `https://www.wikiart.org/`.

[73] L. Wankmüller, *Style Recognition in Paintings using Deep Learning.* bachlor thesis, Ludwig Maximilian University of Munich, 2017.

[74] C. Florea, R. Condorovici, C. Vertan, R. Butnaru, L. Florea, and R. Vrânceanu, "Pandora: Description of a painting database for art movement recognition with baselines and perspectives," in *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 918–922, IEEE, 2016.

[75] C. Florea, C. Toca, and F. Gieseke, "Paintings Dataset for Recognizing the Art movement. Pandora18k.." `http://imag.pub.ro/pandora/pandora_download.html`. [Online].

[76] C. S. RODRIGUEZ, *DEEP LEARNING TECHNIQUES FOR THE FINE ART AND AUSTRALIAN ABORIGINAL PAINTING ANALYSIS*. PhD thesis, 2021.

[77] Y. Jiao and P. Du, "Performance measures in evaluating machine learning based bioinformatics predictors for classifications," *Quantitative Biology*, vol. 4, no. 4, pp. 320–330, 2016.

[78] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[79] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

[80] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[81] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[82] C. Imane, "EfficientNet [model architecture] — iq.opengenus.org." `https://iq.opengenus.org/efficientnet/`. [Accessed 24-September-2023].

[83] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[84] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

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

[86] K. Team, "Keras documentation: ReduceLROnPlateau — keras.io." `https://keras.io/api/callbacks/reduce_lr_on_plateau/`. [Accessed 20-September-2023].

[87] M. Agarwal, A. Rajak, A. K. Shrivastava, *et al.*, "Assessment of optimizers impact on image recognition with convolutional neural network to adversarial datasets," in *Journal of Physics: Conference Series*, vol. 1998, p. 012008, IOP Publishing, 2021.

[88] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[89] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, no. 5, 2014.

[90] P. Verma, V. Tripathi, and B. Pant, "Comparison of different optimizers implemented on the deep learning architectures for covid-19 classification," *Materials Today: Proceedings*, vol. 46, pp. 11098–11102, 2021.

[91] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: a self-gated activation function," *arXiv preprint arXiv:1710.05941*, vol. 7, no. 1, p. 5, 2017.

[92] A. Soleimanipour, M. Azadbakht, and A. Rezaei Asl, "Cultivar identification of pistachio nuts in bulk mode through efficientnet deep learning model," *Journal of Food Measurement and Characterization*, vol. 16, no. 4, pp. 2545–2555, 2022.

[93] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[94] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[95] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." `https://www.tensorflow.org/`, 2015. Software available from tensorflow.org.

[96] F. Chollet *et al.*, "Keras." `https://github.com/fchollet/keras`, 2015.