Machine Learning-Based Numerical Characterization of Paintings Using Variation in Brushstroke Styles

Gundeep Singh
gxs372@case.edu

Supervised by Prof. Michael Hinczewski

Department of Physics
Case Western Reserve University

Submitted on April 27, 2020 to the
Senior Project Committee,
Department of Physics, CWRU

Abstract

The capability of automatically extracting micro-scale height information from paintings has further advanced the possibilities of quantitatively analyzing artworks, which are especially needed to identify cases of art forgeries. In this work, we have proposed a deep convolutional neural network (CNN) based algorithm that is capable of distinguishing between four similar paintings and identifying the corresponding artists who painted them. Our data consist of small tiles extracted from the paintings and magnitudes of vertical brushstroke heights at every 2D transverse location on the tiles. We have hypothesized that variation in the style and pattern of brushstroke can be used to 'teach' an artificial intelligence algorithm to distinguish between the four artists. A comparison between the network performance of different training architectures supports our hypothesis, where we have demonstrated overall accuracies of over 80% for classification of small image tiles.
In the latter part of this work, we have utilized the training characteristics of four-artist classification to develop and implement a complementary neural architecture to perform single-class outlier detection, which has direct applications in detecting art forgeries. The network is trained on height profile of a particular target artist along with a group of paintings from other outlier artists. The network is then capable of differentiating style of any other artist, not used in the training dataset, from that of the target class. Thus, using the combined capability of our networks, we are able to distinguish the stylistic patterns of individual artists from one another. The workflow developed in this project has potential to be utilized to separate 'real' art from its imitation.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Michael Hinczewski, for giving me the opportunity to work under his guidance and mentorship. I am also grateful to Prof. Hinczewski for his unwavering support and enthusiasm throughout the course of this project. Furthermore, I would like to thank two of Hinczewski Lab's former students, Shishir Adhikari and Marcio O'Dwyer, for providing helpful machine learning resources during the initial stages of the project.

I would also like to thank all the members of our weekly meeting group – Dr. Kenneth Singer, Dr. Ina Martin, Fang Ji, Jake Trookman, Michael McMaster, and Lauryn Smith. These weekly meetings and conversations were vital in shaping the direction of this project. I am especially grateful to Fang for being an awesome coding partner.

I am also grateful to my Senior Project instructor Dr.Gary Chottiner for providing valuable information and feedback both during and outside of the seminar sessions. Finally, I am also thankful to my external committee member, Prof. Harsh Mathur for his insights and discussions during the project update meetings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In 2014, a Geneva-based Fine Art Expert Institute (FAEI) reported that between 70 to 90 percent of the artwork that the institute analyzed turned out to be fake. [8] Similarly, in 2018, a small art museum called 'The Museé Terrus' in Elne, France, saw its collection cut by more than half when it was revealed that 82 of the museum's 142 works were forgeries. [20] Investigations into the cases of fake artwork or art forgeries currently require a combination of a variety of techniques including X-ray, infrared scan and carbon dating in order to scientifically separate fakes and forgeries from real artwork. Institutes like FAEI charge up to $1900 to verify the authenticity of paintings using a combination of aforementioned techniques. [1] In pursuit of lowering this cost and eliminating the labor-incentive methodologies involved, in this study, we propose a unique combination of 3D profilometry technique and Artificial Intelligence (AI) that lays the foundation for developing automated systems that are capable of distinguishing between real art and their imitations.

Non-contact 3D profilometry is a surface analysis technique that measures surface morphology of materials without touching the sample. Therefore, this technique has been prominent in measuring micro-scale height variation in the solid film of dried paint, while avoiding any shape alteration that may be caused by contact technologies. [10] In this study, we propose to develop a deep-learning algorithm that is capable of distinguishing between paintings by utilizing the data obtained from non-contact 3D profilometry scans of the paintings. To the extent of our knowledge, this is the first time AI and profilometry have been used in combination to quantitatively analyze the surface features and patterns in paintings.

1.2 Literature Review

The first publication that proposed the application of identifying characteristic stylistic patterns in solving cases of art forgeries was published in 1968 by Theodore Rousseau. [16] Rousseau cited a case in which a forger used a chemical medium similar to Bakelite to make the painting appear centuries old in just a few hours, and thus became successful in escaping the identification of forgery through the X-ray method. [16] The forgery in this case was only identified much later when stylistic differences in the use of color palette were recognized by a team of art historians that included Rousseau. [16]

Artist-specific stylistic patterns can comprise of various factors that can be utilized to distinguish real art from its imitations. However, one factor that has contributed the most toward scientific generalization of style-identification methodology of paintings is the variation in artist-specific brushstroke techniques. [15] In 2013, Zang et al. developed a mathematical algorithm to automatically extract stroke features from paintings and used them to characterize different stroke-related artistic styles. [21] Their main purpose was to lay out a foundation towards the development of style transfer algorithm using stroke-based artistic style characterization. However, this task was recently accomplished in 2015 by implementing an artificial intelligence system based on deep neural network. [6]

Deep learning-based neural networks have been remarkably successful in solving many related problems in both forensics ([3]) and analysis of artworks like classification of painting genres ([9]). However, as Gatys et al. described in their 2015 deep learning-based style transfer paper, the task of characterizing artistic 'style' is fairly recent and conceptually different than the classical problem of characterizing artistic 'content'. [6] Style is a series of distinguishable and recurring characteristics, while content is related to the subject matter of the artistic piece. [5] A single content can be expressed in different artistic styles, depending upon the artist's training, experience, and interpretation.

In order to characterize an artist's style, it is essential to disentangle their painting's content and style. This entanglement can never be done perfectly. However, to make it more efficient, in this work, we selected only height data in small patches as our training data, to highlight the brushstroke-related stylistic features in the swamp of content-specific information. This approach of only using height data is in conjunction with previous studies that have shown that height characteristic data contain relevant information about extracting brushstroke styles [21].

The task of style characterization consists of processes that include identification, extraction, and grouping of common style elements and features. Previous studies in this regard have accomplished classification of art on the basis of their genres and periods ([9]), transfer of artistic styles to reconstruct painting photographs in different 'general' styles ([6]), and brushstroke characterization and identification of Vincent Van Gogh's painting photographs ([14]). However, classification of brushstroke style of any general artist based off of only height data has not been attempted to date - this clearly has direct applications to our problem of identifying cases of forgeries.

1.3 Background

1.3.1 Deep Learning

Deep learning is a subclass of machine learning algorithms that derive their hierarchical structure from biologically-inspired neural systems. Deep learning networks are also referred to as hierarchical feature learning modules as they consist of layers of mathematical transformations that are arranged together to extract features or representation from input data. The most widely applied transformations in deep learning are convolution operations, and the networks that are based on these convolution operations are known as Convolutional Neural Network (CNN). In each of the convolution operation layer, the input data is convoluted with multiple filter arrays to produce a feature map, which is then used as in input for another mathematical operation. The basic idea behind implementing a deep learning algorithm is to optimize the parameters involved in these operations such that your resulting features match the desired features. The optimization is performed using the process called backpropagation, where operation parameters of the network are changed according to the magnitude of the loss function between the desired and resulting features. A number of iterations of this process are performed in order to minimize the loss and maximize the network performance.

1.3.2 Transfer Learning

Reaching the global minimum of a huge parameter space is a computationally expensive and challenging task. Transfer learning is a method in machine learning where knowledge of one pretrained network is 'transferred' to a new network to simplify the task of finding the configuration that minimizes the loss. [13] Transfer learning has been exceptionally successful in improving neural network performances, especially in cases when access to training data is limited. [18] Since accessing multiple paintings by artists involved in cases of alleged forged painting can be particularly challenging or even impossible, transfer learning can be of significant advantage in our problems.

In this work, we have used several popular transfer learning-based architectures, including a network architecture called VGG-16[1], which has been pretrained on the ImageNet dataset [4]. ImageNet is a continually expanding training dataset of images obtained using different modalities that has been pretrained to identify 100,000 different categorical images ranging from 'balloon' to 'strawberry'. It is truly remarkable that a neural network that was originally trained for general google search images can be utilized and customized to separate artistic style from content, as we will show in this thesis work. A plausible explanation for this is that neural network has to become invariant of the specifics of the image input at deeper layers in order to be able to identify omnipresent features in images. Thus, transfer learning on pretrained ImageNet images can be utilized

[1]VGG-16 gets its name from the group at the University of Oxford, UK that first designed and trained the network. The group is called 'Visual Geometry Group'. The numeral 16 in the name indicates that it is a 16-layer model

to extract stylistic features and thereby, improve network performance.

1.3.3 Supervised and Unsupervised Learning

As we will describe in detail in the next section, in this work we have attempted to solve two problems - matching a patch of a painting to its artist out of four given choices and detecting stylistic outliers for one of the artists. These two problems are classified as supervised and unsupervised machine learning problems respectively. In supervised learning, all of the data (training and testing) is 'labelled', i.e. the desired feature categories are known and the task is to maximize the classification performance. A classic example of supervised learning would be to train a neural network using images of cats and dogs, and then evaluate classification performance by inputting a different test picture of either cat or dog. In unsupervised learning, on the other hand, the network is trained without strictly defining the classes or categories. In this work, we will use unsupervised learning to detect stylistic outliers. In general, these outliers can belong to any possible artist with features of different styles, as it can be in the case of art forgery, and we cannot train the network on all possible artist styles. Therefore we would have to rely on the power of unsupervised learning to detect outliers, i.e. imitations of 'real' art.

1.4 Description of Problem

The overall goal of this project is to develop and test a methodology that can directly be translated for applications in identifying cases of art forgeries. In this regard, we have concretely defined the problem in two parts:

1. Artist Style Classification: Given height data from paintings of multiple artists, we want to implement deep learning algorithms (both traditional and transfer) to match patches obtained from the paintings to their corresponding artists. The patch sizes should be small enough such that they only capture and highlight details of brushstroke styles and not the content of the paintings.

2. Single-Class Anomaly Detection: Given height data from paintings of 'target' artist(s),

we want to develop a machine-learning based tool that flags out out-of-class brushstroke style datapoints from the in-class brushstroke style, i.e. patches painted by the target artist(s). In the case of art forgery, the 'target' artist will be the one whose style is being attempted to be copied.

While our two problems may appear to be disconnected from each other, the second part of our project actually depends on the deep learning model developed in part one. Therefore, we need to be able to solve both the problems to arrive at a successful methodology to be translated to real-world art forgery-related problems. Finally, as suggested in the title, we want to be able to characterize the brushstroke style of our artists numerically. This means that our final result would include probabilities and confidence intervals related to how sure we are about identification and classification of the 'style' of each of our artists.

Chapter 2

Method

2.1 Painting Data

We asked 4 artists (1-4) from Cleveland Institute of Art to paint 3 similar paintings (a-c) in equivalent conditions, including the availability of color palette, brush widths and canvas size. High-resolution images of the painting photographs are shown in Fig. 2.1. The twelve paintings were then scanned with a non-contact 3D profilometer. A schematic of the optical profilometer is shown in Fig. 2.2 (and is reproduced from [7]). As shown in this figure, the optical profilometry works by measuring the differences in the path length of two light beams - one that reflects off the surface with height h and the other that reflects off an almost completely reflective surface. The path difference generates an interference pattern, which is then converted to physical height values at different spatial points. Profilometry scans of images 1a, 2a, 3a, and 4a are shown in Fig 2.3. Each of the paintings is 3000x2400 pixel square (px sq.) in size, where each pixel corresponds to a length of about 50 μm. The heights are shown to be centered around the mean height value of the overall painting in Fig 2.3.

As seen in Fig. 2.3, the height variation between images is not easily identified by naked eye. This can also be inferred from Fig 2.4, which shows the histogram plots of all 3 paintings combined for each of the 4 artists. However, by implementing Artificial Intelligence we expect to find small local spatial patterns of brushstroke that are capable of distinguishing between the artists.

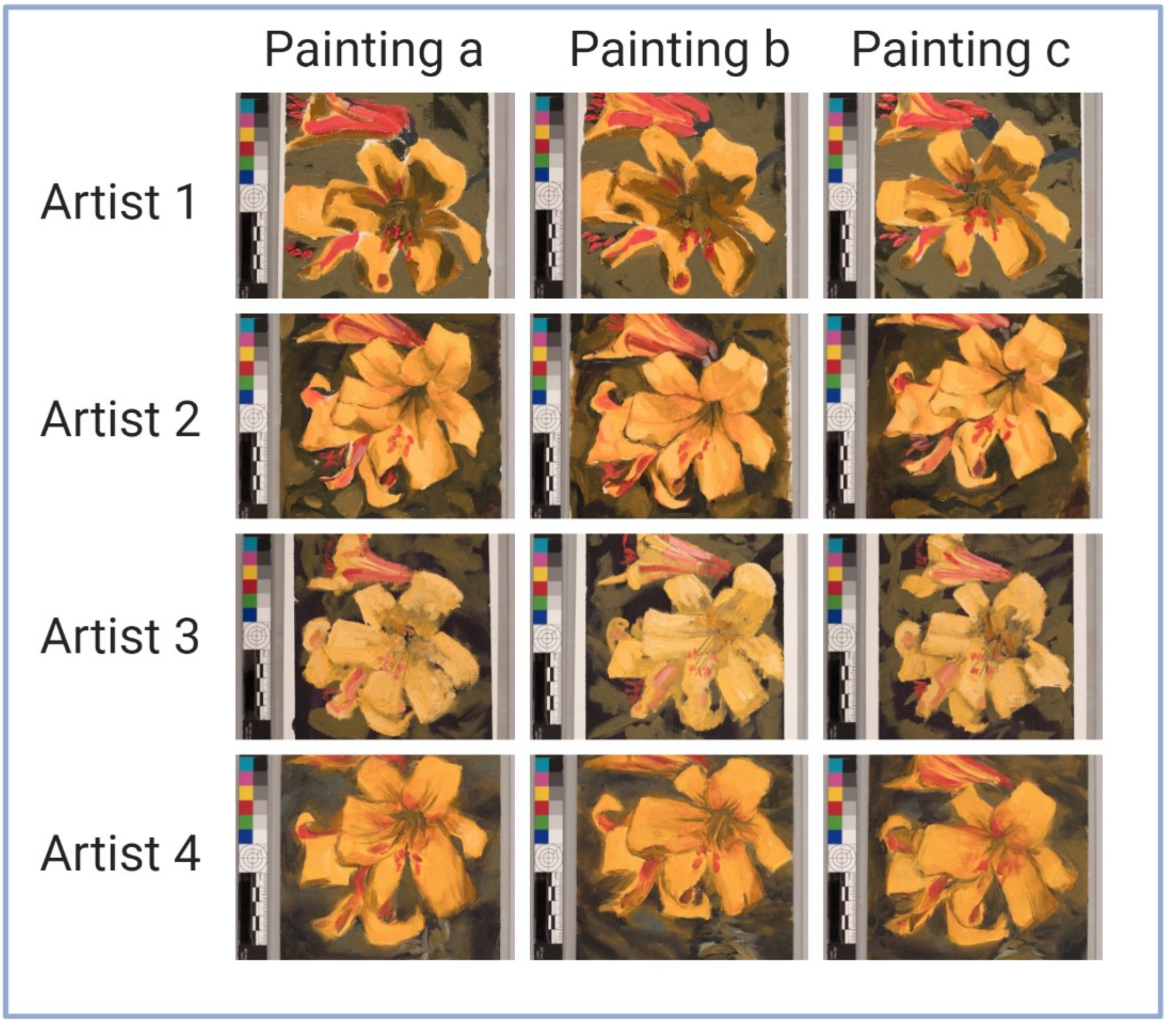

Figure 2.1: High resolution photographs of the 12 paintings of water lilly painted by 4 student artists from Cleveland Institute of Art. Each artist painted 3 duplicates of the painting, shown in the figure as paintings a,b, and c.

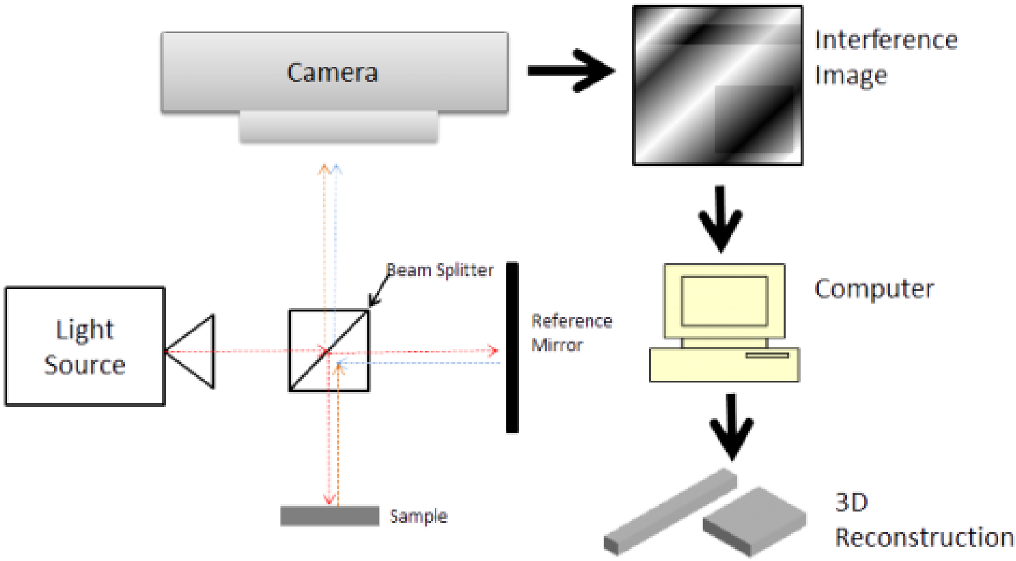

Figure 2.2: Schematic of Non-Contact 3D Profilometer: Visible light is shined on a beam splitter. One ray travels to a reflected mirror, while the other is reflected by a sample point with a certain roughness. When the 2 beams are collected at the detector, interference pattern is observed due to the path difference, which is then decoded to give 3D reconstruction of the surface of the sample.

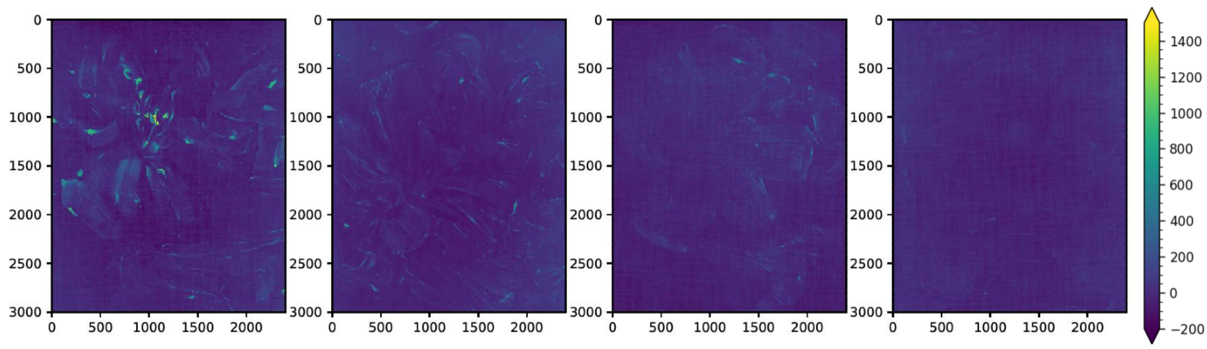

Figure 2.3: Height profilometry data shown for paintings 1a, 2a, 3a, and 4a (L-R). The color bar shown on the right indicates the height values of the dried paint in micrometers.

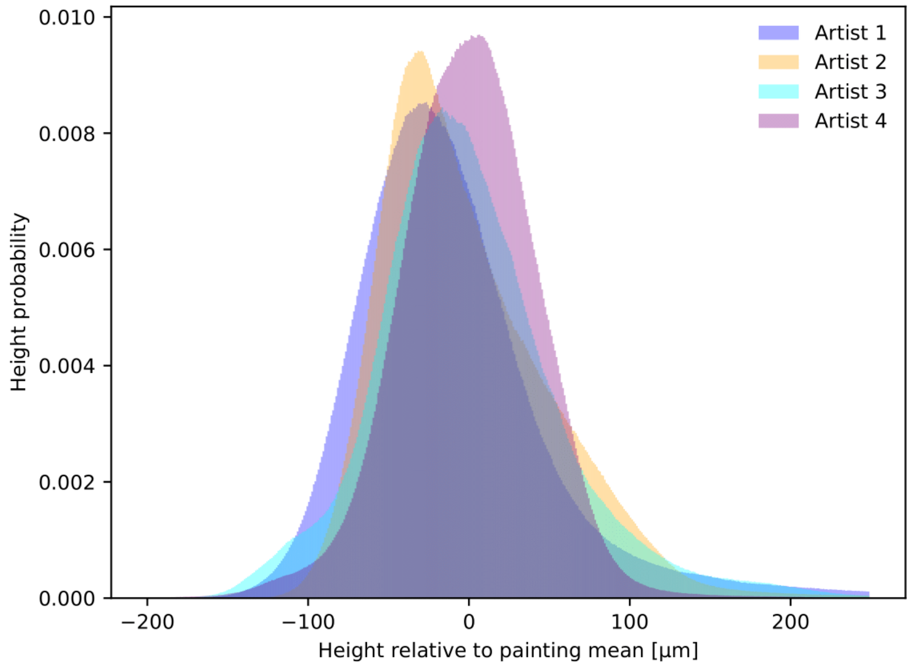

Figure 2.4: Histogram depicting the probability of occurring of each of the height measurement centered around the overall mean height. The three images are combined for each of the 4 artist and the histogram is obtained

As we want to characterize art style and not the content of the painting, we use 'small' patches of painting tiles as our training data, i.e. we want the network to match 'small' patches of the height data to their corresponding artists, which is significantly more challenging than matching complete paintings to their artists. In this case, we evaluate the network performance over a patch size range of 80x80 to 800x800 pixel square, i.e. $0.088\bar{8}$ % to $8.8\bar{8}$ % of the complete painting.

2.2 Data preprocessing

Data preprocessing is a crucial step in machine learning. In order to find consistent patterns in brushstroke styles across different paintings, the training data also needs to be consistent. First, to remove the 'offset' across painting profilometry data, we simply subtract the mean of the overall height magnitudes from each of the datapoint. Next, to make our input patches more consistent, we tried multiple data processing techniques, including the following two popular techniques:

1. Min-Max Scalar - Each patch is scaled such that the minima of the height magnitude equals 0, while the maxima of the height magnitude equals 1.

2. Univariate Standardization - Each patch is normalized such that the standard deviation of the height magnitude equals 1.

The processed data is then 'labeled' from 0 to 3 (in the increments of 1), corresponding to the 4 artists. Labeling is essential in machine learning, as it is used to estimate the loss of the network, which thereby influences the backpropagation or 'training' of the network.

2.3 Neural Network Architecture

2.3.1 Four Artist Classification

We developed and tested various deep learning networks to perform identification and classification of painting patches into the four artist labels. The network architectures we developed in this part of the project can be categorized into the following two categories - those that are trained from 'scratch' using just the painting data (referred to as 'simple' CNN from here on) and those based on ImageNet pretrained networks (referred to as 'transfer' CNN from here on).

Network schematic of a simple CNN is shown in Fig. 2.5. The input layer accepts a square patch tile as an input and then performs a series of convolution operation on them. We varied the number of convolutional layers (thereby changing the level of feature abstraction) and compared the overall network performance for neural networks with different number of convolutional layers. After convolution operations, a pooling operation is performed to 'pool' out only high-weighted feature nodes. This is followed by fully-connected layers, which map the extracted features to even lower-dimensional vector spaces. In this operation all the nodes in two consecutive layers are interconnected (thus the name fully-connected) through trainable weight parameters. Finally, the resulting vector space is mapped to four final nodes representing the four artists. In practice, we get four weight values related to each of the nodes, which correspond to the probability-wise predictions

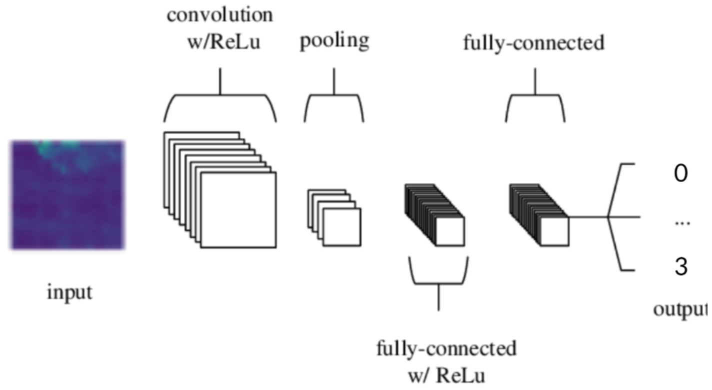

Figure 2.5: Simple Convolutional Neural Network Architecture

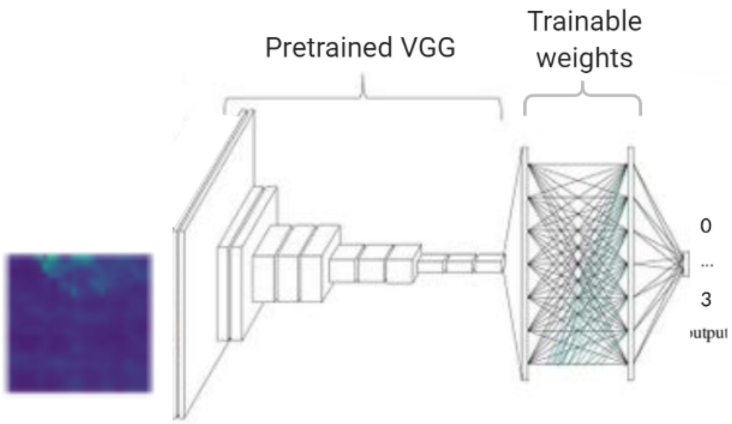

Figure 2.6: Transfer Learning-based Neural Network Architecture (VGG-16)

of the network. Categorical cross-entropy loss is estimated and the parameters of the network are shifted according to a gradient. We want to minimize this loss function, to get the 'best' match of the predicted probabilities to the expected labels.

Network schematic of one of the transfer learning-based neural network VGG16 is shown in Fig. 2.6. The input layer this time is resized to 224x224, as ImageNet dataset consists of images of this size. We keep the convolutional layer structure of the pretrained VGG16 network [17], but replace its ImageNet classification layers with two fully-connected and one four-class node for artist prediction. We only train the weights of the added layers and keep the pretrained weights as non-trainable. However, after the first 25 sets of iterations (called as epochs), we let all the weights of the network become trainable and then train the network for subsequent epochs. This approach first customizes the added classification layers to our profilometry dataset, and then finetunes the training to optimize the network performance, i.e. classification of patches into the four artist classes.

2.3.2 Single-Class Anomaly Detection

Anomaly or outlier detection is a subclass of problems, usually tackled using machine learning, where the goal is to be able to detect events (often rare) that do not belong to the single-majority data class. These problems have found relevant physical application in areas such as identifying bank frauds, structural defects, or automatized medical diagnosis. Developing an outlier detection system has potential for direct applications towards identifying cases of art forgeries, specifically those instances that are related to style-based forgeries. To this end, we first utilized the four artist classification neural network (as described in the previous section) and re-trained them to 'classify' or distinguish the target class from outlier class. Using the difference in capabilities of recognizing different styles, we designed and carried out multiple experiments where we defined and tested different combinations of target and outlier class.

2.4 Network Performance Evaluation

2.4.1 Four Artist Classification

We randomly selected two of the three paintings (a,b,c) by each of the four artists (1,2,3,4) to train the neural network and used the remaining painting as our testing and validation case for evaluating the performance of our classification network. We repeated this process for k=10 times as part of the k-fold cross validation. The accuracy of the network is estimated by considering the predicted class with maximum probability to be the network prediction and comparing it with the known label of the artist who painted the patch. We estimated both individual artist accuracies and overall network accuracy associated with the CNN architecture.

2.4.2 Single-Class Anomaly Detection

We selected artist 1 to be the 'target' artist for testing our network for single-class anomaly detection. The accuracy of the network is estimated by constructing confusion matrix and estimating both overall accuracies and F_1-score values for the network. Calculating F_1-score is a necessary test for the outlier detection network, since the testing class is (usually) imbalanced, i.e. it has more instances of in-class test cases than those that are outliers (or vice-versa). A description for how F_1-score measurement is estimated and how it fixes the issue of class imbalances is provided in Appendix A.

Chapter 3

Results

3.1 Four Artist Classification

As described in section 2.1, for each of the four-artist classification neural networks, we varied the patch size over a range of 80x80 to 800x800 pixel square (px sq.) and calculated the overall network performance. For our 'simple' classification CNN, the results are shown in Fig. 3.1, where we have shown the overall testing accuracy over the patch-size range for multiple network architectures that differ in the number of convolutional layer operations (refer to Fig. 2.5 for simple CNN architecture schematic). In this network performance evaluation, we used 80% of the total number of labeled patches (randomly selected) for training, 10% for validation, and the remaining 10% for testing. As it can be seen from Fig. 3.1, the network is able to achieve above 80% testing accuracies for certain patch sizes (between 100x100 to 160x160 px sq.) regardless of the depth of the network. At higher patch sizes, however, the uncertainty and stochasticity in the network performance is also high.

When we evaluated the networks using only 2 randomly selected painting for each artists (as described in section 2.4.1), i.e. only using $66.\bar{6}\%$ for data for training, the optimal accuracy for 4-convolutional layer simple CNN was achieved at 120x120 px sq. patch size, which resulted in an overall accuracy of 0.80 ± 0.04 (10-fold validation). The accuracies for individual artists for this particular simple CNN architecture is given in Table 3.1. These accuracies can also be visually observed in Fig. 3.2, which shows predicted labels

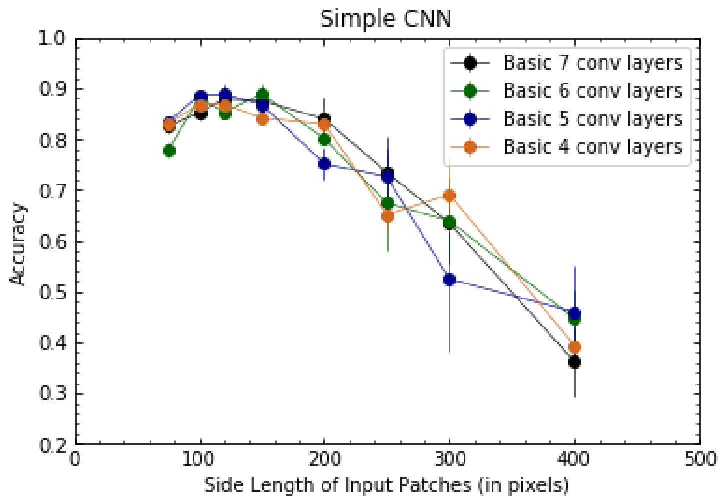

Figure 3.1: Overall network performance of simple CNNs (80% training) over patch size range.

by the network in one of the ten trials overlaid on corresponding profilometry image of the remaining painting.

As it can be seen from both Fig. 3.2 and Table 3.1, our classification result for simple CNN (4 convolutional layers) are not uniform across different artists. As we will show later in this section, this difference in artist-to-artist accuracy is actually a consequence of the fact that stylist differences exist among different artists.

It is noteworthy to mention that even with a simple four convolutional layer neural network, trained from scratch with only 8 paintings (out of 12), the network is capable of distinguishing artist styles in a 'complete' painting, as shown through the combined accuracy row in Table 3.1 or prediction labels in Fig. 3.2. The network performance extremely well for artist 2, and the performance is also relatively decent for the other artists, especially considering that the accuracy for random guessing would be merely 0.25.

The network testing accuracies are further improved using transfer learning, specifically VGG-16 network (Fig. 2.6) previously trained on ImageNet dataset ([4]) as described in section 2.3.1. Fig. 3.3 shows the comparison of overall accuracy of VGG-16 and a simple CNN (with 4 convolutional layers) when trained on $66.\bar{6}\%$ training data for a range of input patch sizes. As it can be seen, the network performance of VGG-16 is an improvement over the simple CNN for all patch sizes (within error bars). Additionally, the drop in accuracy at larger patch sizes is relatively smaller in VGG-16 network than

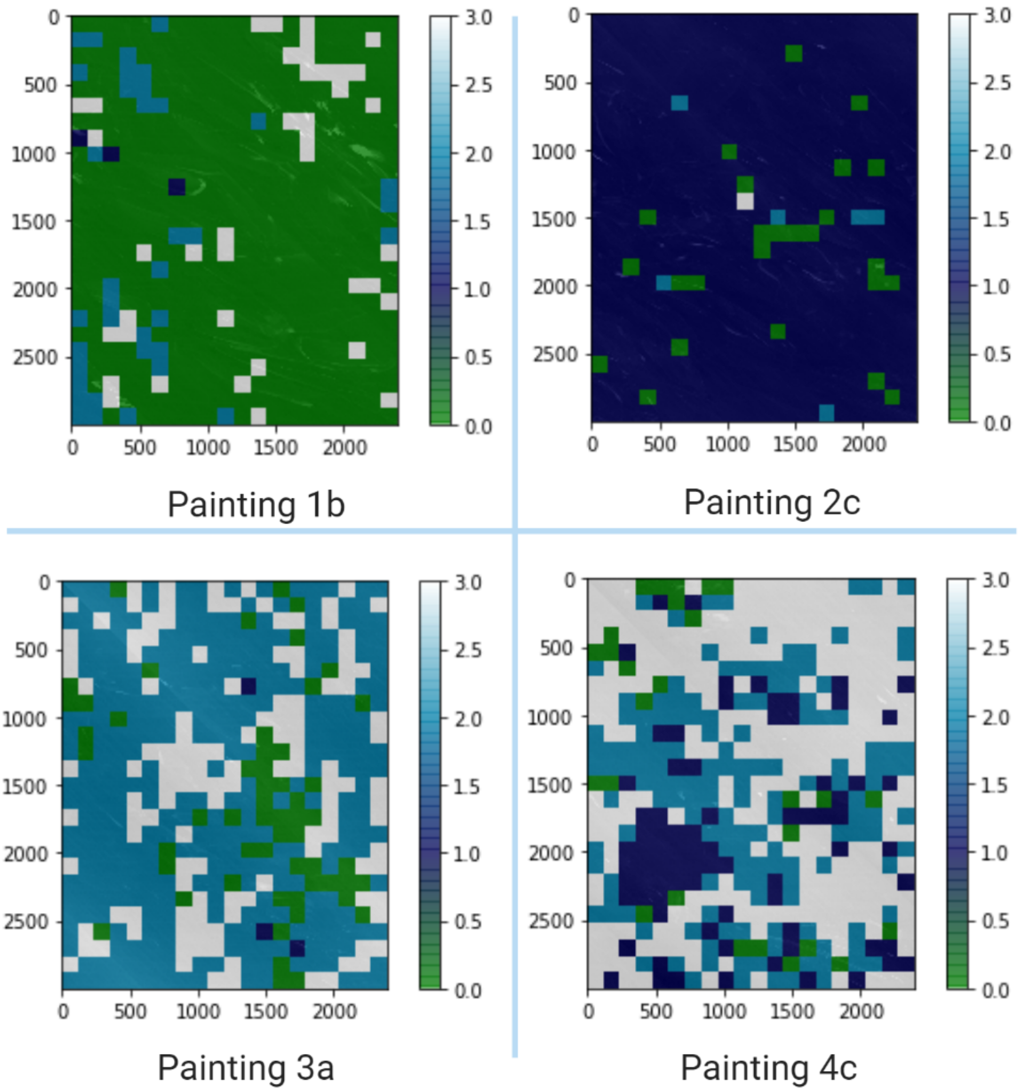

Figure 3.2: Simple CNN Prediction classes overlaid on testing images of the dataset.

in simple CNN, which shows a sharp decrease in performance for patches above 200x200 px sq. size.

Similar to our individual artist-based analysis for simple CNN, we computed prediction accuracies for the four artists separately for patch size of 120x120 px sq., as shown in Table 3.1 and Fig. 3.4. The individual accuracies still remain really high for artist 2, and are particularly improved for artist 1 and artist 3. While, we do observe an increase in identification of artist 4 for some training instances (Fig.3.4), with uncertainties, the performance almost remains unchanged (Table 3.1).

We argue that the difference in testing accuracies across our four artists is due to stylistic differences among them, hence we claim that the network is learning differences in brush-

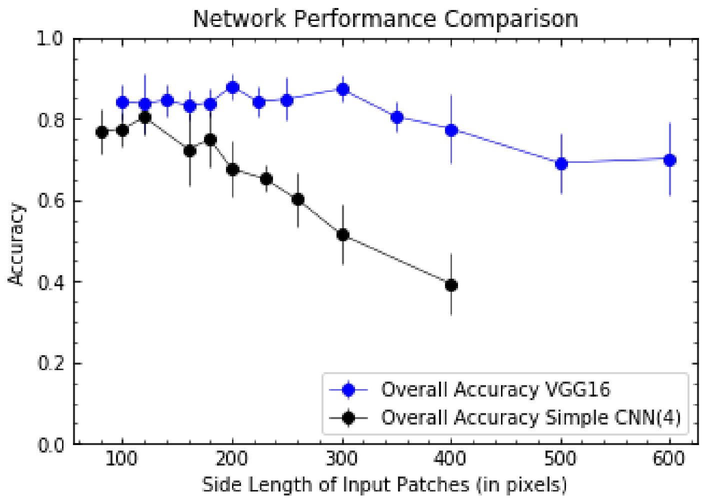

Figure 3.3: Overall network performance comparison for VGG-16 and Simple CNN with 4 convolutional layers (67% training) over patch size range. Units of patch size are in pixel square

Artist	Simple CNN Accuracy	VGG16 Accuracy
Artist 1	0.80 ± 0.07	0.88 ± 0.04
Artist 2	0.97 ± 0.02	0.95 ± 0.04
Artist 3	0.7 ± 0.1	0.8 ± 0.1
Artist 4	0.7 ± 0.1	0.7 ± 0.2
All Combined	0.80 ± 0.04	0.84 ± 0.07

Table 3.1: Overall and individual test accuracies for four artist classification using simple CNN (4 conv layers) and VGG-16 with patch size = 120x120 px sq. (n=10)

stroke style and not the content of the paintings. This is supported by the fact that the area of our patch size (120x120) is only 0.2% of the total profilometry scan, which is too small to be able to carry any significant content-specific information to influence machine learning. We also overlaid the network performance over high-resolution photographs of the paintings, where we found that there are no easily-discernible visual correlations between the network performance to common elements of the painting (shown in Appendix C).

An analysis of local brushstroke height variation further augments our claim that the network is being trained on the artist style. Thickness of brushstroke, brush wetness and release, and brushstroke evolution all comprise important stylistic techniques that differ from artist to artist. [11] We have used patch-wise standard deviation as a parameter to mathematically assess these brushstroke-related stylistic differences across artists. A

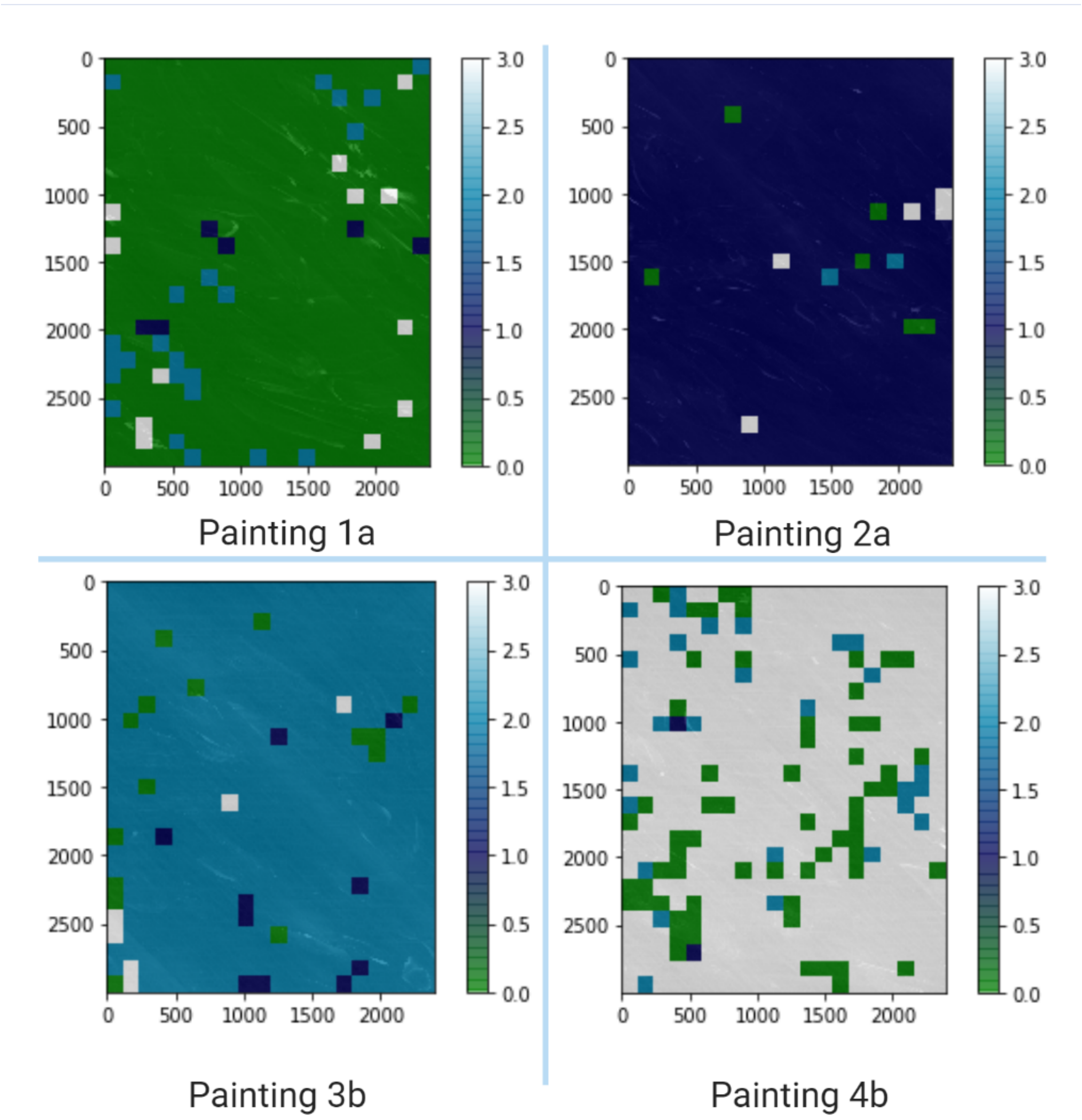

Figure 3.4: Transfer VGG16 prediction classes overlaid on testing images of the dataset.

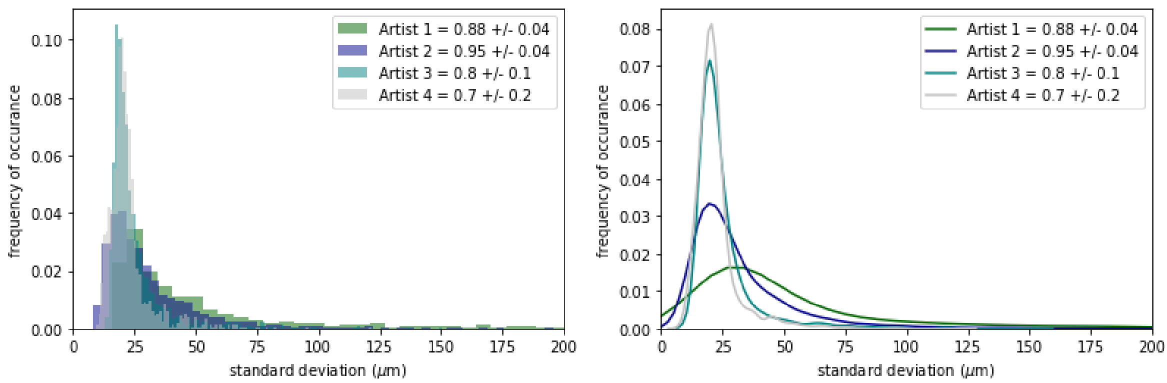

Figure 3.5: Left: Histograms of local standard deviation of height data in patches of size 120x120 px sq. The histogram is normalized for each artist separately such that area within the histogram equals unity. Right: Best fit traces of each of the histograms are computed using Seaborn package[19]. Legend show the corresponding artist accuracy using VGG-16 network

greater standard deviation value corresponds to an artist with greater brushstroke thickness and less amount of residual liquid that remains after the stroke of the brush.

Fig. 3.5 shows the histogram plot of standard deviation values for 120x120 px sq. patches extracted from each of the four artists. The right panel of Fig. 3.5 shows the respective best-fit traces of the histograms, which were estimated using the 'Seaborn' library in python. [19] The histograms are normalized such that the area inside each of the trace equals unity. As it can be seen from Fig. 3.5 (right panel), patches extracted from artists 3 and 4 paintings have mostly small height variations ($\sim$25 μm). While for artist 1, the standard deviation in patches occupies a relatively larger spread of values, including values as high as 200 μm. Artist 2, on the other hand, has an intermediate standard deviation distribution.

A physical interpretation of these distributions would be as follows. Artists 3 and 4 have stylistically smaller brushstroke thickness, such that most of the patches have small standard deviation in height values, possibly caused by the micro-scale height differences of the canvas. The greater spread of standard deviation in artist 1 patches, on the other hand, shows that their brushstroke style is relatively thicker. Artist 2 has an intermediate distribution, hence, they also have an intermediate stylistic technique vis-à-vis brushstroke thickness and wetness.

The legend presented in Fig. 3.5 also indicates the artist-specific accuracies for VGG-16 network. Since, the accuracies are relatively smaller for artists 3 and 4, who have thinner brushstroke style, we have argued that it is relatively more difficult to train a machine learning to identify and characterize these styles. In Fig. 3.6, we have separately plotted the histograms of standard deviation of patches that are correctly identified by one of the VGG-16 networks trained on 120x120 px sq. input patches. For this particular example network, the overall testing accuracy is 0.89, while artist-specific accuracies for artists 1-4 are 0.92, 0.99, 0.82, and 0.83 respectively (also provided in the inset of the sub-figures of Fig. 3.6). As it can be seen from the last sub-figure of Fig.3.6, majority of the patches which are incorrectly classified by the network fall have relatively smaller standard deviation ($\sim$25 μm). However, since we are already getting an overall accuracy

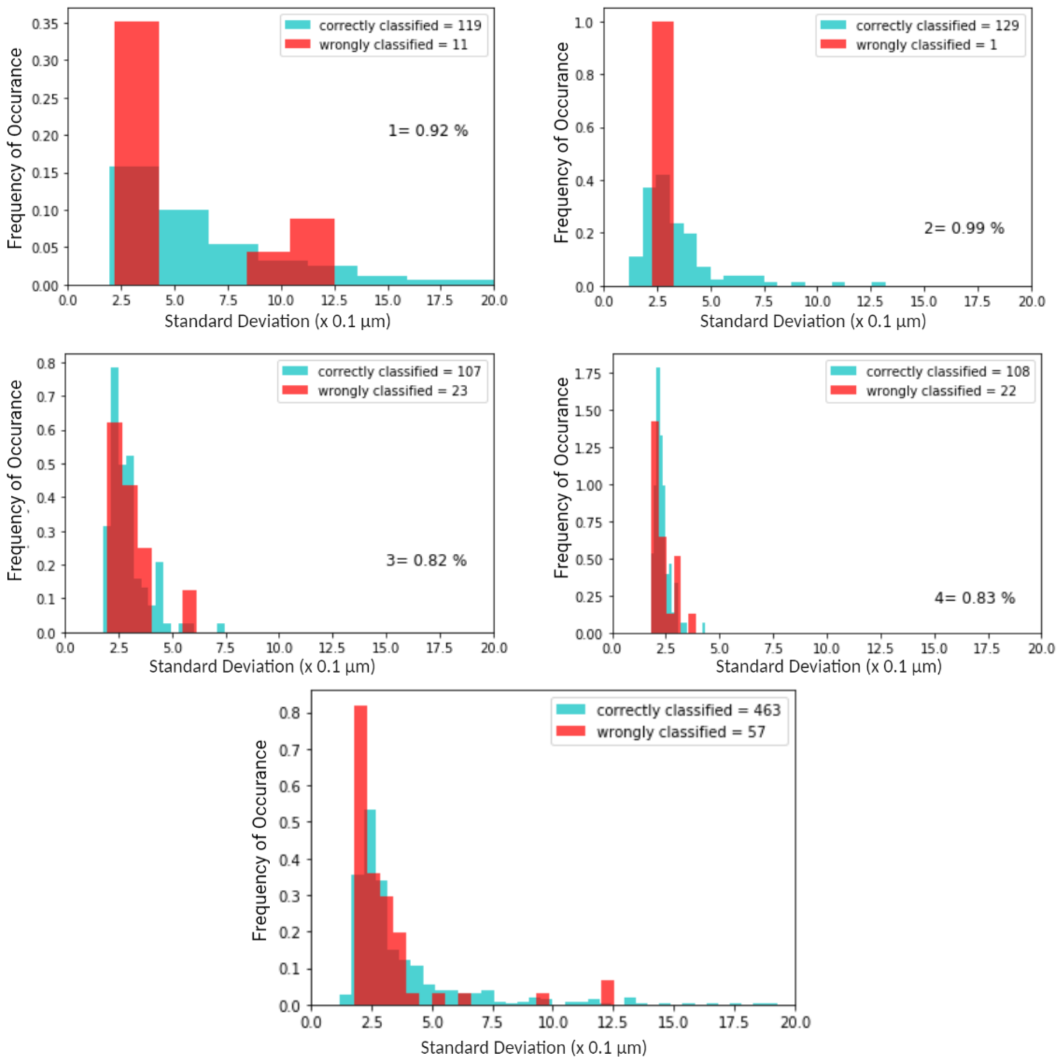

Figure 3.6: Top Four Subfigures: Histograms of local standard deviation for each of the four artists separated on the basis of whether VGG-16 network identified them correctly or not. The artist class is shown in the inset of each of the figures, with its corresponding accuracy. Bottom figure: Histograms of local standard deviation for all artists combined together, but again separated on the basis of accuracy in the prediction by the network. Note: the units for the standard deviation for the plots are (10^{-1} μm)

of $\sim$0.90, due to stochastic nature of machine learning, there are also some cases where the height variation is significantly large, but they are still being incorrectly classified. Thus, our analysis of local brushstroke variation shows that that our machine learning algorithms are able to distinguish artists based on their stylistic techniques, including factors such as thickness of brushstroke, brush wetness and release, and brushstroke evolution. The fact that some artists are more easily characterizable than others is itself an artifact of artist-to-artist technique differences. However, it is important to point that even when two artists have a significant overlap in one aspect of style, say brushstroke thickness, the machine learning is still able to learn other stylistic aspects to distinguish between them. For instance, as observed in Fig. 3.5, artists 3 and 4 have overlapping standard deviation histograms, a VGG-16-based training module is still able to achieve individual average test accuracy of 0.8 and 0.7 respectively, when trained on all four artist classes. However, to critically assess the stylistic characterization for different artists separately, it is important to analyze only two classes at a time. In the next section, we will specifically focus on identifying and separating target classes from outlier class to assess the success of the network in distinguishing between two classes of styles (or their combinations).

3.2 Single-Class Anomaly Detection

Multiple experiments were performed to re-train and analyze the performance of the VGG-16 network when it comes to separating between two different combination of stylistic classes, target-class and outlier class (Table 3.2). We decided to keep our input patch constant to 120x120 px sq. across all trainings. For each of the 12 paintings, we got a total of 500 patches (=(3000x2400)/(120x120)). In each of the experiments, we used equal number of patches of target-class and outlier class for training to balance the input classes. However, this number varied depending upon the number of available patches for a given experiment (Table 3.2). The remaining patches were mostly used for testing, that is, they were used to evaluate accuracy and F_1-measure of the network. In the last two experiments, we did not train the network on any of the patch from artist 4 and kept them as ‘external’ for outlier detection testing.

Label	Target-Class (# of Training Patches)	Outlier Class (# of Training Patches)	External Outclass (# of Test Patches)
12 & 34	Artists 1 and 2 (2400)	Artists 3 and 4 (2400)	None
1 & 234	Artist 1 (1350)	Artists 2, 3, and 4 (1350)	None
2 & 134	Artist 2 (1350)	Artists 1, 3, and 4 (1350)	None
3 & 124	Artist 3 (1350)	Artists 1, 2, and 4 (1350)	None
4 & 123	Artist 4 (1350)	Artists 1, 2, and 3 (1350)	None
1 & 23 (4)	Artist 1 (1200)	Artists 2 and 3 (1200)	Artist 4 (1500)
3 & 12 (4)	Artist 3 (1200)	Artists 1 and 2 (1200)	Artist 4 (1500)

Table 3.2: VGG-16-based experiments performed for single class anomaly detection. For each case the network was trained on input patch size of 120x120 px sq. The target class and outlier classes are defined for each of the case. The table also reports the corresponding number of training patches. In the final two experiments, external class of artist 4 was used, where none of the patches from artist 4 were used to train the network, but they were still used for testing the network, i.e. testing whether the network can flag-out artist 4 as an outlier class.

In our first experiment, we wanted to test whether we are able to train the network distinguish the brushstroke thickness-related stylistic difference we found in artists 3 and 4 in the previous section from artists 1 and 2. As shown in the first row of Table 3.3, the network is able to distinguish between the two classes with average accuracy and F_1-score of 0.96.[1]. In the next set of experiments, we kept each of the four artists as our target artist (one at a time) and attempted to distinguish their style from all other styles. Our accuracies for all the cases again turned out to be greater than 0.90, with artist 2 again resulting in the highest classification accuracy (average being 0.97). The F_1-scores for these experiments resulted in even higher scores (greater than 0.95 on average) as they fixed the test class imbalance (150 for in-class and 3150 for outlier class).

However, in all of the experiments described hitherto, we explicitly trained the machine learning algorithms on what the style of ‘outlier’ class looks like. This may not be possible in real-life cases, where little to no information might be available regarding the characteristic style of outlier class. To mimic such situations, in our last two experiments, we did not train/validate our network with any of the patches extracted from artist 4 paintings. In the first case, we set up our system to distinguish outlier artist 4 from target class artist

[1]Accuracy and F_1-score in this case are expected to be the same since there is no class imbalance in the testing set (600 patches for both classes)

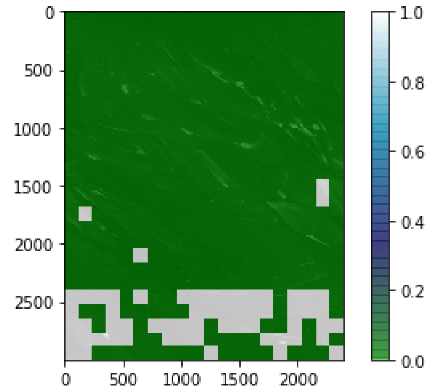

Figure 3.7: Visualization of Single-Class Anomaly Experiment: VGG-16 based network is tested for an example forged painting, where the last 100 patches of the in-class test patches (artist 1, labeled with 0 or color green) are replaced with those of external class (artist 4, labeled with 1 or color gray) to mimic an actual forgery case.

1, when patches from artist 3 were already present in the training data as outlier class. Since we have already shown that artist 3 and 4 have greater similarities in their styles than artist 3 and 1, we expected high accuracy and F_1-score for this problem (significantly greater than 0.5). As shown in Table 3.3, our the measurements for an average of 10 trial were 0.82 and 0.89 respectively.

In the second problem, however, we chose artist 3 as our target class and kept artist 4 as the external outlier class. This is a relatively more difficult than the previous one on its face value due to style-based similarity in artist 3 and 4. The performance evaluation parameters are reported in Table 3.3, with average accuracy being 0.70, while average F_1-score being 0.80. Therefore, although the performance decreased when distinguishing styles of artists 3 and 4, than those of artists 1 and 4, an F_1-score of 0.80 still indicates that majority of the time artist 4 is predicted as an outlier class.

Experiment	Accuracy	F_1-measure
12 and 34	0.96 ± 0.01	0.96 ± 0.01
1 and 234	0.92 ± 0.02	0.96 ± 0.01
2 and 134	0.97 ± 0.01	0.98 ± 0.01
3 and 124	0.92 ± 0.05	0.95 ± 0.03
4 and 123	0.94 ± 0.01	0.97 ± 0.01
1 and 23 (4)	0.82 ± 0.03	0.89 ± 0.02
3 and 12 (4)	0.70 ± 0.04	0.80 ± 0.03

Table 3.3: Performance of VGG-16-based outlier detection (n=10) (for 120x120 px sq. input patch)

This result can also be inferred from visualization of one of the test paintings in Fig. 3.7. In this case, we basically simulated an example forgery case, where we used replaced the last 100 patches in one of the artist 1 paintings with last 100 patches from one of the paintings from artist 4. This is an example case from our penultimate experiment in Table 3.2. In-class and outlier patches are labeled as 0 and 1 respectively. Fig. 3.7 shows that the network is able to flag-out majority of patches from artist 4 as being an outlier class. This is an incredibly fascinating result considering we never trained the network on any of the patches from artist 4 paintings to be an outlier class. This identification was solely based on the fact that there was a group of outlier class (patches from artists 2 and 3) and the network learned features that separated the collective styles of those of outlier class from the style of artist 1.

Chapter 4

Discussion

The network performance for the different architectures tested for our four artist classification problem show that it is possible to classify artists based on their styles. The maximum overall accuracy for a simple CNN and a transfer learning-based VGG-16 network are evaluated to be 0.80 ± 0.07 (n=10) and 0.88 ± 0.03 (n=10) respectively, which are both significantly greater than the accuracy of random prediction (0.25). However, these maximum overall accuracy are obtained at different input patch sizes for the two networks – 120x120 px sq. for a simple CNN network and 224x224 px sq. for the VGG-16 network (Fig. 3.3). Therefore, the training performance is not homogeneous, and it depends on multiple aspects, as we will discuss in this chapter.

Firstly, as observed in Fig.3.3, the performance of both the classification network dampens at both small and large patch sizes. At small patch sizes, the accuracy is reduced since the patches are so small such that they do not include relevant brushstroke style-related features and information. While we were not able to go below the patch size of 80x80 px sq. (due to memory constraints), the accuracy at 80x80 and 100x100 are always smaller than that at 120x120 px sq. because of lack of features available at sizes smaller than 120x120.

At larger patch sizes, on the other hand, we argue that the training performance is dampened due to lack of sufficient data needed to train the network. Amount of data is always a critical factor in machine learning and becomes an even more important factor for deep learning modules, since the number of parameters to train are significantly higher in deep learning than traditional machine learning problems. For instance, for a patch size of 800x800 px sq., we only have around 11 total patches per painting. If we consider $66.\bar{6}\%$ training data, as we used for classification network, we only have 88 (=11x8) total patches to train on, which is clearly not sufficient compared to training data size used in general deep learning training modules. Therefore, amount of training data is a critical factor that determines the success of a training module. We also see that the uncertainties increase at larger patch sizes (Fig. 3.3). This is also due to insufficient data issues, as fewer number of training patches correspond with greater stochasticity in the training progress.

Another important factor that influences the performance of style learning is related to the inherent differences in artistic styles. As we showed in our analysis for local height variation, the success of our classification network is different for different artist styles, i.e. some stylistic features are easy to characterize than others. In our case, we achieved almost perfect accuracy for artist 2 every time we tried to characterize artist 2's style. While the performance for artist 4 was the weakest on most occasions. We were able to relate this variation in artist-specific accuracies with histogram for local standard deviation, which corresponds to features like brushstroke thickness, brush wetness and release, and brushstroke evolution. However, these are just some of the elements together make up artistic style, that we were able to quantitatively characterize in this work.

In real-life scenarios of identifying cases of art forgery, while we may find these quantitative tools helpful in distinguishing an artist's style from another, we can not rely on certain mathematical operations to separate style from content, as it would likely fail on many occasions (for instance, distinguishing artist 3 and 4 using local standard deviation). In fact that is the reason why we used deep learning in the first place, as it is capable of finding features and representations that are not easily perceivable with known arithmetical operations.

In our second problem of single-class anomaly detection, we showed that deep learning can successfully overcome this challenge and separate out classes that might appear to be similar in one form of analysis. When we tested our VGG-16 network to separate out style of artist 4 from target class artist 3, we got an average F1-score of 0.80. In this case, the basic idea was to group together and train on an outlier class (consisting of styles of artists 1 and 2), such that any other external artist will have more features in common with the grouped outlier class, than a specific target artist class. Hence, a transferable approach in detecting art forgery could be to have a pool of outlier class as part of the training module, and then any other external patch that does not belong to the target artist (whose style is being copied) will be flagged out.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We hypothesized that variations in brushstroke styles and patterns can be used to 'teach' an AI algorithm to distinguish between distinguish artistic styles. A comparison between the network performance of different artist classification architectures supports our hypothesis, where we have demonstrated overall accuracies of above 80% for classification of small image tiles. For majority of our analysis, we chose the size of image tiles to be 120x120 px sq., which resulted in an optimum overall classification accuracy for our simple CNN architecture of 0.80 ± 0.07. This patch size was also small enough ($0.08\bar{8}\%$ of the total painting) such that it increased our prospect of training the machine learning on artistic style and not artistic content. This was further confirmed using analysis of local patch variation, as it showed that our training performance was dependent on features that directly correspond to elements of artistic styles, particularly that of brushstroke thickness.

Our work in this project also adds to the growing literature related to style characterization, as we have shown that it is possible to classify brushstroke style of any general artist based off of only height data, which had not been shown until now. This aspect of being able to identify and categorize artistic styles clearly has direct applications to our motivation towards identifying cases of forgeries.

In the final part of our work, we have developed and proposed a methodology that can

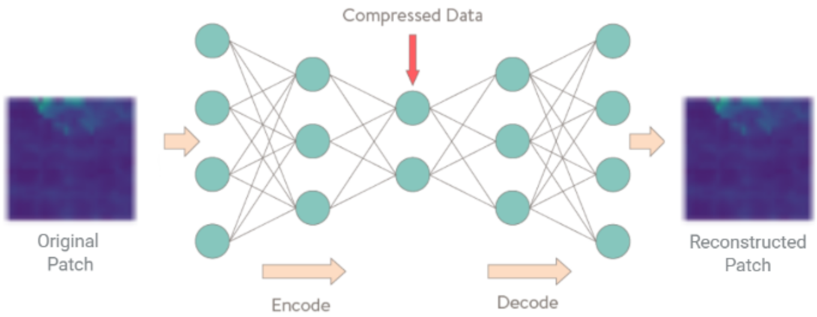

Figure 5.1: Autoencoder-Based Network

directly be translated for applications in the art world. This involves redefining the problem as a single-class outlier detection problem, where the network is trained on a particular target class as in-class and a group of paintings from other artists as outlier class. Any other outlier patch will then be more likely to have more commonalities with the collective outlier class, than the specific target class.

5.2 Future Work

In single-class anomaly detection, ideally only a single class (in our case, target artist class) is involved in training. Recently, Borghesi et al. described an approach of using autoencoders to detect anomalies. Autoencoder is a deep learning based neural network, where the task is to reconstruct the image with a dimensionally-reduced feature space (Fig.5.1). The network structure has two main parts - encoder and decoder. The encoder part of the network 'encodes' the style-related features of the painting data. While, the decoder part of the network is 'up-convoluted' such that final layer being the same size as the input patch size. The loss function in this case is simply the mean square error (mse) between the input and the reconstructed patches. As proposed by Borghesi et al., the stylistic outliers are then separated from the in-class patches by comparing the performance of the autoencoder network. [2] It is hypothesized that if the network is only trained on a single class, then the error in the reconstructed images of in-class and out-of-class will be different and can be used as a threshold for detecting stylistic anomalies.

[2] While we attempted to use this approach to develop an outlier detection methodology that does not require training of any outlier class, we were unable to find a significant stylistic threshold due to reconstruction errors being dominated by minute features in canvas and not in artistic style. Therefore, in future, we plan to use decomposition techniques to separate the canvas height variation from brushstroke levels, before training the autoencoder network.

In near future, we are also planning to test our methodologies on real-cases of art history. In particular, we will be examining specific painting contribution in El Greco's *Baptism of Christ*. It is known that the painting was finished by his son Jorge Manuel Theotocópuli. However, the actual are still a topic of debate in the field of art history.[12]

5.3 Code Availablility

The code is made available in the following Github repository:

`https://github.com/gundeep15/ML_Art_Project`

Bibliography

[1] ArtnetNews. (2014). Over 50 Percent of Art is Fake, October 13, 2014, Art and Law, url: https://news.artnet.com/market/over-50-percent-of-art-is-fake-130821.

[2] Borghesi, A., Bartolini, A., Lombardi, M., Milano, M., and Benini, L. (2018). Anomaly detection using autoencoders in high performance computing systems. arXiv preprint arXiv:1811.05269.

[3] Carriquiry, A., Hofmann, H., Tai, X. H., and VanderPlas, S. (2019), Machine learning in forensic applications. Significance, 16: 29-35.

[4] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR).

[5] Denis, D. (2009). The art instinct: Beauty, pleasure, human evolution. Oxford University Press, USA.

[6] Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. Nature Communications.

[7] Instuments, N. Optical Profilometry, Nanoscience.com, url: https://www.nanoscience.com/techniques/optical-profilometry/.

[8] Larson, N. Fine Arts Experts Institute: Lab sleuths in Geneva help art world uncover fakes, ND, ArtDaily, url: https://artdaily.cc/news/73562/Fine-Arts-Experts-Institute–Lab-sleuths-in-Geneva-help-art-world-uncover-fakes.

[9] Lecoutre, A., Negrevergne, B., and Yger, F. (2017). Recognizing Art Style Automatically in painting with deep learning, Proceedings of Machine Learning Research, vol. 77, pp. 327–342.

[10] Li, D. (2017). Automated Real-Time Evolution Monitoring of Drying Paint Morphology, Nanovea, url: https://nanovea.com/App-Notes/paint-drying-morphology.pdf.

[11] MacEvoy, B. (2015). The Brush and Brushstrokes https://www.handprint.com/HP/WCL/tech18.html.

[12] Maré, E. A. (2008). El Greco's Representation of Mystical Ecstacy. Acta Theologica Supplementum 11. http://www.scielo.org.za/pdf/at/v28s11/06.pdf.

[13] Olivas, E. S., Guerrero, J. D. M., Sober, M. M., Benedito, J. R. M., and Lopez, A. J. S. *Chapter 11, Transfer Learning, Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes.* Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.

[14] Putri, T. (2019). Artistic Style Characterization of Vincent van Gogh's Brushstrokes Doctoral Thesis. http://fs.unm.edu/vanGogh-NeutrosophicSegmentation.pdf.

[15] Putri, T., Mukundan, R., and Neshatian, K. (2017). Artistic Style Characterization of Vincent Van Gogh s Paintings using Extracted Features from Visible Brush Strokes. In ICPRAM (pp. 378-385).

[16] Rosseau, T. (1968). The stylistic detection of forgeries. In The Metropolitan Museum of Art Bulletin, volume 26, 6, pages 247–252. The Metropolitan Museum of Art.

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

[18] Torrey, L., and Shavlik, J. (2009). Transfer learning. IGI Global.

[19] Waskom, M. (2012). Seaborn: Statistical Data Visualization (0.10.0) https://seaborn.pydata.org/index.html.

[20] Zachos, E. How More Than Half the Art in This French Museum Was Forged, May 1, 2018, National Geographic, url: https://www.nationalgeographic.com/news/2018/05/fake-art-france-culture-spd/.

[21] Zang, Y., Huang, H., and Li, C.-F. (2013). Stroke style analysis for painterly rendering. In Journal of Computer Science and Technology, volume 28, 5, pages 762–775. IEEE.

Appendix A

Confusion Matrix and F1-score for Binary Classification

Performance of a binary classification network in machine learning is evaluated using multiple different measures (accuracy, precision, recall, F1-score, etc.), depending upon the system at hand. However, a common step involved in computing all the different measurements is writing down the confusion matrix for the system. An example of what a confusion matrix looks like for a binary system is shown below:

		Predicted Class		
		In-Class	Outlier	Total
True Class	In-Class	True Pos (TP)	False Neg (FN)	TP + FN
	Outlier	False Pos (FP)	True Neg (TN)	FP + TN
	Total	TP + FP	FN + TN	N

In a binary classification case, there are four possible cases that can occur for each test patch, given by the terminology below:

- True Positive (TP) - In-class patch is classified as in-class
- True Negative (TN) - Outlier patch is classified as an outlier
- False Positive (FP) - Outlier patch is classified as in-class
- False Negative (FN) - In-class patch is classified as an outlier

A common model evaluation parameter is accuracy (which we have used in this paper):

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

However, this parameter isn't an effective measure of the model performance in cases when the testing classes are imbalanced, i.e. there are more instances of in-class test patches than those that are outliers (or vice-versa). For example, consider a case where the in-class test patches are outnumbered by outlier patches (as is the case in some of our VGG-16 based outlier detection experiments). Say, 80 outlier and 20 in-class. Then if we have a network that is perfectly skewed towards the outlier class, i.e. predicts outlier for all input, then, it would give an accuracy of $= 80/100 = 80\%$, which is too good to be true for a perfectly skewed network. Therefore, to fix the class imbalance, we introduce other model evaluation parameters, namely precision and recall, which are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

For a perfect network, we want both precision and recall to be equal to 1. In our example, the measure of precision is not defined (0/0), while the recall measure is 0. On the other hand, if the network was perfectly skewed towards in-class, then precision would be 0.2, and the recall is 1. Neither of the skewed networks are 'good', but in one case we got recall as 0 and in other, 1. Therefore, goodness of the network is defined by combining these two parameters using the F1-score. In this case, both precision and recall are weighted equally, and a harmonic mean of the two is obtained. Finally, the expression for F1-score is:

$$\mathrm{F}_1 = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Appendix B

Data Preprocessing

Since machine learning algorithms are employed to find common patterns and representations, it is important that the training data is consistent. For instance, we need to remove the data offset from each of the patch such that the algorithm does not learn on the offset, but rather the pattern within the local scope of the patch. A common example to justify the need for data processing is that in image recognition and classification of, say, cat and dog, the classification should be invariant of the amount of brightness of the input photograph. Otherwise, there is a much greater chance that the learned features would represent classification based on image brightness, and not anything related to the physical features of cats and dogs.

In our case, data processing is a critical step of the workflow, since we want to train on local spatial patterns in brushstroke style and not the content of the patches. Due to the poorly-predictable nature of machine learning, generally, different techniques are tried and implemented until a set of processing steps are found that highlight the aspects in the patches necessary to successfully train the network. As described in the methods section, we implemented multiple data preprocessing techniques. The performance each of the techniques for transfer learning VGG-16 network are reported in table below:

Preprocessing Technique	Overall Accuracy (120 px patch)
Min-Max Scalar	0.36 ± 0.03
Univariate Standardization	0.71 ± 0.06
Narrowing the range by a factor of 10	0.84 ± 0.07

Appendix C

Style, Not Content

In this appendix, we have attempted to further show that our four-artist classification neural network are learning based on their stylistic differences and not due to the differences in physical content of the paintings. Figure C.1 shows the individual artist testing accuracies for our two networks - Simple CNN and transfer learning based VGG-16 network. As seen in the figure, the performance accuracies follow similar trend across all patch sizes (within error bars) as shown in the results section for 120x120 px sq input patches. Since these differences in network performance for different artist are consistent over a range of patch sizes, this shows that these accuracy differences are due to inherent differences in painting styles of the artists, and not an artifact of training setup or content of the painting.

In Fig. C.2 we have also shown labels for network predictions for testing patches for each

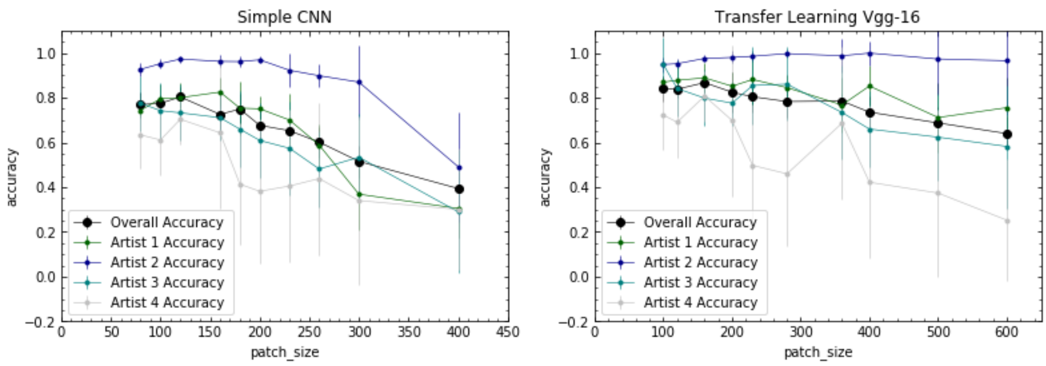

Figure C.1: Overall and individual artist performance for both simple CNN and VGG-16 networks. Units of patch size length is in pixels.

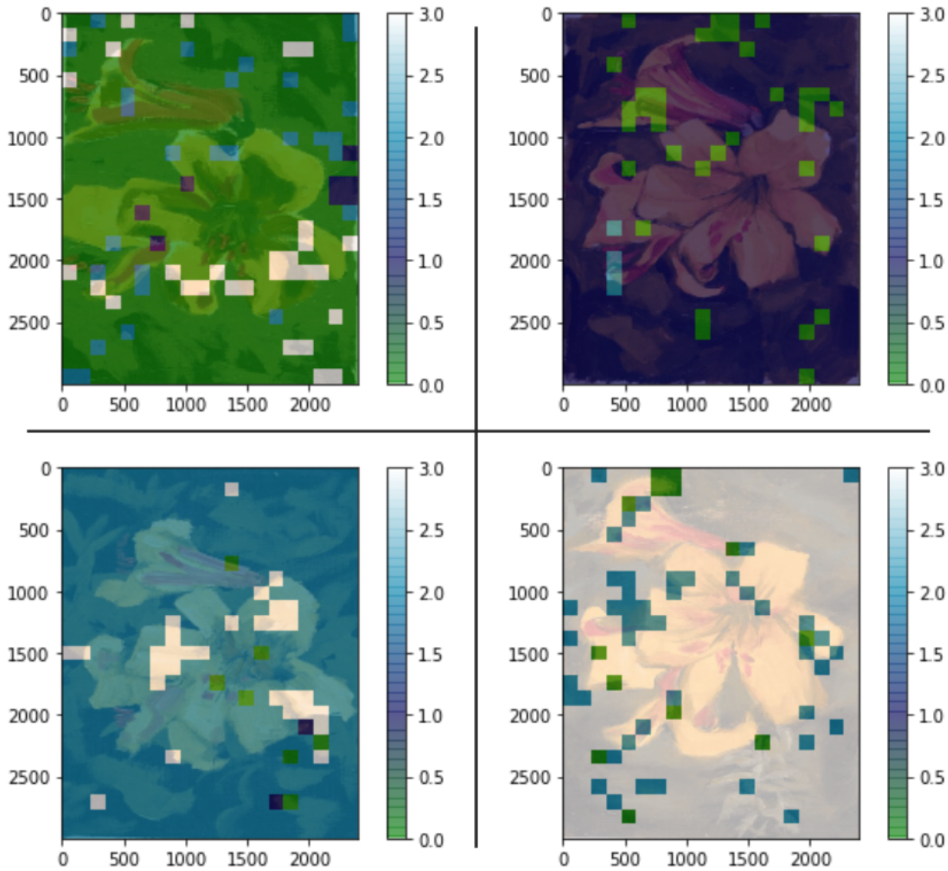

Figure C.2: Labels for network predictions for testing patches for each of the four artists overlaid on the actual high-resolution photographs (CW: 1a, 2b, 3c, 4a)

of the four artists overlaid on the actual high-resolution photographs of those paintings. The testing images in this particular case were paintings 1a, 2b, 3c, 4a. Through Fig C.2, we look at qualitative correlations between instances the network is predicting incorrectly to physical features of the paintings, like edges, curvatures, colors, and other elements of the water lily painted by the four artists. However, as we can see in the figure, the network does not appear to be skewed toward any particular aspect of the paining. For instance, there are almost as many incorrect predictions for the background as there are for the foreground. While, this analysis isn't as concrete as other analyses discussed in the paper, it is still helpful in showing that the correlation between stylistic features (like brushstroke thickness) to accuracies is much more easily discernible that that between painting content and accuracies. Therefore, it further strengthens our claim that our machine learning network are learning based on style and not content.