Automated Identification of Painters Over WikiArt Image Data Using Machine Learning Algorithms

MSc Research Project
Data Analytics

Mrinmoy Dutta Choudhury
Student ID: x18182658

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mrinmoy Dutta Choudhury
Student ID:	x18182658
Programme:	MSc in Data Analytics **Year:** 2019-20
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	…………………………………………………………………………………………………….
Project Title:	Automated Identification of Painters Over WikiArt Image Data Using Machine Learning Algorithms
Word Count:	…………………………………… **Page Count**……………………………………………….…….

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………………………………………………………………………………………………………

Date: ………………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automated Identification of Painters Over WikiArt Image Data Using Machine Learning Algorithms

Mrinmoy Dutta Choudhury
X18182658

Abstract

Recent times have witnessed digitization of work from various sectors with the advancement of computer vision and artificial intelligence techniques. There has been a significant growth in the establishment of several online art libraries as well. A handful of work has previously been done on the classification of paintings based on style, however, very limited work has been performed towards classification of painters. The traditional approach of annotating images manually is time consuming and demands domain expertise. To enable efficient and faster annotation of image data, this project proposes the use of machine learning and deep learning algorithms to perform identification and classification of painters by extracting complex features from the images of available paintings. Several machine learning algorithms were implemented to accomplish this task and the results were compared using different evaluation matrices. The best classification accuracy of 75 percent was obtained using a pretrained ResNet-50 transfer learning approach. In addition to this, the results of the implemented models have been compared with the results of existing models in the subject.

1 Introduction

The recognition and classification of artwork is an imperative task for monitoring and understanding purposes. Recent times have witnessed digitization of work from all spheres and thus the internet is now home to several online art libraries.

1.1 Background and Motivation

The transition of artwork towards digitization has gained significant importance in making them available to the public in various web repositories (Kelek et al., 2019). Artworks: modern or ancient carry certain metadata, that are annotated by art historians (Saleh and Elgammal, 2015). These artworks carry information pertaining to the genre, style, and the author of the paintings. Collections available online are usually annotated and are easily identified by art experts as a piece of work belonging to an artist or to a genre. This technical report focusses on identification of impressionist painters from the images of paintings by extracting complex features using computer vision technologies. Impressionist painters are artists who impersonate famous people, objects, landscapes, etc[1]. Impressionists differ slightly from regular painters in the sense that they paint a scene or an object as if they only had a glance of it.

A lot of the paintings now available on the web contain collections of work done by painters from different generations (Agarwal et al., 2015). These works have been classified to a large extent by art experts, while a lot of it remains unclassified. This highlights the importance of having in place a system that can act as a classification gateway of artwork based on genre, style, and author.

Machine learning applications for object detection, image classification and feature extraction have gained significant importance in the recent years (Khan and Al-Habsi, 2020). Furthermore, deep learning methodologies such as Convolutional Neural Network (CNN) [2]architectures have produced state-of-the-art results in image classification and object recognition (Seo and Shin, 2018). A handful of work has been done previously on classification of artworks based on genre and style of painting using several machine learning methodologies. However, not a lot of work has been done on the classification of paintings by their authors, although the usage of computer vision for painter identification spans back to the early 2000s. A Naïve Bayes classifier was used alongside extraction of local features from the images (Keren, 2002). Previous work done on painter identification have considered reasonably smaller datasets, details of which are available in the later sections. However, identification of painters from paintings by extracting features is one area that has not yet been vastly explored and is the area on which this project is centered around. This research project uses a much larger dataset for accomplishing the task of painter identification.

The motivation behind this research is to find whether every artist leaves a fingerprint and to build a model that can identify a painter by extracting features from the available images using several machine learning algorithms. Automating these tasks using machine learning algorithms is important as the conventional hand-picked annotation process is time consuming requiring relevant domain expertise. There are two approaches in the literature for the task of identification of painters from images, namely the conventional machine learning and the deep learning approaches. This project focusses on the application of both the approaches and performs a comparative study. The project also looks at the application of transfer learning techniques using two pretrained deep learning architectures for the image classification task. Feature extraction techniques have been used for Random Forest[3] and Support Vector Machine (SVM)[4] classifiers, while the deep learning methodologies automatically extract features on their own as they pass through different hidden layers. The scope of the research is to classify the work done by 10 impressionist painters, namely, Camille Pissarro, Childe Hassam, Claude Monet, Edgar Degas, Henri Matisse, John Singer-Sargent, Paul Cezanne, Paul Gauguin, Pierre-Auguste Renoir, and Vincent van Gogh. Five machine learning methodologies namely, Random Forest, SVM, Convolutional Neural Network (CNN), transfer learning using Resnet-18 and Resnet-50 (Atliha and Šešok, 2020)

---

[1] https://www.tate.org.uk/kids/explore/what-is/impressionism

[2] https://www.jeremyjordan.me/convolutional-neural-networks/

[3] https://builtin.com/data-science/random-forest-algorithm

[4] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/#:~:text=A%20support%20vector%20machine%20(SVM,able%20to%20categorize%20new%20text.

have been implemented. Model evaluations have been carried out and a comparative study showing the performances of the identified models has been highlighted. This project, to the extent of knowledge gathered from the literature is the first to perform the task of painter classification on the largest unified database of artwork, the WikiArt repository, using transfer learning techniques. This repository consists of artworks of over 1000 artists across different styles and genres (Saleh and Elgammal, 2015).

Figure 1: Sample Images of Paintings of Different Impressionist Painters

1.2 Project Requirement Specifications

This research project aims to build a model that can recognize and classify the paintings done by different impressionist painters. This would help in quick segmentation of works by authors providing an easier access to users. The research question and objectives are as follows:

1.2.1 Research Question

The research question focusses on comparing and analysing the performances of several conventional and deep learning machine learning methodologies for painter classification tasks. Machine learning, deep learning, and transfer learning models such as SVM, CNN, ResNet-18, ResNet-50, AlexNet, etc. have proven to be effective in the task of classifying paintings based on style and genre (Zhao et al., 2017). However, little work has been done on the classification of painters by extracting complex features from available images of paintings.

> **RQ:** *"To what extent can classification and predictive machine learning models (RF, SVM, CNN, ResNet-18 and ResNet-50) be used on impressionist painter data to perform multi-class classification and identification of painters based on their style of work?"*

1.3 Research Objectives and Contributions

The investigation of the research question follows a series of objectives and sub-objectives, which are listed below. The research project includes the generation of data from WikiArt database, implementation of painter identification models using conventional machine learning algorithms and deep learning algorithms like CNN and CNN based transfer learning architectures. It also includes evaluation and performance comparison between these models. The objectives of the research are listed in Table 1.

Table 1 Research Objectives

Objective	Description
Obj1	Critically review the literature on artwork and painter classification.
Obj2	Download painter data from the WikiArt database and prepare a balanced dataset and split the images into train, test, and validation sets.

Obj3	Implementation and evaluation of classification and predictive models
Obj3(a)	Implementation and evaluation of Random Forest (RF)
Obj3(b)	Implementation and evaluation of Support Vector Machine (SVM)
Obj3(c)	Implementation and evaluation of Convolutional Neural Network (CNN)
Obj3(d)	Implementation and evaluation of Resnet 18 transfer learning
Obj3(e)	Implementation and evaluation of Resnet 50 transfer learning
Obj4	Comparison between the models developed
Obj5	Comparison of the developed models with existing models in the subject

Contributions: The contribution of the project is the development of a machine learning model that performs classification of painters from images. These models will allow online libraries to classify artwork by their authors and enable users to easily navigate and search the work done by an artist.

The technical report at hand is structured as follows. Chapter 2 highlights and reviews literature relevant to the objectives of this project. Chapter 3 details out the steps involved in the project, the methodologies being used, the architecture and design specifications. The data preparation is also described in Chapter 3. Chapter 4 describes the implementation of the selected algorithms, their evaluation, and results. Chapter 5 and 6 cover the discussion and future work sections, respectively.

2 Literature Review on Classification of Artwork and Painters

2.1 Introduction

The literature review of this technical report focusses on usage of several machine learning algorithms for image classification tasks. This section covers a critical analysis of works in the relevant area and domain from the recent past and focusses on the applications of different machine learning algorithms in object recognition and classification. The literature review is divided into sub-sections to segregate different approaches by putting them under one section.

2.2 Classification of Paintings by Genre and Style

Recent years have witnessed a rapid transformation of artwork to the digital environment that has resulted in the creation of several online libraries (Kelek et al., 2019). The use of automated recognition methods helps generate already existing metadata at a much faster rate and with more efficiency and makes it possible to generate new metadata that relate to the content of paintings. Most artworks present on the web contain annotations in the form of style, genre, date, and location. Earlier studies regarding classification of paintings varied largely in content and size, due to the absence of one common dataset, making comparison of classification accuracies a difficult task (Cetinic and Grgic, 2016). Their work used the WikiArt (Pirrone et al., 2009) dataset that includes a broader set of annotations consisting of genre, style, artists, date, etc. The common challenge of converting painting characteristics into numeric descriptors was met by CNN-derived features, SIFT, HOG, etc. Different approaches were able to extract different set of features from the available paintings. This project makes use of data available from the WikiArt database. Several feature extraction techniques have also been carried out which are discussed in the later sections of this technical report.

An important reason for classification of artwork into classes and sub-classes is that is eases up the search operations. Agarwal et al. (2015) stressed on how it was possible to annotate artworks with a range of metadata available making things easier for buyers, sellers, and viewers. The work focussed on feature extraction from paintings for classification by genre using machine learning algorithms by training a model based on available tags and then letting the machine perform the annotations on unseen data. Their work focused on multi-label genre and style classification. Classification accuracies of around 85 and 62 percent were achieved in classifying the painting by genre and style, respectively.

Another application of painting genre classification has been discussed in the work by Nunez-Garcia et al. (2018). The work focussed on extracting salient features from paintings. The work focused at classification of seven different genres using Artificial Neural Networks (ANN). The experiment showed that by integrating features in the framework, better classification accuracies were obtained. Zujovic et al. (2009) performed a similar task of classification by genre of paintings. The classification performed depended on two features; Gray-level features (Gabor filters and edges) and Colour features (HSV – Hue, Saturation of colours and Value of how dark the colours are). The classification was done using both conventional machine learning methodologies (Naïve Bayes, KNN, SVM, AdaBoost) and Artificial Neural Networks (ANN). The use of multiple features led to better results; however, it was also observed that increase in the number of iterations resulted in significantly greater training times. The results were affected largely by non-uniformity in the dataset, variation in image size and quality.

2.3 Ancient Artwork and Chronological Classification Using Supervised Classification and Transfer Learning

Paintings, especially, the ancient and traditional ones are often considered as invaluable treasures in every culture. Because of several artists and their different style of work, it becomes a tough ask for humans to perform classification and recognition tasks. The work in Liu and Jiang (2014) focused on supervised learning methods to classify traditional Chinese paintings into two different classes, namely, meticulous school and free sketch school. Their work involved data generation, feature extraction and classification; something that would be the approach of this technical report as well. They used Support Vector Machines (SVM) for image recognition and classification using several feature extraction techniques. Classification accuracy of around 80 percent was obtained using the Tamura feature (Karmakar et al., 2017). The experiment yielded accuracy on the higher end of things; however, it was done only on two classes and it would be interesting to see how well the proposed model performs when given the task of multi class classification. This research focusses on multi class classification involving 10 classes.

Another work of ancient art classification has been discussed in Li et al. (2019). Their work was motivated by the great success of deep learning methodologies in image classification tasks. They performed chronological classification of the paintings in Mogao Grottoes (Zou et al., 2014). State of the art classification architectures of CNN like AlexNet, VGGNet and the ResNet were used to investigate the effect of these techniques. Later, the performance of the CNN was tested by making certain modifications to the Resnet architecture. The last mean pooling layer in the ResNet architecture was replaced by a set of sequential layers consisting of a Max Pooling layer, ReLU layer and a convolution layer. The research showed that the modified ResNet architecture produced better results than the three classification methods in the chronological classification of the paintings. Also, due to the small size of the dataset, their work focused on transfer learning, where a pretrained ImageNet model was migrated to new models for optimization. This technical report is based on a similar approach where transfer learning is used to train the model using a pretrained architecture. A model trained on a large dataset like ImageNet[5] is taken and the knowledge is transferred to the WikiArt dataset considered for this project.

Figuring out the era or the time of a painting is a challenging task. Chen et al. (2017) addressed this task using multi-view appearance and colour features using a supervised method of learning. The hypothesis was built around the fact that painting style and era can be determined by multi-view and colour features. They used a dataset from the Mogao Grottoes and were able to obtain better results than some of the other works at that time.

A major motivation behind image classification is to speed up the process of index-based search and retrieval. Arora and Elgammal (2012) performed a comparative study of different classification techniques for automated classification of fine-art genres. The problem was approached using supervised learning techniques. The study was more focused on extracting intermediate level features (Bag-of-Words, BoW) instead of low-level features like colour, lights, etc. SVM based classifier was used on local and semantic level descriptors. The study proved that semantic level features extraction yielded better classification accuracies when compared to the discriminative BoW and generative BoW.

2.4 Classification of Paintings using Pre-Trained Architectures and Usage of WikiArt Database

The past few years have witnessed several scholars perform classification of painting and artwork based on computer vision technologies. With the advent of technology, further experiments started being carried out using machine learning technologies. The more conventional machine learning models lacked the ability to extract precise information from the images of paintings due to lack of information the brushstrokes carry. The hypothesis behind the study by Kim et al. (2019) was based on the fact that visualized depth information of brushstrokes was an effective parameter that would help improve the accuracy of the predictive machine learning models. They built a new data utilization technique with Reflectance Transformation Imaging (RTI) images, that makes the most of visualizations in three-dimensional shape of brushstrokes. RTI images reveals uniqueness in brushstrokes of artists, which is not the case with usual images. Convolutional Neural Networks (CNN) was used for performing the classification task using different architectures like VGG-16, ResNet-50 and DenseNet-121. The results showed improvement in classification accuracies with RTI images as input.

The availability of large collections of art data gives rise to a need of having multimedia systems that can retrieve and archive this data. Humans tend to form impressive perception about things they are exposed to, likewise, a viewer may identify similarities in certain paintings and may be able to figure out something in common about them. However, as discussed in the earlier section, the key here is to develop a system that can recognize similarities in paintings and classify them accordingly, which would invariably save a lot of time. The usual tasks include predicting genre, painting's style, and the artist. Saleh and Elgammal (2015) discussed in detail the list of important features that can be extracted from

[5] http://image-net.org/about-overview

paintings and focussed on machine learning methodologies for the achieving the prediction tasks. They proposed and compared the performances of machine learning methodologies for genre, style and artist identification and classification. The comparison was done using the publicly available WikiArt database.

Object recognition and semantic recognition are two different categories of image classification. While the former is about what an image depicts, the latter is more about understanding what meaning an image conveys. Sandoval et al. (2019) aimed at improving the accuracy of style-based classification by using a two-stage style classification approach. They focused more on the problem with semantic classification of images of fine art. To address the issue, they used a two-stage machine learning technique. In the first stage, the input image is split into several patches and a deep neural network was used to classify each of these patches. In the second stage, the style labels were generated. The main purpose of the second stage was to deal with the potential mistakes the first stage might do at performing the classification tasks. The method proposed used six pre-trained CNN architectures in the first stage and a shallow neural network in the second.

2.5 Classification and Identification of Painters and Gaps

Determining the authenticity in identification of paintings is of prime importance. Identification of painters from the images of paintings is a difficult task because a painter may have different styles of painting and different artists might have the same style. Jangtjik et al. (2017) divided images into multiple patches and tried to study the correlations among the patches of the image using long short-term memory (LSTM)[6]. They proposed a CNN-LSTM model that would return multiple labels for a given image and a fusion technique that determines decision quality of each layer of patches, were combined to get the outcome. The work focused on classification of authors by the images of their paintings. A somewhat similar work was done in the study by Sun et al. (2015). They classified Chinese ink-wash paintings (IWP) using hybrid Convolutional Neural Networks. The CNN network was designed to extract brushstroke features, a replacement to the commonly used methods of analysing colours and edges.

Over the course of time, painters differ in movement, the movement to one style from another, based on the need of the painting. This makes the task of identification of painters from images more challenging. The variation of styles and movements make it difficult for conventional techniques to perform the identification task. Kelek et al. (2019) discussed the use of latest deep neural networks for the task of identification of painters based on the available images of their paintings. Several pretrained CNN architectures, namely GoogleNet, Inceptionv3, Resnet50, Resnet101 and DenseNet were used and a classification accuracy of around 80 percent was obtained using the DenseNet network. However, the research lacked in the sense that only 46 images each from 17 different painters was considered. CNNs tend to perform much better with larger datasets. In this project, a much larger dataset comprising of 500 images per artist has been used.

Levy et al. (2013) performed the task of classification of painters using a genetic algorithm (GA). The proposed methodology was a combination of dimensionality reduction and computation methodologies. The preprocessing done for dimensionality reduction yielded several complex features such as fractal dimension, texture and Fourier spectra coefficients.

[6] https://wiki.pathmind.com/lstm

The dataset taken for the research contained paintings from three different artists. The style of individual artists were recognized by extracting specific features to distinguish between the work by different artists in the study by Cetinic and Grgic (2013). Their method focussed on the measurable elements of an image that can be represented by a set of global image features. The dataset chosen consisted of 25 images for each of the artists considered. Several classifiers like multi-layer perceptron (MLP) , SVM, Naïve Bayes, random forest were implemented. Classification accuracy of around 75 percent was obtained using MLP and random forest with AdaBoost.

Levy et al. (2014) discussed about a novel hybrid approach for the problem of classification of painters. The two algorithms used in their study were genetic algorithms (GA) and deep restricted Boltzmann machines (RBM). The features were extracted using generic image processing functions and deep RBMs. The approach was able to achieve almost 90 percent accuracy for the classification of work done by three painters.

Quantitative analysis of paintings is essential to understand the statistical differences between the artworks of different artists. Narag and Soriano (2019) focussed on differentiating the works of Juan Luna from some of the other Filipino artists by extracting features from different portions of the available images like foreground, background. Neural networks and SVM were applied and an accuracy of around 83 percent was achieved. Their work differed in the sense that most works previously relied on feeding high dimensional features to the machine learning models, whereas, here, they tried to see if smaller combinations of six low dimensional features extracted satisfactory outcomes. This approach required low computational power. However, the task was more centred towards the work done by one particular artist.

The work done on classification of natural images is comparitively more when compared to something more distorted like oil paintings. Liao et al. (2019) addressed this issue. Their work focused on application of algorithms to recognize oil painters from the images of oil paintings. The proposed methodology consisted of a cluster multiple kernel learning algorithm which was able to extract color, texture, and spatial layout features from oil paintings. It then generated multiple kernels with different functions and sub-kernels that produced better classification performance were selected. Based on the literature, painter classification task has not yet been done on the WikiArt dataset using transfer learning techniques, which is addressed in this research.

2.6 Conclusion

In this section, several research papers were studied with regards to the application of computer vision in the classification of artwork based on genre, style and artists. It also focussed on highlighting that although there has been a handful of work done in classification of art by genre and style, automated identification of painters is something that has been carried out by a very few researchers. Obj1 was addressed in this section. The next section presents the methodology and design adopted in this technical report.

3 Painter Identification Methodology and Design

3.1 Painter Identification Approach

Most data mining related projects follow the Knowledge Discovery in Databases (KDD)[7] and CRISP DM[8] methodologies. In this research, however, the KDD methodology has been followed. KDD is a vast concept that makes use of computer vision, machine learning and statistical approaches to extract potentially meaningful information from the available data (Vernickel et al., 2019). The overall flow of the project is presented that shows the stages involved in the implementation of the research from data collection to evaluation of the results. The KDD flow diagram for painter identification is shown in Figure 2.

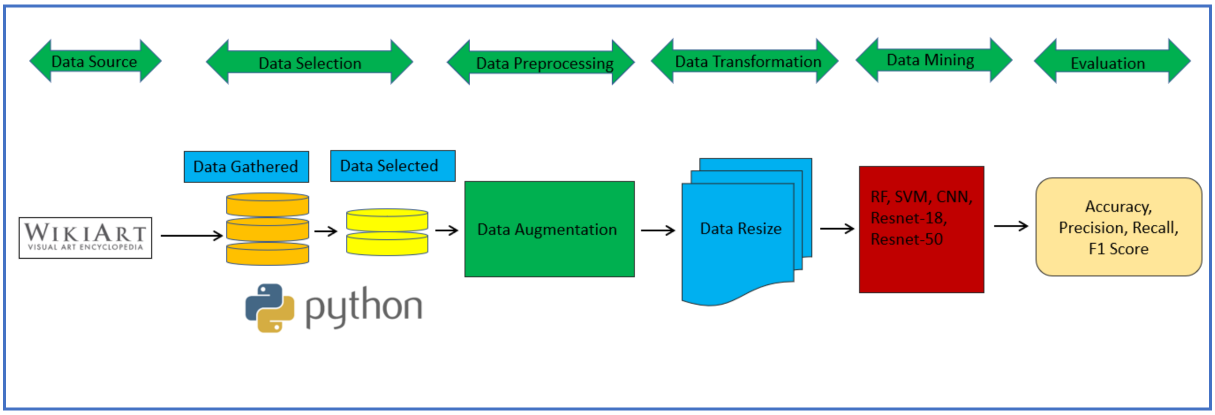

Figure 2 Methodology for Painter Identification

3.1.1 Data Selection

The dataset chosen for this project was taken from the WikiArt[9] paintings database which is the largest online collection of fine artworks. This database has images of around 81,500 paintings from 1,119 artists ranging from fifteen centuries to contemporary artists (Saleh and Elgammal, 2015). It contains paintings from multiple artists, genre, time, and style. For this project, images of paintings done by 10 artists was downloaded using a script provided by Lucas David [10]. The downloaded data, however, varied largely in numbers from around 512 images for Paul Gauguin to close to 2000 images for Vincent Van Gogh as shown in Figure 3.

[7] https://www.javatpoint.com/kdd-process-in-data-mining
[8] https://www.sv-europe.com/crisp-dm-methodology/
[9] https://www.wikiart.org/
[10] https://github.com/lucasdavid/wikiart

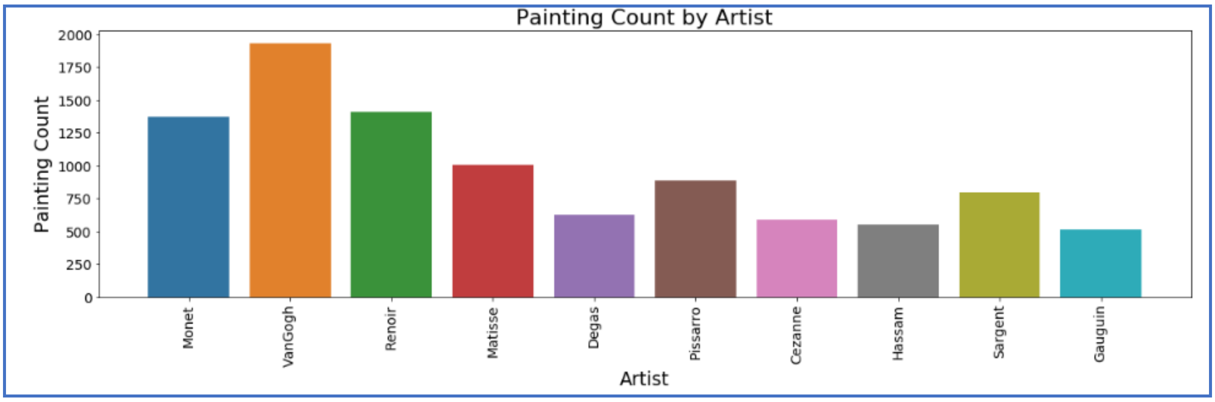

Figure 3 Number of Paintings by Artists

3.1.2 Data Pre-Processing, and Transformation

Figure 3 shows a large variation in the distribution of data. To overcome this, the downloaded data was split into 380, 50 and 70 images for training, validation, and test sets respectively for each artist using a python script. The script was modified to include a block that would download a specified number of images to the test folder as well. A total of 500 images per artist (Figure 4) was considered for the experiment, making it a total 5000 images for 10 impressionist artists in total. Obj2 of this project was addressed here.

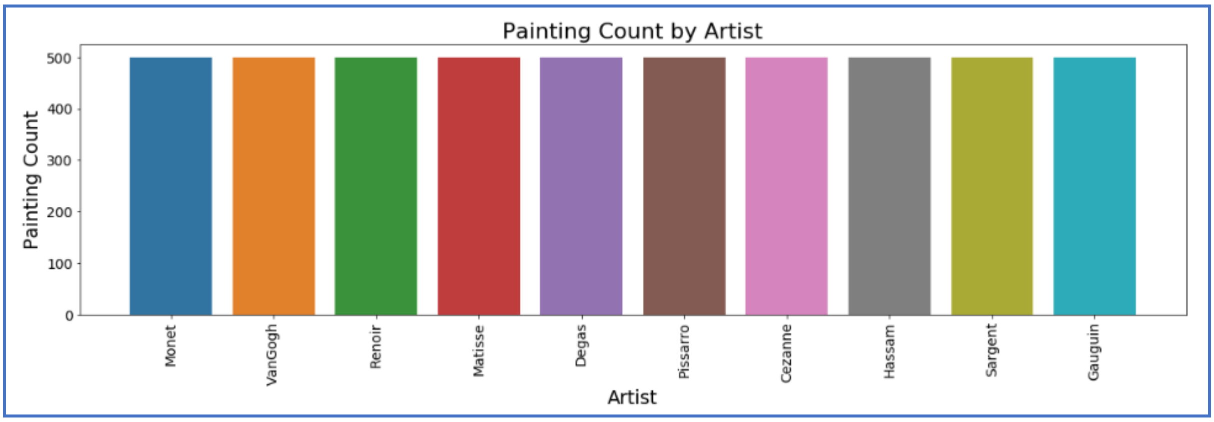

Figure 4 Paintings by Artists Considered for Implementation

Some of the previous studies like the work in Kelek et al., (2019) and Cetinic and Grgic (2016) that were done on painter identification were able to obtain reasonable classification accuracies using 46 and 25 images per artist, respectively. This technical report makes use of a much larger number of samples; 500 images per artist. However, as deep learning algorithms tend to perform better with larger set of records, data augmentation was performed on the training set to increase the diversity of the training samples without collecting new samples for the experiment. Several data transformations were performed using the transform library from pytorch. Transformations like random resized crop, random horizontal flip, normalize, random rotation, center crop, etc. were performed and clubbed together using the transforms.Compose functionality of PyTorch[11] deep learning framework. The detailed description of the same is provided in the respective implementation sections.

[11] https://ai.facebook.com/tools/pytorch/

3.1.3 Data Mining

The conventional machine learning algorithms require feature extractions and principal components analysis to be performed externally. In this research, for the SVM and Random Forest, feature extractions were carried out and several features were extracted. However, these feature extraction techniques are not relevant to deep learning algorithms. The application of deep learning algorithms like CNN are capable of extracting features from images as they pass through different hidden layers before finally performing the classifications in the final layer. This feature in deep learning cuts the need to carry out feature extraction and principal component analysis (PCA) externally. Algorithms like SVM, Random Forest, Convolutional Neural Network (CNN), Resnet-18 and Resnet-50 transfer learning were applied on the data to perform the task of multi-class classification.

3.1.4 Data Interpretation and Evaluation

The results of the implemented models are presented and are compared in section 4.9. The performances of the algorithms are plotted using a confusion matrix and a classification report. The relevant F1 scores, accuracy and precision are calculated for measuring the model performance and are presented within the implementation section of each of the models. The primary components and formulas for calculation of these evaluation matrices are discussed below.

Primary Components

- True Positive (TP): True Positives are a measure of the classification model correctly predicting the positive class.
- True Negative (TN): True Negatives are a measure of the model correctly predicting the negative class.
- False Positive (FP): False Positives are a measure of the model that fails to predict the positive class correctly.
- False Negative(FN): False Negatives give a measure of the model incorrectly predicting the negative class.

Formulas for Calculating the Evaluation Matrices:

- Precision: Precision is the ratio of the number of correctly predicted positive class to the total number of positive predictions made by the model (correct and incorrect).

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- Recall: Recall is the ratio of the predicted positive observations to all the observations belonging to the positive class.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- F1- Score: F1-score is calculated as a weighted average between precision and recall12.

[12] https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=Precision%20%2D%20Precision%20is%20the%20ratio,the%20total%20predicted%20positive%20observations.&text=F1%20score%20%2D%20F1%20Score%20is,and%20false%20negatives%20into%20account.

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

- Accuracy: Accuracy is the ratio of the total correct predictions to the total number of observations.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Since the dataset is well balanced, accuracy can be considered as a good evaluation matrix. However, for each model implemented, all the above-mentioned matrices are calculated.

3.2 Design Specification Process Flow

The three-tier architecture followed for the implementation of this project is shown in Figure 5, briefly outlining the steps followed and the technologies and tools used.

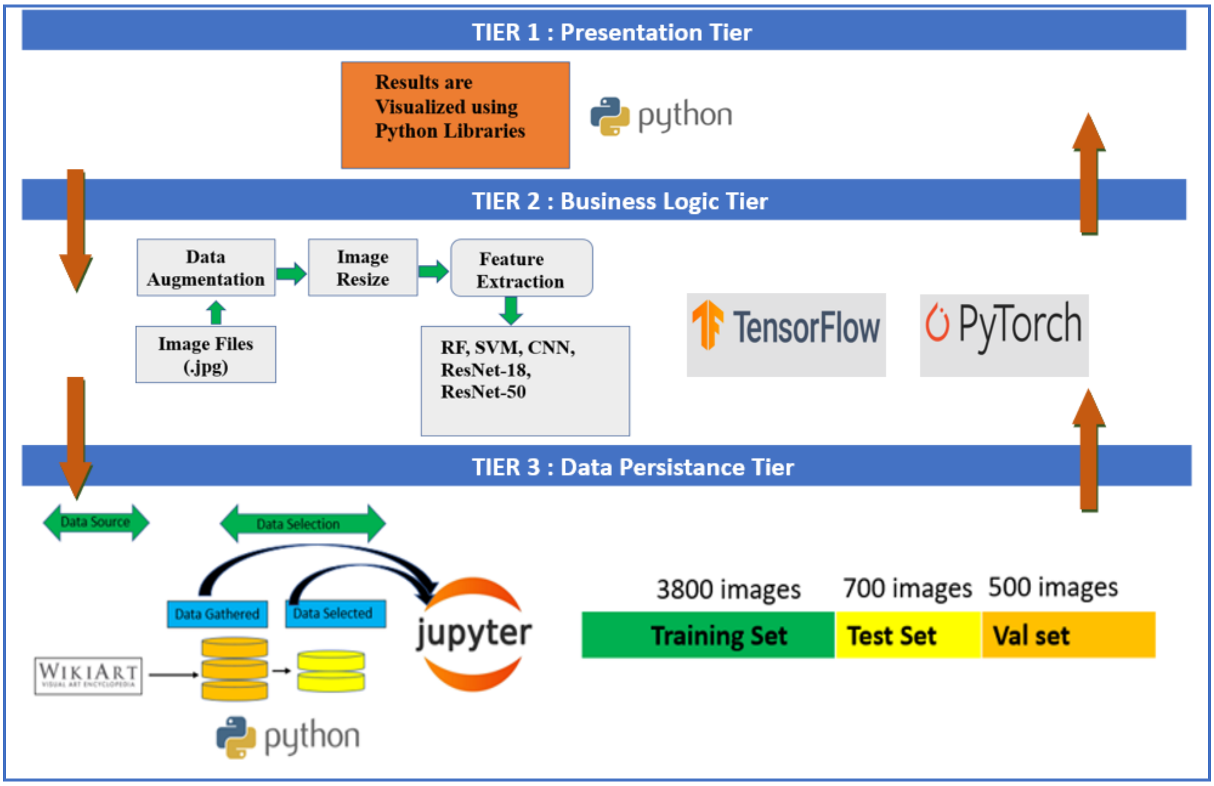

Figure 5 Impressionist Painter Classification Design

3.3 Conclusion

The painter identification methodology approach for the research was developed to suit the requirements of the project. The required data for the project was extracted from the WikiArt database. To address the issue with imbalance of data, 500 images each of 10 impressionist painters were considered for the final implementation. The train and test data sets were created from the selected images. The following chapter presents the implementation steps, evaluation matrices and results obtained from the implemented methodologies.

4 Implementation, Evaluation and Results of Painter Classification Models

4.1 Introduction

The implementation, evaluation and results of the models used for identification of painters from the images of paintings are discussed in this section. The project focusses primarily on the implementation of supervised learning methods as the networks are fed with labelled inputs. The extraction of features from the paintings is also discussed in this section. Confusion matrix and classification reports[13] are plotted to evaluate the performance of the implemented models. The performance of the machine learning algorithms used are determined using accuracy, precision, recall and F1 scores for each of the models. A comparison of the developed models is carried out and these models are also compared with the existing models. The model that gives the best performance is selected.

4.2 Creation of the Dataset from WikiArt Database:

The WikiArt dataset was chosen as it is a reliable source for online digitized artwork containing artworks belonging to different styles, genre, and artists. The paintings of 10 impressionist painters were considered for the research. These 10 impressionist painters constitute the 10 classes for this multi-class classification problem. They are Camille Pissarro, Childe Hassam, Claude Monet, Edgar Degas, Henri Matisse, John Singer-Sargent, Paul Cezanne, Paul Gauguin, Pierre-Auguste Renoir, and Vincent van Gogh. These images were downloaded using a python script. However, the downloaded data was biased as the number of images for some of the artists were lower than the others. To overcome this issue, 500 images per artist were considered and were split into training, validation, and test sets using python scripts[14]. The split up of the datasets for each of the models is mentioned under their respective implementations in the following sections.

4.3 Implementation, Evaluation and Results of Random Forest

4.3.1 Feature Extraction from Images

Unlike with deep learning algorithms, feature extraction must be performed externally for conventional machine learning algorithms like RF and SVM. Before applying the RF model, several global feature extraction techniques were carried out, namely, Hu moments, Haralick texture, and Histogram.

Texture is an important factor that needs to be considered for automated interpretation of images and is particularly important for tasks involving image classification (Salhi et al., 2018). It differentiates between different classes based on same Gray level. Haralick method examines texture known as the Gray Level Cooccurrence Matrices (GLCM). The Histogram of Oriented Gradients (HOG) method of feature extraction is used for computation of gradient and the gradient route of an input image (Mahmud et al., 2018). The Hu Moments consist of a set of seven numbers that remain invariant to transformation of images. The method extracts invariant moments that are not affected on application of zooming, translation, and rotation functionalities to an image (Lv et al., 2020).

[13] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
[14] https://github.com/panchambanerjee/Impressionist-Classifier

4.3.2 Implementation

Random Forests are ensemble learning methods consisting of one or more decision trees for performing classification and regression tasks. It combines the outputs from different models that are trained on a subset of data, using a bagging algorithm [15]. The results are obtained using a voting classifier. This way the issue of overfitting is reduced. This method was implemented using the RandomForestClassifier function provided by the skikit learn library in python [16]. Because the implementation of conventional methods like Random Forest and SVM do not require a separate validation set, all the images were read from the same folder. A total of 5000 images were considered for this implementation in a 90-10 train and test set ratio making it 4500 and 500 images for the train and test sets, respectively. The number of trees were set to 80, 100, 120, 150. The classes were encoded as follows:
["Cezanne":0,"Degas":1,"Gauguin":2,"Hassam":3,"Matisse":4,"Monet":5,"Pissarro":6,"Renoir":7,"Sargent":8,"VanGogh":9]

4.3.3 Evaluation and Results

The performance of the model was evaluated with the aid of a classification report and a confusion matrix. The best prediction accuracy of 53.4 percent was obtained with 120 trees. The classification report shows that average precision, recall, F1-score, and accuracy obtained were 0.55, 0.53, 0.54 and 0.53, respectively. The classification report and the confusion matrix are shown in Figures 6(a) and 6(b), respectively.

	precision	recall	f1-score	support
0	0.49	0.55	0.52	44
1	0.50	0.50	0.50	54
2	0.38	0.47	0.42	51
3	0.54	0.69	0.61	49
4	0.60	0.55	0.57	44
5	0.59	0.48	0.53	63
6	0.34	0.38	0.36	42
7	0.55	0.59	0.57	37
8	0.65	0.57	0.61	54
9	0.78	0.56	0.65	62
accuracy			0.53	500
macro avg	0.54	0.53	0.53	500
weighted avg	0.55	0.53	0.54	500

Figure 6(a) RF Classification Report

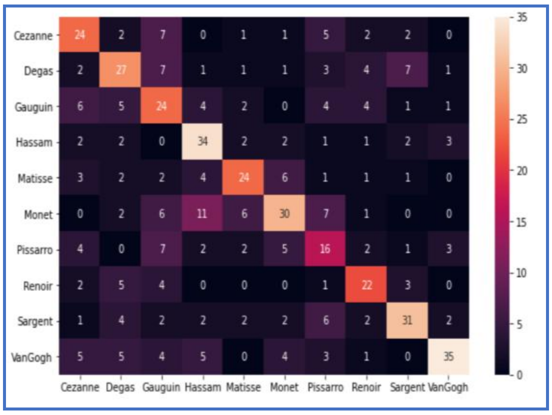

Figure 6(b) RF Confusion Matrix

The values of True Positives (TP) for the 10 classes are diagonally placed in the confusion matrix in Figure 6(b). The accuracy thus obtained is not satisfactory.With the implementation and evaluation of Random Forest, the Obj 3(a) was solved.

4.4 Implementation, Evaluation and Results of Support Vector Machine Model

Support Vector Machine (SVM) is a supervised learning method used for classification and regression problems. SVM uses a kernel based technique to transform data based on which it finds optimal three dimensional hyperplanes to segregate or separate the data on the basis of classes. The same feature extraction techniques that were used for the implementation of Random Forest were used for SVM as well.

[15] https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/
[16] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

4.4.1 Implementation

The SVM model was trained on 4500 images and tested on 500 images. The classes were encoded in a manner similar to the encoding adopted for the implementation of the Random Forest algorithm (section 4.3). The model was implemented in a grid with different values for C value, kernel functions (linear and rbf) and gamma parameter. SVM was implemented using the skikit learn python library and SVC function[17].

4.4.2 Evaluation and Results

A poor classification accuracy of around 42 percent was achieved on the test set. The classification report (showing the accuracy, precision, recall, F1 scores) and the confusion matrix for the experiment on the test set are shown in Figures 7(a) and 7(b), respectively. A precision of 0.43, recall value of 0.42, f1-score of 0.42 and accuracy of 0.42 were obtained from the experiment. The result of the SVM classifier was lower than the Random Classifier. Obj 3(b) was thus met, with the implmentation and evaluation of SVM.

	precision	recall	f1-score	support
0	0.36	0.50	0.42	44
1	0.34	0.33	0.34	54
2	0.37	0.35	0.36	51
3	0.38	0.59	0.46	49
4	0.58	0.43	0.49	44
5	0.49	0.33	0.40	63
6	0.26	0.29	0.27	42
7	0.38	0.38	0.38	37
8	0.44	0.37	0.40	54
9	0.65	0.58	0.62	62
accuracy			0.42	500
macro avg	0.42	0.42	0.41	500
weighted avg	0.43	0.42	0.42	500

Figure 7(a) SVM Classification Report

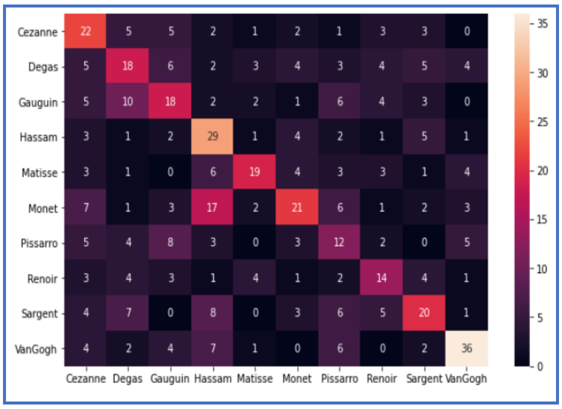

Figure 7(b) SVM Confusion Matrix

4.5 Implementation, Evaluation and Results of Deep Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm that is particularly known for its outstanding capabilities in image classification and recognition. It does so by extracting features from the input images as it passes from each hidden layer. Each hidden layer extracts certain features from the images and classification happens in the last layer, which is a fully connected layer.

4.5.1 Implementation

The CNN developed for the experiment consisted of 3 convolutional layers having 32,64 and 128 filters respectively. These layers extract several features from the sub-regions of the images that are under consideration. Two different feature or kernel sizes, 5 X 5 and 3 X 3 are considered, that scan over the images and produce a new output matrix. The three convolutional layers were activated using the Relu activation function followed by 2D max pooling layers. The pooling layers reduce the size of the data. 2X2 max-pooling windows are applied on the images after they pass through the Relu activation function. The data emerging out of the final max-pooling layer is flattened to a single vector before they are passed onto the fully connected dense layers. The dense layers aggregate the features that were learnt in

[17] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

the convolutional layers. The actual classification happens in the dense layers. The size of the last layer depicts the number of classes considered, which is 10 in this case. This layer was activated using a softmax activation function. Dropout layers were added to deal with the issue of overfitting. Dropout layers turn off a certain proportion of the neurons at random reducing the dependency on training data for performance. The model was compiled using both adam and stochastic gradient descent (SGD) optimizer and a categorical_crossentropy loss function, since the research is a multi-class classication problem.

CNN was implemented using the keras library in python using 5000 images (500 from each of the 10 classes. The data was split into 90-10 train and test ratio. The model was trained on 4500 training images and tested on 500 images.

4.5.2 Evaluation and Results

An overall classification accuracy of only 36 percent was obtained on the test set using the adam optimizer. The model did not show any significant improvement upon performing data augmentation as well. The classification report and confusion matrix depicting the performance of the CNN model are shown in Figures 8(a) and 8(b) respectively. Precision, recall and f1-score obtained were 0.41, 0.36 and 0.35, respectively.

	precision	recall	f1-score	support
0	0.38	0.21	0.27	58
1	0.30	0.29	0.29	55
2	0.29	0.52	0.37	46
3	0.58	0.14	0.22	51
4	0.43	0.40	0.41	45
5	0.37	0.54	0.44	50
6	0.27	0.52	0.36	48
7	0.58	0.26	0.35	43
8	0.55	0.37	0.44	49
9	0.35	0.38	0.37	55
accuracy			0.36	500
macro avg	0.41	0.36	0.35	500
weighted avg	0.41	0.36	0.35	500

Figure 8(a) CNN Classification Report

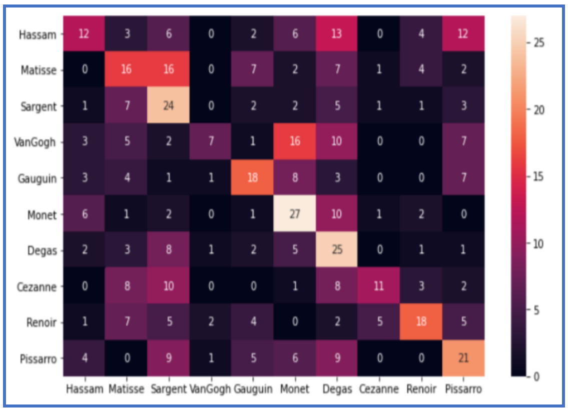

Figure 8(b) CNN Confusion Matrix

The train and validation loss-accuracy plot are shown in Figure 8(c).

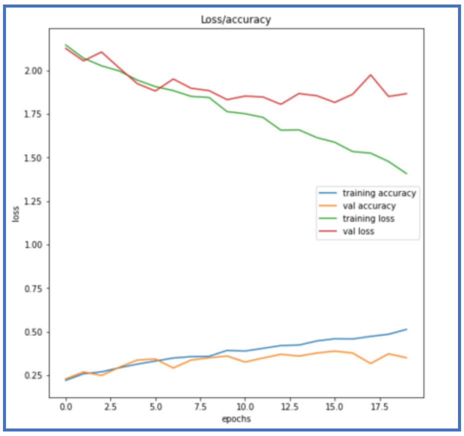

Figure 8(c) Loss-Accuracy Curves

From the above plot, the training accuracy increases linearly with every epoch. However, the validation accuracy shows fluctuations and there are random drops in accuracy in every 2-3 epochs. The validation loss too does not decrease linearly like the training loss. Thus, the basic CNN model did not perform as expected. Obj 3(c) was met with the implementation and evaluation of CNN.

4.6 Pretrained CNN Architectures using Transfer Learning

The basic idea behind transfer learning is to take a model that has been trained on a large dataset and transfer the knowledge acquired by that model to a smaller dataset. Transfer learning is particularly useful in situations where there is a lack of training data or training data may only be available over the course of time (Hirt et al., 2020). The convolutional layers extract low-level features such as lines, edges, patterns, etc and the final layers recognize the specific classes in the images. From the results obtained in sections 4.3, 4.4, 4.5, it can be concluded that the predictions and classification accuracies have not been as per expectations even with a reasonably large dataset. To overcome this, transfer learning has been implemented in this project as pretrained models are made of many hidden layers which extract the minute features from images. Pretrained CNN architectures, namely, ResNet-18 and ResNet-50 have been implemented in this project.

4.7 Implementation, Evaluation and Results of ResNet-18 Transfer Learning

4.7.1 Implementation

ResNet-18 is pretrained convolutional neural network (CNN) which is 18 layers deep, meaning it consists of 18 hidden layers. ResNet-18 has been trained on the large sized Imagenet dataset[18]. A pretrained ResNet-18 network is loaded. The ResNet-18 transfer learning was implemented using the PyTorch machine learning library. Image processing is carried out using the transforms functionality provided by PyTorch. To deal with the relatively smaller data size, data augmentation is performed. Random transformations are applied to the training inputs. Other transformations like resize, center crop were performed to the train, test and validation sets. ImageNet models require the dataset size to be 224 X 224 and hence the input images were resized accordingly[19]. Once all the transformations are carried out, the data is converted to tensors that eventually go into the network. The images are finally normalized with precomputed mean and standard deviations.

Once the transformations were carried out, the train, test and validation datasets were created using the datasets.ImageFolder function. PyTorch assigns correct labels to the images once the folders are set up properly. The generated datasets are then passed onto DataLoader that generates batches of labels and images. A batch size of 32 has been considered for the implementation. The pretrained ResNet model is then loaded using the models functionality of the torchvision library. The two ways to customize a pretrained model are feature extraction and fine-tuning[20]. In this project, the feature extraction technique has been implemented wherein, only the weights of the final layer are updated where are the predictions take place. The pre-trained CNN is used as a fixed feature extractor. Initially, all the weights of the model are frozen and only the custom defined fully connected dense layers are trained. The number of outputs in the last dense layer is set to the number of classes, which is 10 in this case. The model was compiled using the categorical cross entropy loss

[18] https://www.mathworks.com/help/deeplearning/ref/resnet18.html

[19] https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce

[20] https://www.tensorflow.org/tutorials/images/transfer_learning

function and the performances of the SGD and Adam optimizer were carried out. The training, validation and test sets consisted of 3800, 500 and 700 images, respectively.

4.7.2 Evaluation and Results

The model was compiled using both SGD and adam optimizers on two different instances. Classification accuracy of around 68 percent was obtained using SGD optimizer after 20 epochs. The application of transfer learning showed significant rise in accuracy when compared to the CNN implemented in section 4.5. The classification report and confusion matrix showing the performance of the ResNet-18 transfer learning is show in Figures 9(a) and 9(b), respectively. A value of 0.68 was obtained for precision, recall and f1-score.

	precision	recall	f1-score	support
0	0.66	0.67	0.67	70
1	0.72	0.77	0.74	70
2	0.71	0.67	0.69	70
3	0.71	0.49	0.58	70
4	0.78	0.76	0.77	70
5	0.55	0.54	0.55	70
6	0.61	0.59	0.60	70
7	0.64	0.80	0.71	70
8	0.69	0.73	0.71	70
9	0.73	0.79	0.76	70
accuracy			0.68	700
macro avg	0.68	0.68	0.68	700
weighted avg	0.68	0.68	0.68	700

Figure 9(a) ResNet-18 Classification Report

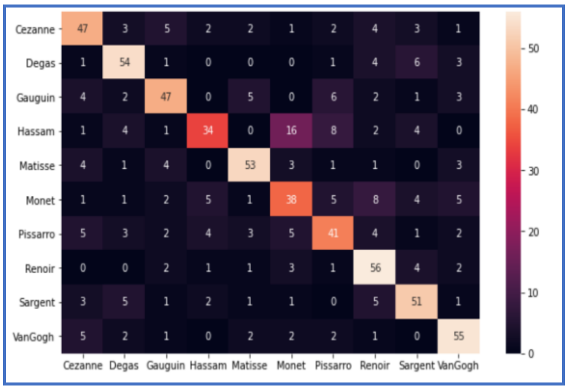

Figure 9(b) Resnet-18 Confusion Matrix

With the implementation of ResNet-18 transfer learning, Obj 3(d) was addressed.

4.8 Implementation, Evaluation and Results of ResNet-50 Transfer Learning

4.8.1 Implementation

Similar to ResNet-18, ResNet-50 is another pretrained CNN network trained on over a million images from the ImageNet dataset[21]. ResNet-50, as the name suggests, is 50 layers deep. The transformations and augmentations performed are similar to the ones performed during the implementation of ResNet-18. The implementation was done on 3800 training, 500 validation and 700 test images. The ResNet-50 transfer learning was implemented using the PyTorch library.

4.8.2 Evaluation and Results

The model showed the best classification accuracy of around 75 percent using the SGD optimizer. There was an improvement of performance when compared to the performance of ResNet-18 transfer learning. The classification report showing the values of precision, recall and f1-score and confusion matrix are shown in Figures 10(a) and 10(b), respectively.

[21] https://www.mathworks.com/help/deeplearning/ref/resnet50.html#:~:text=ResNet%2D50%20is%20a%20convolutional,%2C%20pencil%2C%20and%20many%20animals.

	precision	recall	f1-score	support
0	0.77	0.71	0.74	70
1	0.80	0.70	0.75	70
2	0.77	0.71	0.74	70
3	0.66	0.79	0.72	70
4	0.85	0.87	0.86	70
5	0.69	0.59	0.64	70
6	0.62	0.64	0.63	70
7	0.84	0.80	0.82	70
8	0.74	0.86	0.79	70
9	0.77	0.83	0.80	70
accuracy			0.75	700
macro avg	0.75	0.75	0.75	700
weighted avg	0.75	0.75	0.75	700

Figure 10(a) ResNet-50 Classification Report

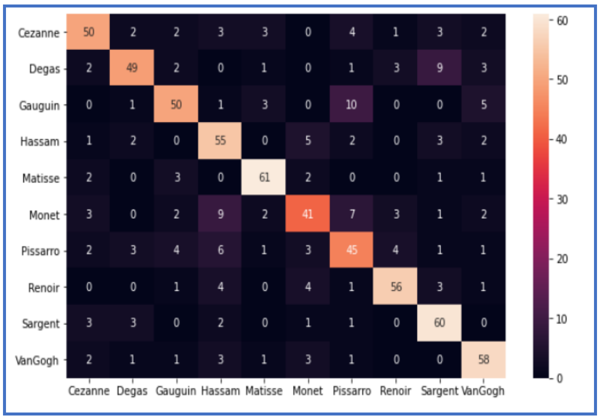

Figure 10(b) ResNet-50 Confusion Matrix

With the implementation of ResNet-50 transfer learning, Obj 3(e) was solved.

4.9 Comparison of the Implemented Models

A comparison of the models implemented for this project, namely, Random Forest, SVM, CNN, ResNet-18 and ResNet-50 is shown in Table 2.

Table 2 Comparison of Implemented Models

Sl. No	Name of the Model	Feature Extraction	Optimizer	Dense Layer Activation Function	Accuracy (in percentage)
1	Random Forest	Hu Moments, Haralick, Histogram	NA	NA	53
2	Support Vector Machine	Hu Moments, Haralick, Histogram	NA	NA	42
3	CNN	NA	Adam	SoftMax	36
			SGD	SoftMax	35
4	ResNet-18	NA	Adam	SoftMax	66
			SGD	SoftMax	68
5	ResNet-50	NA	Adam	SoftMax	70
			SGD	SoftMax	75

The comparison between the models implemented for painter classification accomplishes Obj4 of this technical report.

4.10 Comparison of the Developed Models with Existing Models

From Table 2, the ResNet-50 transfer learning model achieved the highest accuracy of 75 percent in the classification of painters. Table 3 shows the comparison of this model with the performances of some previously developed models that were also developed for the identification and classification of painters.

Table 3 Comparison with Existing Models

Sl. No	Author	Method Implemented	Type of Classification	Number of Classes	Accuracy (in percentage)
1	Kelek et al. (2019)	GoogleNet, DenseNet, ResNet, Inceptionv3	Multi-class	17 classes, 46 images per class	78 percent
2	Narag and Soriano (2019)	SVM	Binary Classification	2 classes, 12 images per class	83 percent
3	Levy et al., (2014)	Genetic Algorithms and deep RBMs	Multi-class	3 classes, 40 images per class	90 percent
4	Cetinic and Grgic (2013)	MLP, SVM, Naïve Bayes, Random Forest	Multi-class	20 classes, 25 images per class	77 percent

Some of the best works in painter classification from images have been listed in Table 3. Most of these works, however, were done on comparatively smaller datasets. The work stated in this technical report was carried out on a much larger dataset having 10 classes and a total of 5000 images and the best classification accuracy obtained was around 75 percent. The comparison of the implemented model with previously built models completes Obj5 of the project.

4.11 Conclusion

The implementation, evaluation and results of the models implemented for the task of classification fully answers the research question stated in the section 1.2.1. In addition to this, the research objectives stated in Table 1 were also accomplished. The performances of the pretrained architecture models were better as these have been trained previously on huge datasets containing a lot of classes and produced benchmark results. In this project, the ResNet-50 transfer learning model produced the best results and is thus, the model selected for the classification task.

5 Discussion

This project happens to be the first to make use of transfer learning techniques on the WikiArt database for painter classification alongside other methodologies. The WikiArt database consists of artworks performed by over 1000 artists across several genres and generations. This project was performed on a smaller subset of data consisting of artworks by 10 impressionist painters. In this research project, feature extraction techniques were used to perform classification using Random Forest and SVM. Apart from these methods, deep learning architectures were implemented on the data and the best results were obtained with the use of ResNet-50 transfer learning technique. Precision, recall, F1 scores of 0.75 was obtained and a classification accuracy of around 75 percent was obtained with RestNet-50 transfer learning. The scope of the project was to perform a multi-class classification to identify and classify the works done by 10 impressionist painters by extracting features from the images of their paintings. The model performed reasonably well in performing the multi-class classification, although it fell short in terms of accuracy when compared to the work in

Kelek et al. (2019), Levy et al. (2014) and Narag and Soriano (2019). A much larger dataset was used when compared to some of the previous studies. Pre-processing was carried out to have 500 images for each of the 10 classes considered. This research provides a comparison of the various approaches towards the classification task highlighting the poor performance of the conventional approaches. It also provides knowledge of how transfer learning techniques significantly help better the classification results when the data at hand is insufficient for the experiment.

The ResNet-50 transfer learning model that obtained 75 percent classification accuracy may be applied to classify the works performed by other artists as well, which makes the model plausible. The performance of the model can possibly be tested to painter data gathered from multiple sources to check the how it performs on varying data coming from different data sources. Also, this project implements only two transfer learning architectures and it would be interesting to see how some of the other pretrained architectures perform on the task at hand.

6 Conclusion and Future Work

The primary objective of the project that had to be answered was*, "To what extent can classification and predictive machine learning models (RF, SVM, CNN, ResNet-18 and ResNet-50) be used on impressionist painter data to perform multi-class classification and identification of painters based on their style of work?"*

In addition to that, this project identified a list of objectives which included the implementation of several machine learning methodologies for performing the task of image classification. Machine learning algorithms like Random Forest, SVM, CNN, ResNet-18 and ResNet-50 were implemented on the impressionist painter dataset. The results of all these models were evaluated using precision, recall, F1 score and accuracy. Through these experiments, the best classifier was able to classify the images to its respective painter with 75 percent accuracy, which was slightly lesser than the best works in the field, however, performed reasonably well.

Painter identification from images is a difficult task as different painters may exhibit the same style and a painter may display different styles in their work. The conventional machine learning methodologies that need external feature extraction techniques performed poorly, as it is a tough ask to identify the exact features that need to be extracted. The basic CNN with three hidden layers also did not work well. Hence, transfer learning techniques were implemented, and significant performance improvement was observed. It can thus be concluded that in scenarios where the data size is not large or the proposed networks do not perform well on the data, it is a good idea to implement pretrained transfer learning techniques, which are trained on large datasets.

The dataset chosen for this project was well labelled and not prone to a lot of noise. In the future, it would be interesting to see how these models perform on other benchmark datasets and datasets with noise. A larger dataset may lead to better classification accuracies. It would also be interesting to see how unsupervised learning methods perform on painter data, as hardly any work using unsupervised learning has been carried out in this subject.

Acknowledgement

I would like to thank my Parents, Sister and my Nephew for the love and support throughout. Special mention goes out to Dr. Catherine Mulwa for her constant support, supervision, and guidance throughout the length of this research project.

References

Agarwal, S., Karnick, H., Pant, N. & Patel, U., 2015. *Genre and Style Based Painting Classification.* Waikoloa, IEEE Winter Conference on Applications of Computer Vision, pp. 588-594.

Arora, R. S. & Elgammal, A., 2012. *Towards automated classification of fine-art painting style: A comparative study.* Tsukuba, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pp. 3541-3544.

Atliha, V. & Šešok, D., 2020. *Comparison of VGG and ResNet used as Encoders for Image Captioning.* Vilnius, Lithuania, IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), pp. 1-4.

Cetinic, E. & Grgic, S., 2013. *Automated painter recognition based on image feature extraction.* Zadar, Proceedings ELMAR, pp. 19-22.

Cetinic, E. & Grgic, S., 2016. *Genre classification of paintings.* Zadar, International Symposium ELMAR, pp. 201-204.

Chen, L. et al., 2017. Multi-View Feature Combination for Ancient Paintings Chronological Classification. *Journal on Computing and Cultural Heritage (JOCCH)*, pp. 1-15.

Hirt, R., Kühl, N., Peker, Y. & Satzger, G., 2020. *How to Learn from Others: Transfer Machine Learning with Additive Regression Models to Improve Sales Forecasting.* Antwerp, IEEE 22nd Conference on Business Informatics (CBI), pp. 20-29.

Jangtjik, K. A., Ho, T., Yeh, M. & Hua, K., 2017. *A CNN-LSTM framework for authorship classification of paintings.* Beijing, IEEE International Conference on Image Processing (ICIP), pp. 2866-2870.

Karmakar, P. et al., 2017. *Improved Tamura Features for Image Classification Using Kernel Based Descriptors.* Sydney, International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1-7.

Kelek, M. O., Calik, N. & Yildirim, T., 2019. *Painter Classification Over the Novel Art Painting Data Set via The Latest Deep Neural Networks.* Nanning, Proceedings of the 9th International Conference of Information and Communication Technology [ICICT-2019], pp. 369-376.

Keren, D., 2002. *Painter identification using local features and naive Bayes.* Quebec City, Object recognition supported by user interaction for service robots, pp. 474-477.

Khan, A. I. & Al-Habsi, S., 2020. Machine Learning in Computer Vision. *International Conference on Computational Intelligence and Data Science*, pp. 1444-1451.

Kim, J. et al., 2019. Classification of oil painting using machine learning with visualized depth information. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 617-623.

Levy, E., David, O. E. & Netanyahu, N. S., 2014. *Genetic Algorithms and Deep Learning for Automatic Painter Classification.* New York, Genetic and Evolutionary Computation, pp. 1143-1150.

Levy, E., David, O. & Netanyahu, N. S., 2013. *Painter classification using genetic algorithms.* Cancun, IEEE Congress on Evolutionary Computation, pp. 3027-3034.

Liao, Z. et al., 2019. An Oil Painters Recognition Method Based on Cluster Multiple Kernel Learning Algorithm. *IEEE Access,* pp. 26842-26854.

Liu, C. & Jiang, H., 2014. *Classification of traditional Chinese paintings based on supervised learning methods.* Guilin, IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 641-644.

Li, X., Zeng, Y. & Gong, Y., 2019. *Chronological Classification of Ancient Paintings of Mogao Grottoes Using Convolutional Neural Networks.* Wuxi, IEEE 4th International Conference on Signal and Image Processing (ICSIP), pp. 51-55.

Lv, C., Zhang, P. & Wu, D., 2020. Gear Fault Feature Extraction Based on Fuzzy Function and Improved Hu Invariant Moments. *IEEE Access,* pp. 47490-47499.

Mahmud, I. et al., 2018. *Efficient Noise Reduction and HOG Feature Extraction for Sign Language Recognition.* Gazipur, International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), pp. 1-4.

Narag, M. J. G. & Soriano, M. N., 2019. *Identifying the painter using texture features and machine learning algorithms.* New York, Cryptography, Security and Privacy, pp. 201-205.

Nunez-Garcia, I., Lizarraga-Morales, R. A. & Hernandez-Gomez, G., 2018. *Classification of Paintings by Artistic Genre Integrating Color and Texture Descriptors.* New York, Artificial Intelligence and Pattern Recognition, pp. 66-70.

Pirrone, R. et al., 2009. *WikiArt: An Ontology-Based Information Retrieval System for Arts.* Pisa, Ninth International Conference on Intelligent Systems Design and Applications, pp. 913-918.

Saleh, B. & Elgammal, A., 2015. *A Unified Framework for Painting Classification.* Atlantic City, NJ, IEEE International Conference on Data Mining Workshop (ICDMW), pp. 1254-1261.

Salhi, K., Jaara, E. M., Alaoui, M. T. & Alaoui, Y. T., 2018. *GPU Implementation of Haralick Texture Features Extraction Algorithm for a Neuro-morphological Texture Image Segmentation Approach.* Kenitra, International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), pp. 1-4.

Sandoval, C., Pirogova, E. & Lech, M., 2019. Two-Stage Deep Learning Approach to the Classification of Fine-Art Paintings. *IEEE Access,* pp. 41770-41781.

Seo, Y. & Shin, K.-s., 2018. *Image Classification for Vehicle Type Dataset Using State-of-the-art Convolutional Neural Network Architecture.* New York, Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference, pp. 139-144.

Sun, M. et al., 2015. *Brushstroke based sparse hybrid convolutional neural networks for author classification of Chinese ink-wash paintings.* Quebec City, IEEE International Conference on Image Processing (ICIP), pp. 626-630.

Vernickel, K. et al., 2019. *A Revised KDD Procedure for the Modeling of Continuous Production in Powder Processing.* Macao, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 340-344.

Vijayan, V. & KP, P., 2020. A Comparative Analysis of RootSIFT and SIFT Methods for Drowsy Features Extraction. *Third International Conference on Computing and Network Communications,* pp. 436-445.

Zhao, B., Huang, B. & Zhong, Y., 2017. Transfer Learning With Fully Pretrained Deep Convolution Networks for Land-Use Classification. *IEEE Geoscience and Remote Sensing Letters,* pp. 1436-1440.

Zou, Q. et al., 2014. Chronological classification of ancient paintings using appearance and shape features. *Pattern Recognition Letters,* pp. 146-154.

Zujovic, J. et al., 2009. *Classifying paintings by artistic genre: An analysis of features & classifiers.* Rio De Janeiro, IEEE International Workshop on Multimedia Signal Processing, pp. 1-5.