Classification of painting style with transfer learning

Emiel Oomen
ANR: 1247723

Thesis Submitted

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Communication and Information Sciences,

Master Track Data Science: Cognitive Science and Artificial Intelligence,

at the School of Humanities and Digital Science

of Tilburg University

Thesis committee:

Supervisor: Prof. Dr. Eric Postma

Second reader: Dr. E.A. Keuleers

Tilburg University

School of Humanities and Digital Science

Tilburg, The Netherlands

July, 2018

Preface

In front of you lies the thesis: “Classification of painting style with transfer learning”. It has been written as partial fulfillment of the requirements for the Cognitive Science and Artificial Intelligence at Tilburg University. I was engaged in researching and writing this thesis from February 2018 to August 2018. My research question was formulated together with my supervisor at Tilburg University, Prof. Dr. Eric Postma. I would like to thank my supervisor for his excellent guidance and support during this process, without his advice to keep things simple I would still be running algorithms. I would also like to thank my family and friends, especially my mother for being patient and believing in me.

Emiel Oomen

30-07-2008

Abstract

With the availability of large paintings datasets and advances in computer vision techniques a large number of researchers started investigating paintings in new ways. While previously paintings where predominantly studied for forgery detection, new studies investigated different painting categories such as genre and style. Most of these studies started using Convolutional Neural Networks to classify these categories. While CNNs have been successfully applied to object detection in natural image tasks, several studies investigated the possibility of utilizing these networks for painting classification. The studies showed that CNNs attain the best performance by either using the features of different layers of a pre-trained CNN or fine-tuning the CNN for the task of painting classification, while training a CNN from scratch did not give good results (Tan et al. 2016; Lecoutre, Negrevergne, & Yger, 2017; Balakrishan, Rosston, & Tang, 2017; Kedia, 2017). However, these studies did not investigate to what extent the features of different layers of different pre-trained CNNs are able to complement each other. This thesis investigates if concatenating different layers of a pre-trained VGG-19 and Resnet-50 complement each other by increasing the accuracy for the task of painting style classification in comparison to using a single feature vector. The thesis uses the Wikiart-dataset and shows that concatenating feature vectors of different pre-trained CNNs that are both trained on the same object-detection task, does not yield better results than are obtained with a single feature vector. However, concatenating feature vectors of CNNs that are trained on different tasks does improve accuracy. Interestingly, the results of previous studies can be improved by concatenating the feature vector of the fine-tuned feature vector with a feature vector that has not been fine-tuned. Therefore, future research should focus on training CNNs in combination with multitask learning to extract more information from the paintings.

Contents

Section 1: Introduction

This section discusses the motivation for the thesis in 1.1. The next part of this section (1.2) introduces related work and discusses the main model of this thesis, more specifically Convolutional Neural Networks and ends with the main research questions and sub-questions. The last part of this section gives an outline of the thesis.

1.1 Motivation

Nowadays vast amounts of paintings get digitized and become publicly available on the internet[1]. The digitization of these paintings together with advancements in computer vision techniques facilitates the automatic classification of these works of art. While most of these digital collections are already annotated by associated metadata, these annotated paintings can be used to create datasets. These datasets can be used to train models that are capable of automatically classifying unannotated paintings into classes such as painter, style, genre and date. By genre of a painting we refer to the content or theme of the painting, for example landscape, portrait, cityscape, still life and religious painting. Style is defined as "...a distinctive manner which permits the grouping of works into related categories"(Fernie, 1995) Examples of styles are: Baroque, Romanticism, Rococo, Primitivism, Ukiyo-e and Abstract Expressionism.

There are four applications for such automatic classifiers. The first application is that the automatic classification of paintings could increase the speed of the digitization process by automatically annotating digitized unannotated pictures and thereby enlarging current datasets. While this cannot be a goal in itself, the availability of a large dataset is a necessary condition for most classification algorithms, especially deep neural networks. This also demonstrated by the tremendous progress made in natural image classification after the introduction of the ImageNet-dataset in 2009 by Fei-Fei, Deng & Li and different successful models that resulted from this dataset (Krizhevsky, Alex, Sutskever, & Hinton, 2012; Simonyan, Karen, & Zisserman, 2014; Szegedy, Christian et al., 2015).

The second application is that classifying the author and date of paintings can be used to assist art experts or historians to detect forgeries (Polatkan et al., 2009). Additionally, both professionals can utilize models that are able to detect objects, scenes and moods in paintings to instantaneously analyze large amounts of artworks on different statistics (Crowley, & Zisserman, 2014a; Crowley, & Zisserman, 2014b). These same models can be used to annotate and archive the paintings more specifically, making it possible to retrieve paintings with different requirements (e.g. paintings of 'cat playing') (Mensink, & Van Gemert, 2014).

The third application is recommendation systems for art enthusiasts. Such systems suggest artworks to enthusiasts based on their history of choices of artworks. Moreover, art gallery visitors could

[1] E.g. https://www.wikiart.org/, http://arkyves.org/, https://www.artsy.net/, https://www.behance.net/, https://www.artnet.com/ and https://artuk.org/.

use these models on their mobile devices to classify artworks or regions of attribution in artworks that are created by different artists (Noord, Hendriks, & Postma 2015).

The fourth and last application is *style transfer* (Gatys, Ecker & Betge, 2015) which can be used for the creation of new artworks by transferring the style of a painting to a natural image or other paintings.

1.2 Convolutional Neural Networks for Art Classification

The classification of paintings has different applications, but is according Tan et al., a more challenging problem than standard classification tasks such as recognizing objects, sceneries and architecture in natural images (2016). [One could argue that classifying objects such as dogs and cats in paintings is certainly harder than classifying the same categories in natural images, since the dog/cat could be portrayed more creatively within paintings. Figure 1 illustrates the large variety of portraying buildings within different styles. Classifying 'buildings' is harder in paintings but does this hold for the classification of style or artist, since the signature of both categories is possibly distributed through the whole the painting. Another problem is that critical indicators such as color and texture are highly prone to variations in the digitization process (Polatkan et al., 2009). While the latter is also problematic for natural images, contours of objects are much more reliable features for those types of images. Until recently it was hard to compare both classification tasks, since a dataset of sufficient size for the classification of paintings was missing[2]. In 2014 however, the Wikiart paintings dataset, which consists

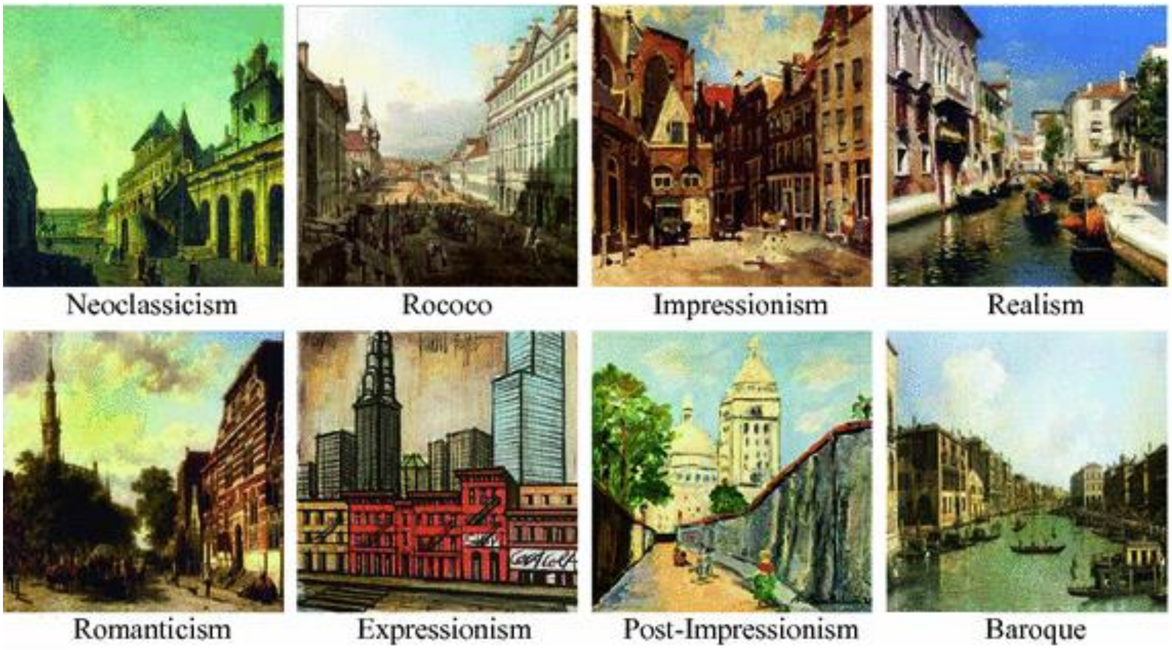


Figure 1. Paintings portraying buildings within different styles. Captions are the styles of the pictures. Reprinted from "Fine-Art Painting Classification via Two-Channel Deep Residual Network," by X. Huang, S. Zhong, and Z. Xiao, 2017, Pacific Rim Conference on Multimedia, p. 84. Copyright 2017 by Springer.

[2] Examples of object retrieval datasets are PASCAL VOC, CIFAR-10/100, Places-dataset and the already mentioned ImageNet dataset.

of 81,449 paintings was created by Karayev, while previous datasets contained a maximum of around 8,000 paintings (Johnson et al., 2008; Zujovic et al., 2009; Shamir et al., 2010; Khan et al., 2014)

Classification of paintings is not impossible with smaller datasets, but the size of the dataset restricts the type of models that can be used, since complex models are likely to overfit on small datasets (LeCun, Bengio & Hinton, 2015). Most of the studies before 2015 used low-level handcrafted features, such as color, shades, texture or brush strokes and edges in contrast to higher- or semantic-level features (Sablatnig, Kammerer, & Zolda, 1998; Keren, 2002; Khan, van de Weijer, & Vanrell, 2010; Carneiro, da Silva, Del Bue, & Costeira, 2012). Stork describes several of the most frequent techniques used about 10 years ago (2009). These techniques were are also referred to as *feature engineering* techniques, since they are ‘engineered’ by domain experts which decide which features are meaningful for certain tasks (Bengio, Courville, & Vincent, 2014). The studies performed around 2010 focused on classification of one or a small set of painters. The task was also known as *author attribution*, which is most dominantly utilized for authentication (Lombardi et al., 2005). Combinations of the extracted engineered features are represented as vectors and are used by classifying algorithms such as Support Vector Machines, Multiple Kernel Learning, k-Nearest Neighbor and Decision Trees (Zujovic et al., 2009).

With the introduction of the Wikiart-dataset and advances in deep learning techniques, a considerable amount of studies switched from using solely feature engineering to representation learning (RL) or a combination of both. RL relies on Neural Networks (NN), more specifically Convolutional Neural Networks (CNN) (LeCun, & Bengio, 1995). RL differs from feature engineering in that it does not use the predefined notion of a domain expert, but tries to learn the proper representations using the raw data and is thereby able to capture larger amounts of variation of the data. The larger amount of variation together with the larger dataset results in the accurate classification of in a larger amount of classes (e.g. styles, genres).

CNNs achieve to capture larger variations of data by using convolutional layers and pooling operations. A convolution layer consists of convolutional filters. A convolutional filter is defined by a matrix of a certain size, e.g., 3 x 3 or 5 x 5 filter coefficients, that is applied to every part of the image by means of a convolution operation. Figure 2 shows an example of a simple 3 x 3 matrix of coefficients that defines an "edge filter". For each image position, the filter is overlaid on an image region of 3 x 3 pixels, centered on the target pixel. The sum of the 3 x 3 multiplied pixel values and coefficient values defines the convolution value associated with the target pixel. Applying the filter to all image positions yields a convolution image. The result of applying the convolution filter shown at the top of the figure to the image of the house on the left, results in a convolution image shown on the right. This image indicates where the edges are in the picture, irrespective of their position in the image. The same filter could indicate edges of cars, boats and different kinds of houses, so it could detect edges of a large variety of data. In practice CNNs learn the coefficients (or weights), of the filters by using a gradient descent algorithm, such as backpropagation. By stacking several convolutional layers on top of each other the filters start to cover regions increasing in size and thereby start to respond to more complex features of the image. So, whereas the lower layers in CNN are able to detect edges and contours, higher layers are able to detect parts of the face, such as eyebrows, nose and mouth. In addition to stacking several layers of convolutional filters, a pooling operation is also used to increase the filter size. Pooling operations apply a window of a certain size e.g. 2 x 2 to an image and return the maximum (in "max pooling"), or average (in "average pooling") value of the window resulting in a smaller image.

-1	-1	-1
-1	8	-1
-1	-1	-1

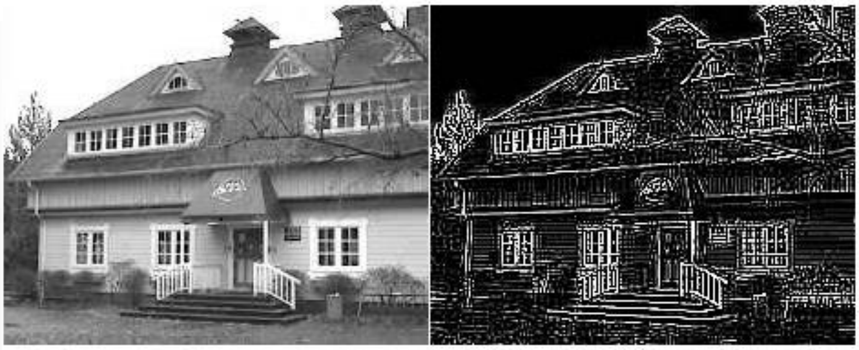


Figure 2. Top: A 3 x 3 filter containing 9 coefficients. Bottom left: Applying the filter to the image of a house yields the convolution image shown on the bottom right. Retrieved from: http://aishack.in/static/img/tut/conv-edge-detection-result.jpg

The different levels of the CNN extract features that support the classification task at hand. The lower layers extract features that are more generic, whereas the higher layers extract features that are more specific to the task. As a consequence, lower-level filters of a CNN trained on pictures of boats, cars and bikes can also be used to classify other objects. The process of using the weights of a trained CNN for the classification of a different task is called *transfer learning* (Pan, & Yang, 2010). Several papers have already shown that transfer learning is possible for natural image classification tasks (Oquab et al., 2013; Donahue et al, 2013; Razavian, Azizpour, Sullivan, & Carlsson, 2014; Yosinski, Clune, Bengio, Lipson, 2014). Moreover, several papers have examined the prospect of using transfer learning within painting classification tasks and demonstrated the possibility of using pre-trained CNN’s trained on natural image classification tasks for painting style and genre classification tasks (Bar et al., 2015; Tan et al. 2016; Lecoutre, Negrevergne, & Yger, 2017). Transfer learning is especially useful in case of limited samples and while the Wikiart dataset is larger than previous datasets it is not sufficiently large to train a CNN from scratch (Goodfellow et al., 2006; Tan et al., 2016))

Pre-trained CNNs are used in two different procedures, some studies use the features extracted from some of the layers, for instance the penultimate layer and train a classifier on the feature vector of activations of this layer, sometimes combining engineered features. The other strategy is to retrain some or all of the layers and thereby adjusting the network to the task of painting classification, also known as fine-tuning. This study focuses on the former strategy, i.e., transfer learning. Previous studies have already showed that the concatenation of the feature vectors of different layers of a pre-trained CNN and concatenating CNN feature vectors with low-level descriptors improves accuracy (Bar et al., 2014; Karayev et al., 2014, Saleh, & Elgammal, 2015; Balakrishan, Rosston, & Tang, 2017). This study improves upon the current literature by exploiting the combined feature vectors of the layers of two different pre-trained CNN's for the classification of paintings from the Wikiart-dataset into one of 25 painting styles.

The rationale of using two CNNs that are both trained on the same task is that by utilizing both models, our model could be able to capture the larger variance in presentation of textures, object parts, abstract forms and objects (content) within paintings. This larger variance exists, because different painters or styles each have their own characteristic manner of painting. This results in variant presentation of content within paintings or inconsistent color and texture patterns of visual classes within images (Carneiro et al., 2012). As a result, certain lower level textures are more prevalent within certain styles or painters. Figure 1 demonstrates the difference between the styles Expressionism and Realism with regard to textural and object presentational difference, for instance the edges of the buildings from Expressionism are much sharper than those of Realism which seems to contain more details overall. That different CNNs could be able to capture slightly different textures and presentations of objects is based on the following two conjectures. The first conjecture is that, while the CNNs are trained on the same task (natural image object detection) the backpropagation algorithm does not necessarily converge in the global optimum, but is more likely to converge in an efficient representation of the data (Choromanska, 2014). The latter could result in slightly different filters for different CNNs and thereby enables the different CNNs to capture a larger variation in presentation. While it is likely that certain filters of both CNNs are sensitive to the same information, but with slightly different coefficients, it also reasonable that the different CNNs are sensitive to different information or content. The second conjecture is that different CNNs could be sensitive to different information is that the number of filters per CNN differs, therefore the CNNs with more filters could capture more or different information. Although the CNN with the lesser number of filters could also capture the same but less information than the CNN with more filters. This study investigates if the feature vector of different CNNs complement each other and are able to capture the larger variance of content that exists in paintings. Combining different classifiers is also known as *ensemble learning* and is based on the fact that classifiers with similar training performance may have different generalization performance (Polikar, 2006). While our networks are trained on different types of data (natural images vs. paintings) it is certainly the case that the training performance is not representative and that there is a difference between training and generalization performance.

To investigate if feature vectors of different CNNs complement each other this thesis utilizes the pre-trained VGG-19 and Resnet50, which are both pre-trained on ImageNet (Simonyan, Karen, & Zisserman, 2014; He et al., 2016). The main reason that this study utilizes both networks is that both networks reach state-of-the-art performance for painting style classification and the results of our study can be compared to their results (Balakrishna, Rosston, & Tang, 2017; Kedia, 2017; Lecoutre, Nevergrevergne & Yger, 2017). Balakrishna, Rosston, & Tang already showed the feature vector of the ultimate layer of VGG19 can be successfully exploited for classifying painting style, while the study also showed that the ultimate layer of Resnet18 (shallower version of Resnet50) surpasses VGG19 (2017). Contrarily, Kedia (2017) showed that it is possible to achieve state-of-the-art performance with VGG-19 by fine-tuning the network in combination with a weighted loss-function and multitask learning. Lecoutre, Nevergrevergne & Yger (2017) demonstrated that the feature vector of ResNet50 outperforms VGG19 and that fine-tuning the last 20 layers of ResNet50 leads to state-of-the-art performance.

The main goal of this thesis is to examine whether the concatenation of the Imagenet pre-trained Resnet50 and VGG19 yield better results than obtained with solely Resnet50. This study uses the results of the finetuned Resnet50 of Lecoutre, Nevergrevergne and Yger as a baseline while previous studies have shown that the pre-trained feature vectors of Resnet50 outperform VGG19 (Tan et al., 2015). Moreover, the dataset of the former study is available, which makes comparison of results more reliable. This study investigates if the feature vectors of the different layers of the different models are able to extract different information or the variance in presentation of objects within paintings and do not correlate too much. Both models could extract the same features and thereby correlate with each other, since both models are trained on the same object-detection task and with the same dataset. Therefore, our study also investigates if feature vectors trained on a different task are able to complement each other. The problem statement of this study is:

PS: Can painting-style classification be improved by combining feature vectors of VGG-19 and ResNet50?

To investigate this problem statement, the accuracy of the vectors of different layers of both CNNs are first investigated to determine which vectors to concatenate. Tan et al. already investigated the effectiveness of different feature vectors and showed that ultimate convolutional layer feature vector is more effective than the softmax output vector (2015). Bar et al. investigated the effectiveness of combining several feature vectors and found that combining different feature vectors of a pre-trained CNN with a feature engineered vector improves accuracy (2015). The unique aspects of this study are threefold. First, the combination of the feature vectors of different pre-trained CNNs has not been studied before. Second, this study differs from previous studies by investigating the feature vectors of lower layers of the network. These lower layers could indicate textures, object parts or abstract forms, which could be a better representation for artistic images than higher layers that are concerned with entire object detection. Third, while most studies of painting style classification use SVM or a FC-softmax layer, this study also investigates the effectiveness of adding an extra layer.

Our problem statement and unique aspects lead to the following three research questions:

RQ1 What are the accuracies associated with the individual layers of VGG-19 and ResNet50?
RQ2 What is the benefit of adding a fully connected layer instead of using a traditional classifier?
RQ3 What is the performance benefit of combining the best performing feature vectors of both CNNs?

1.2 Outline of thesis

The thesis is organized as follows. Section 2 gives an overview of the literature on painting classification, discussing related work. The first part of this section focusses on literature regarding feature engineering, while the second part reviews studies that use feature learning and transfer learning techniques. Section 3 introduces the convolutional neural networks used by this study. The first part reviews the VGG network, while the second part examines the Resnet-50 network. Section 3 demonstrates the experimental setup of this study, discussing the dataset in the first part and the feature extraction and learning methods in the second part. The experiment and results of this setup are discussed in Section 5. Section 6 finishes with a general discussion and conclusions.

Section 2: Related work

In this section we review related work on feature engineering (2.1), feature learning and transfer learning (2.2), and ends with a summary of related findings and a description of our approach (2.3).

2.1 Feature engineering

The digital analysis of paintings begun as a tool for assisting art experts/historians in measuring low-level quantitative features of paintings, such as space, texture, form, shape, color, tone and brushstrokes. These analyses focus on identifying distinctive artistic qualities by utilizing several descriptors (e.g. pigmentation analysis), which could be utilized by experts to assist their judgement (Stork, 2009). For most of the studies the primary goal is artist authentication or style detection and target one or a limited amount of artistic styles with a small set of high resolution images. An overview of all the studies which reported database size and accuracy can be found in Table 1 in the Appendix. This table clearly shows two patterns, namely that the number of categories, and thereby the database size, but also the error-rate increases as time progresses.

Sablatnig, Kammerer, & Zolda were among the first researchers to investigate artists. The study used color, shape and brush strokes to identify artist's personal style (1998). Several studies that followed analyzed brushstrokes using textures statistics to determine if the painting contained a painter's brushstroke style also called signature (Li, & Wang, 2004; Lyu, Rockmore, & Farid, 2004; Berezhnoy, Postma, & van den Herik, 2005; Johnson et al., 2008; Li et al., 2012). Keren creates local DCT transforms of image patches and uses a Naive Bayes with majority vote to classify five artist with 86%

accuracy (2002). Widjaja, Leow and Wu test patches of several color models RGB, HSV, HSI, HLS, and CIELAB and showed that combining four color channels, more specifically *H* and *L* of HLS, *S* of HSI and *a* of CIELAB with DAGSVM resulted in 85% accuracy with four painters. Lombardi (2005) classifies the styles of artists using feature modeling of the formal elements that art historians use to classify paintings, more specifically the study models light, line, texture and color with several descriptors for each formal element. Additionally, the study tested several classification schemes, such as *k*-Nearest-Neighbors (k-NN), Hierarchical Clustering, SOM, and MDS. Lombardi demonstrated that using Color Palette description performs just as well as other color descriptors that preserve additional spatial and frequency color information with less overhead storage giving it an advantage over other descriptors.

Most studies after Lombardi followed his example and examined a large number of descriptors and classifiers on different classification tasks. These studies also examined intermediate levels descriptors which analyze local regions of the image, e.g. the scale-invariant feature transform (SIFT), rather than whole image as well as higher semantic level features (e.g. Classemes). Zujovic et al. classified 5 styles by encoding color, texture using Canny edge detector, HSV histograms and pyramid decomposition in combination with four different classifiers and showed that a combination of the three descriptors in combination with AdaBoost achieved the best result with 68.3 accuracy (2009). A study conducted by Shamir, Macura and Orlov tested a large set of image features in combination with different image transform such as wavelet-transform and weighted the informativeness of the features using Fisher vectors (2010). The study showed that Fisher vectors are an effective way of dealing with less informative features. Agrawal & Jou used Histogram of Oriented Gradients (HOG) descriptor and several learning schemes to identify five Renaissance painters and achieved the best result 65% accuracy using a Naive Bayes classifier with simple color histogram features (2012). Arora et al. tested intermediate levels (SIFT and Color Sift) versus semantic level features (Classemes) and demonstrated that the latter are more effective (65.4% accuracy) for classifying 7 styles (2012). Ivanova et al. tested the Moving Pictures Experts Group-descriptors (MPEG-7) for the classification of 18 artists. Of MPEG-7 only the color and edge descriptors were utilized, while only the color descriptors showed to be reliable for creating profiles of artists (2012).

The previous studies investigated descriptors with very small datasets (maximum of 1,000 samples). This makes it hard to determine the utility of the descriptors, since some descriptors might be effective for certain distributions of style, genre or painters (Florea et al, 2015). Khan & van de Weijer (2014) is the first study to use a database size of a multiple of thousand, more specifically 4,266. The study used several local and global features within a Bag-of-Words pipeline in combination with SVM to classify artists and styles. The study demonstrated that combining the feature vectors increases the accuracy and CN-SIFT has superior performance compared to SIFT alone on both tasks. The study was able to classify 62% of 13 styles and 53.3% of 91 artists correctly using a combination of the descriptors (2014). Agarwal et al. classified 6 genres and 10 styles using SIFT, GIST, HOG, Local Binary Pattern

(LBP), gray level co-occurrence matrix (GLCM) and the CIELAB color descriptor and demonstrated that HoG and SIFT where the most informative features (2015). Moreover, the study classified 84.56% of six genres and 62.37% of ten styles correctly using an ensemble in combination with LIBSVM with χ2. Interestingly, they increased the dataset size with 300 for each category within each classification task, but still the accuracy rate declined tremendously when more categories were classified. This suggests that it is harder to separate more categories using LIBSVM and an equal number of engineered features. Although, style classification could also be harder than genre classification. Florea et al. were the first to explicitly investigate the effectiveness of certain descriptors with regard to 12 different styles (7,224 images). The study tested several color and texture descriptors and showed that GIST, Color Structure Descriptor (CSD) and pyramid-LBP were generally the most informative features. However, some descriptors perform better in differentiating between certain specific styles. Moreover, they showed that adding more features does not always improve accuracy. Their highest accuracy score was achieved with a combination of pLBP, CSD and SVM (54.7 % accuracy). Florea et al. also tested the old LeNet NN of Lecun et al. and it performed only slightly worse (48.6% accuracy) than their actual model, while they admitted that their dataset was not suited for the model (1998). This exposes the tremendous potential of feature learning, while the former studies show that when datasets and the amount of categories increase it is harder to capture the variance by utilizing solely engineered features. Engineered features perform better if combined by simply concatenating the features vectors or using more complex combining methods. Moreover, the performance of intermediate levels descriptors is generally better. The next section describes studies that implement the successor of feature engineering, more specifically feature learning with Convolutional Neural Networks (CNN).

2.2 Feature learning and transfer learning

Feature learning diverges from feature engineering in that the features of the former are learned from the raw data using a CNN. CCNs were introduced by LeCun et al. in 1998 and used for the classification of handwritten digits, more specifically postal zip codes. Although the true breakthrough of CNNs was arguably in 2012 when AlexNet of Krizhevsky et al. (2012) won the ImageNet Large Scale Visual Recognition Challenge by achieving a top-5 error of 15.3%, more than 10.8% points ahead of the runner up. After the introduction of AlexNet several improvements of the CNN won the competition, thereby showing the great potential of CNNs (Simonyan, Karen, & Zisserman, 2014; He et al., 2016). The strength of CNNs is found in the fact that they are able to learn general abstractions that correspond to different levels of object descriptions, such as low levels that correspond to edges while higher levels correspond to object parts (nose, ear, and eye) or the object itself (face). CNNs are able to detect these different levels of object descriptions under a wide variety of circumstances. For instance, when objects vary in size and location, they can still be classified correctly. While the representations of the CNNs are abstract and contain different levels, it is possible to utilize the features of a CNNs trained on a certain task for a different task (Oquab et al., 2013; Donahue et al, 2013; Razavian, Azizpour, Sullivan,

& Carlsson, 2014). This process is called *transfer learning* and is especially useful in case of limited data. The results of transfer learning can be improved by fine-tuning the more task specific higher layers in the network to the new task, while freezing the weights of the lower layers (Yosinski, Clune, Bengio, Lipson, 2014). Several studies have also used transfer learning, fine-tuning and/or trained a CNN from scratch within the domain of painting classification.

In 2014 Karayev et al. (2014) introduced the Wikiart-dataset consisting of 100,000 paintings with style, genre and painter annotations extracted from www.wikipaintings.org (currently www.wikiart.org). The study used transfer learning of the sixth convolutional layer of AlexNet (pre-trained on ImageNet) to predict 25 styles with at least 1,000 images for a total of 85,000 images. The fifth and the sixth convolutional layer were defined as feature vector, with a dimensionality of 8,000 (5th layer) + 4,000 (6th layer). The study only reports the average precision (AP) and the sixth layer obtained 35.6%. Additionally, they also tested several existing features (e.g. SIFT, GIST, Self-Similarity) resulting a 15,000-dimensional vector which was used to train on a handcrafted "deep" architecture called MC-bit, which achieves 44.1% AP. Although, MC-bit performed better than CNN-layers the vector size is also bigger. Bar et al. tested low level descriptors, assembly of low level descriptors (PiCodes and MC-bit) and transfer learning with the fifth (9216-dimensional), sixth (4096-dimensional) and prediction layers of AlexNet (2014). Their dataset is smaller with 40,724 samples of 27 styles. The highest AP they achieve is 56% with a combination of PiCode-2048, the fifth and sixth layer, while their highest 43% accuracy is obtained with the sixth layer and PiCode-2048. Their confusion matrix also indicates that the subtleties that the algorithm is able to detect reflect reasonable confusions of closely related styles. Single low-level descriptors did not contribute to the accuracy of the CNN. Additionally, they tested two approaches for fusion of features, more specifically early fusion (EF) and late fusion (LF). While EF is training a classifier on the concatenated feature vectors, LF is training each feature vector separately and merging the outputs. For LF they used a re-ranking method based on a Borda-count, but did not find any significant difference between both fusion methods. Saleh & Elgammal used the 1000-dimensional output vector of AlexNet in combination with GIST, Classeme and PiCodes (2015). The study outperformed Bar et al. (45.97 accuracy) by combining the features with a method called Large Margin Nearest Neighbors (LMNN) and training a SVM on the resulting vector (2014). Interestingly, the vector they used for training the SVM was only 10% (400 dimensional) in comparison to the vector of Bar et al. However, when using the vectors of single features Classemes showed to be most effective with 31.77 % accuracy, almost twice as much as CNN-feature-vector (16.83%). This indicates that the output from the prediction-layer might be overly task-specific and thereby a worse predictor than the preceding layers.

Three earlier studies investigated the effect of fine-tuning the layers. Tan et al. investigated the effect of fine-tuning using a CNN that is inspired by AlexNet for the classification of style, genre and painters on the Wikiart-dataset. They demonstrated that a model pre-trained on ImageNet-dataset and fine-tuned on the Wikiart-dataset (54.50% accuracy) outperforms models that only use feature vectors from the pre-trained model (max. 45.95% accuracy) or models that are trained from scratch on the Wikiart-dataset (42.96% accuracy) for the classification of 27 styles. Suggesting that fine-tuning increases accuracy and also that the Wikiart-dataset is too small to train a CNN from scratch. Moreover, they showed that classification with SVM is slightly worse than using fully-connected (FC) layer for classifying the feature vector. Therefore, our study will use a FC-layer. Lecoutre, Negrevergne and Yger investigated transfer learning and fine-tuning using three different architectures, more specifically AlexNet, Resnet34 and Resnet50 (He et al., 2016) for the classification of 25 styles (83,186 images) (2017). They demonstrated that the pre-trained Resnet50 (49.4%) outperforms AlexNet (37.8%) by more than 10%. Additionally, the study showed state-of-the-art performance (61.1% accuracy) by fine-tuning 20 layers of the Resnet50 network, while fine-tuning more layers did not result in increased accuracy. The latter indicates that the lower layers in the network are more general while higher layers are more task specific. The study also tested their model against a different dataset and their model showed similar performance as on the Wikiart-dataset[3]. This indicates the consistency of the model and more importantly the labels of the Wikiart-dataset. Balakrishan, Rosston, & Tang fine-tuned VGG19, Resnet18 for classification of 10 and 20 artists from the Rijksmuseum Challenge (Mensink, & van Gemert, 2014) (2017). ResNet18 clearly outperformed VGG19. Additionally, the study demonstrated the advantage of retraining a network in two different practices. The first practice showed the differences between a VGG19-network with the last, the last two and the last three layers retrained and demonstrated that retraining three layers achieves the highest accuracy. In the second practice, they trained Resnet18 for 20 epochs with different settings. The last layer of Resnet18 is retrained for the first 10 epochs, while all the layers are retrained for the last 10 epochs. The training and validation accuracy increases tremendously after 10 epochs, reaching the peak accuracy of around 90% within 5 epochs while the first 10 epochs stagnated at 80%. Both practices showed that fine-tuning pre-trained models increases accuracy. The latter is also observed in a study of Kedia (2017) which fully fine-tuned the VGG-19 network and demonstrated the best performance reported on the Wikiart-dataset with 68.80% accuracy (2017). They achieved this result by adding a weighted cross-entropy loss-function and multitask learning to VGG-19 network.

2.3 Summary of related findings and our approach

Our review of previous studies demonstrate five things:

1. It is possible to use CNNs that are trained to recognize objects for classification of style.
2. Training a CNN from scratch is ineffective due to the current size of the datasets.
3. Fine-tuning the higher layers in the network shows state-of-the-art accuracy for the Resnet-network

[3] ErgSap-dataset extracted from www.ergsart.com

4. Fine-tuning the entire model in combination with multi-task learning and a weighted loss-function shows state-of-the-art performance for the VGG-network.
5. The accuracy of feature vectors of CNNs can be increased by concatenating with low and intermediate level descriptors at an early stage (early fusion) or later in the process (late fusion).

Kedia (2017) showed that state-of-the-art performance can be achieved by using a weighted loss-function and multitask learning in combination with the VGG19 network. However, the main goal of this thesis is not to achieve state-of-art performance, but to investigate the effectiveness of combining feature vectors of different CNNs. While the study does not use a weighted loss function or multitask learning the study of Lecoutre, Negrevergne and Yger (2017) is used as baseline for this study, since Resnet50-network outperforms VGG19. The scores reported in the studies of Lecoutre, Nevregergne, & Yger (2017) and Kedia (2017) are shown in Table 2. The baseline for our research is the result of the finetuned Resnet50 (boldfaced in Table 2). To investigate the main research question the study investigates the accuracies of the features vectors of different layers of different CNNs. The feature vectors with the highest accuracies are concatenated to investigate if the baseline performance can be outperformed by combining feature vectors.

Table 2. Overview of the best results on the Wikiart-dataset of previous studies. Boldfaced is the baseline score of this thesis.

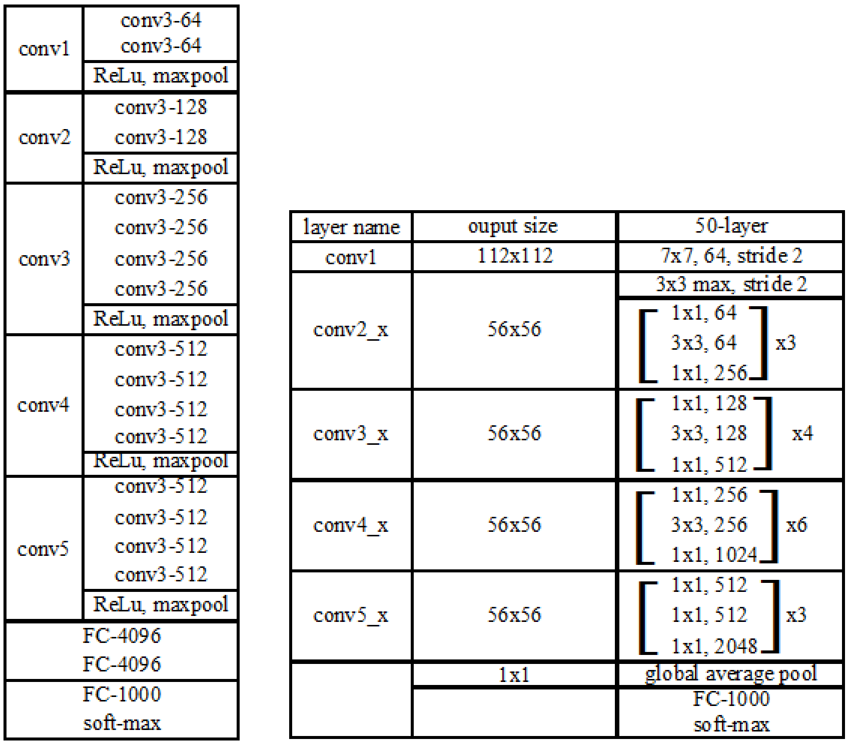
Previous studies	Architecture	Acc.
Lecoutre, Nevregergne, & Yger 2017	Pretrained Resnet-50 with retrained softmax FC-layer	49.4%
Lecoutre, Nevregergne, & Yger 2017	20-layers of Resnet-50 fine-tuned	**62.8%**
Kedia, 2017	Pretrained VGG-19 with retrained softmax FC-layer	45.95%
Kedia, 2017	Fully fine-tuned VGG-19 with weigthed cross-entropy and multitask learning	68.80%

Section 3: Convolutional Neural Networks and architectures

In this section, we review the VGG-network (3.1) and the Resnet-network.

3.1 VGG

VGG is developed by Simonyan, Karen, & Zisserman in 2014. It improved the architecture of Krizhevskiy’s et al. AlexNet by as the title of their paper already suggests having more depth within their network (2012). The reason that the study could go deeper is by applying several small size 3x3 filters instead of a bigger filter. Multiple stacked small size filters is better than on large size filter, because multiple non-linear activations can be applied to each filter which helps the model in learning more complex features. Additionally, the number of parameters is decreased by stacking several small size filters in comparison to stacking larger size filters. The left table of Figure 3 shows the full VGG architecture. The network consists of 5 convolutional layers which are all followed by a max-pooling operation. All hidden layers use a ReLu-activation function, except for the last FC-layer which uses a softmax-activation function. The number of filters doubles after each convolutional layer while the number of convolutional operations doubles after the second convolutional layer. This procedure reduces the spatial dimensions, but increases the depth or the number of features the network can recognize. The last three layers of the network consist of three FC-layers, the first and the second of these layers contain 4096-neurons and 0.5 dropout, while the last layer contains 1000 neurons, which is equal to the number of categories within the ImageNet-competition.



conv1	conv3-64 conv3-64
	ReLu, maxpool
conv2	conv3-128 conv3-128
	ReLu, maxpool
conv3	conv3-256 conv3-256 conv3-256 conv3-256
	ReLu, maxpool
conv4	conv3-512 conv3-512 conv3-512 conv3-512
	ReLu, maxpool
conv5	conv3-512 conv3-512 conv3-512 conv3-512
	ReLu, maxpool
FC-4096 FC-4096	
FC-1000 soft-max	

layer name	ouput size	50-layer
conv1	112x112	7x7, 64, stride 2
conv2_x	56x56	3x3 max, stride 2
		$\begin{bmatrix} 1x1, 64 \\ 3x3, 64 \\ 1x1, 256 \end{bmatrix}$ x3
conv3_x	56x56	$\begin{bmatrix} 1x1, 128 \\ 3x3, 128 \\ 1x1, 512 \end{bmatrix}$ x4
conv4_x	56x56	$\begin{bmatrix} 1x1, 256 \\ 3x3, 256 \\ 1x1, 1024 \end{bmatrix}$ x6
conv5_x	56x56	$\begin{bmatrix} 1x1, 512 \\ 1x1, 512 \\ 1x1, 2048 \end{bmatrix}$ x3
	1x1	global average pool
		FC-1000 soft-max

Figure 3. Left table of the figure shows VGG-19 and the right table of the figure shows ResNet50. The conv+number in the left of both table represent the convolutional layer number, so 'conv1' means the first convolutional layer. The conv1-layer of VGG-19 consists of two layers of 64 convolutional filters of size 3 x 3 each followed by a ReLu-activation function. After the entire convolutional layer a max pooling operation is applied. The conv2_x layer of Resnet50 consists 3 building blocks with each 3 layers shown within the brackets. Each building block contains a residual function (shown in Figure 4) from the beginning of the block to the ending of the block. A building block of conv2_x consists of 64 filters of 1 x 1, 64 filters of 3 x 3 and 256 filters of 1 x 1, which are followed by a ReLu-activation function. The entire convolutional layer is followed by an average pooling operation (not shown in the table). The left table is reprinted and adjusted from: "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan, and A. Zisserman, 2014, arXiv.org, p. 3. The right table is reprinted and adjusted from: "Deep Residual Learning for Image Recognition" by K. He, X. Zhang, S. Ren, and J. Sun, 2015, Proceedings of the IEEE conference on computer vision and pattern recognition, p. 774.

3.2 Resnet

VGG proved that deeper networks perform better than shallower networks. Therefore, He et al. wondered if better networks simply add more layers (2015). Several studies showed this was not the case, the problem is that deep networks are more difficult to train, due the problem of vanishing/exploding gradients (Glorot, & Bengio, 2010). The vanishing gradient problem exists, because during backpropagation the gradient decreases exponentially making the errors of the lower layers vanishingly small. The result of this problem is that with the network depth increasing accuracy saturates and then reduces rapidly. This applies not only to the test-accuracy, but also the train accuracy drops. The latter implies that the accuracy-drop is not caused by overfitting.

He et al. solve this problem by creating shortcut connections between convolutional layers, known as identity connections (2015). Figure 4 shows the architecture of a residual connection. *F*(*x*) is the residual function and is calculated by subtracting the original input *x* from the desired or hypothesized output *H*(*x*), so *F*(*x) = H*(*x*) - *x*. The idea is that the layer learns only the residual function. If *x* is equal to *H*(*x)* it can just skip the learning. This does not resolve the vanishing gradient problem, while it does solve the problem. As Veit, Wilber and Belongie argue, what the network basically does by skipping the connections or the filters is creating ensembles of shallower networks, also known as the *unraveled view*, and thus not resolving the problem. Figure 5 illustrates the unraveled view for a network that consists of three convolutional layers (*f1, f2* and *f3*). The right of the figure shows that with the skip connection it is possible to create eight shallower networks instead of one, for instance one option is to skip *f1,* another option is skipping *f2* or several layers could be skipped for instance by only using *f2.* These residual connections make it possible to go even deeper, since a layer that does not add any extra information is simply skipped solving the vanishing gradient. The right table of Figure 3 shows the architecture of ResNet50, which consists of 50 trainable layers instead of the 19 trainable layers of VGG-19.

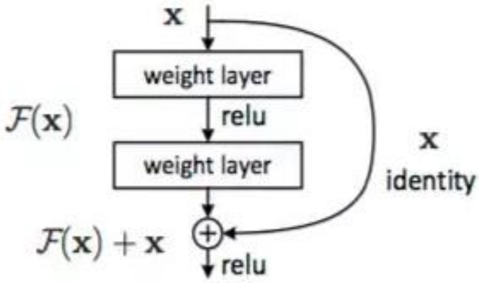

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4. The residual network building block. The x identity is the skip connection of the original input to the layer. Reprinted from “Deep Residual Learning for Image Recognition” by K. He, X. Zhang, S. Ren, and J. Sun, 2015, Proceedings of the IEEE conference on computer vision and pattern recognition, p. 771.

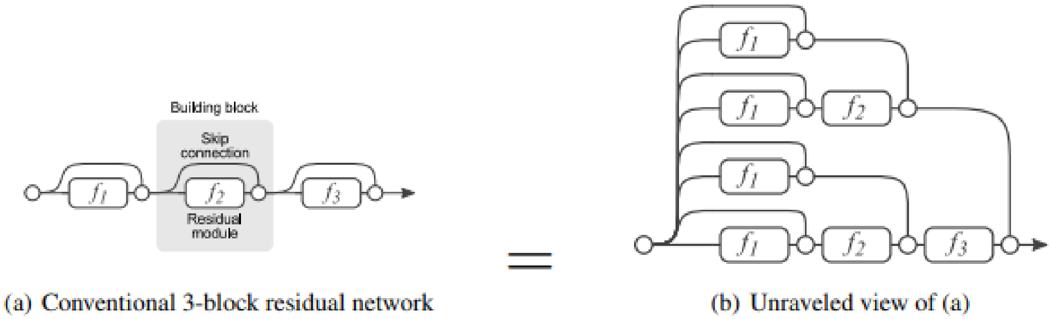

Building block
Skip connection
Residual module
f_1
f_2
f_3
=
(a) Conventional 3-block residual network
(b) Unraveled view of (a)

Figure 5. Illustrating the unraveled view of the residual network. The left of the pictures shows the standard network for the layers *f1, f2* and *f3*. The right of the picture shows all possible shallow networks of *f1, f2* and *f3* that result from the skip connection. Reprinted from “Residual Networks Behave Like Ensembles of Relatively Shallow Networks” by A. Veit, M. Wilber, and S. Belongie, 2016, Advances in Neural Information Processing Systems, p. 3.

Section 4: Experimental setup

In this section, the dataset that is used for this study is described (4.1) as well as the experimental protocol (4.2).

4.1 Dataset description

The dataset was extracted from www.wikiart.org and consists of 100,000 paintings labeled with artist, style, genre and data (Karayev et al., 2014). The paintings are labeled by a community of experts. This study uses a subset of the Wikiart-dataset[4] that was used by Lecoutre, Nevregergne and Yger (2017)[5]. The dataset consists of 82.133 paintings in 25 styles of which 66.549 are used for training and 7.383 for validation and 8,222 for testing. Figure 6 shows the large difference between the amounts of samples in each style, while Impressionism and Realism have more than 10,000 images, 10 styles contain just 10% (1000-1400) of those images. The styles are equally distributed among the splits.

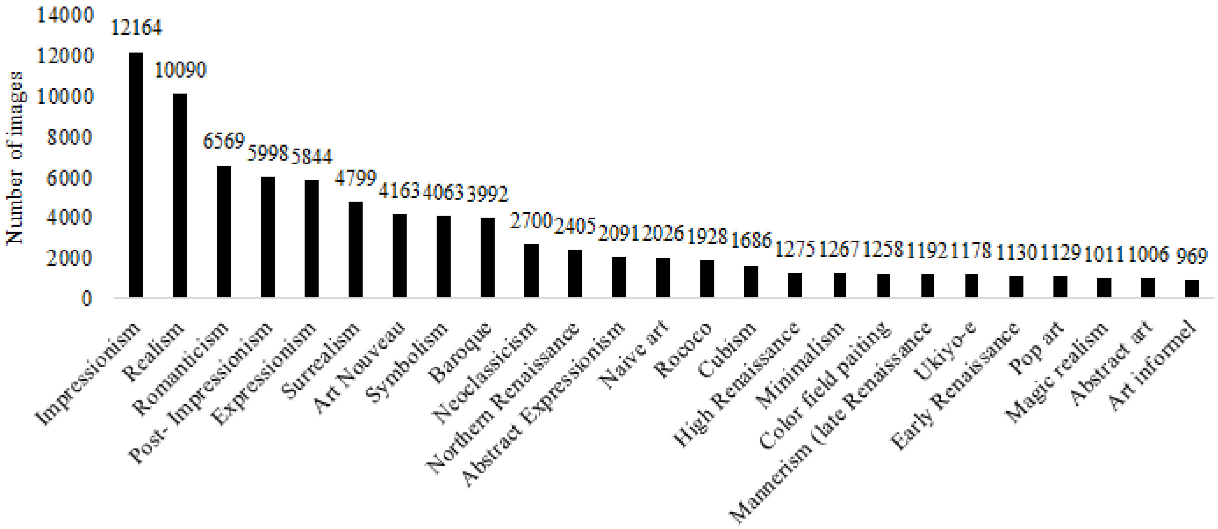


Figure 6. Distrubution of styles in the dataset.

4.2 Experimental protocol

This study uses the ImageNet pre-trained models and weights from Keras[6]. The input shape for both models is 224 by 224 and the images are resized by center-cropping. The images are then preprocessed according to the Keras application pre-processing function for each model. This function subtracts the ImageNet-dataset mean from the paintings, while both models could have slightly different means both models have different preprocessing functions. The pre-trained model of VGG19 and ResNet50 are also downloaded from Keras. From both models several feature vectors are extracted. For VGG19 the 1000-dimensional prediction FC-layer, the two FC-layers (size = 4096) beneath and the third, fourth and fifth convolutional layers are extracted, resulting in five feature vectors. Extracting the feature vectors of

[4] The dataset can be downloaded from: www.lamsade.dauphine.fr/~bnegrevergne/webpage/software/rasta/wikipaintings_full.tgz

[6] www.keras.io/applications/.

convolutional layers is done after the pooling operation is applied, resulting 256 dimensional feature vector for the third and 512 dimensional feature vector for the fourth and fifth convolutional layer. For ResNet50 the prediction layer and the fifth (2048-D), fourth (1024-D), and third (512-D) convolutional layers are extracted. Additionally, the fifth (2048-D) convolutional layer of the fine-tuned model of Lecoutre, Nevregergne and Yger is extracted.

These feature vectors are then used to train one FC-layer with a softmax activation function for predicting the styles. For every experiment in this study, the softmax FC-layer is optimized for 40 or 80 epochs using the gradient descent optimizer Adam with settings: learning rate is 0.001, 0.9 for β1, 0.999 for β2, ε is 10, batch size is 256 and all layer-weights are initialized with keras Glorot-normal-initizialization^{-8}(Kingma, & Ba, 2014; Glorot, & Bengio, 2010). Adam-optimizer is preferred since other algorithms are not adaptive and need to be fine-tuned for each vector (Rudy, 2016). The feature vectors are also trained in combination with a more complex model with an additional 4096-dimensional FC-layer and dropout before the softmax layers, to determine if more complex models are more successful. The amount of dropout is determined by running the model with several dropout settings ranging from 0.1 to 0.9 with steps of 0.1. Analysis of the accuracy scores of the validation-set determines the quality of the feature vectors for painting style classification and determines which feature vectors to concatenate.

The most successful single feature vectors are then concatenated vectors and trained with models of different complexity, more specifically different number of dense layers and amounts of neurons. This study applies early fusion of the feature vectors, while Bar et al. demonstrated that early fusion is as effective as late fusion. The complexity is restricted to the complexity used by the VGG-19 network which adds two dense layers of 4096 connections before the softmax-layer (Simonyan, Karen, & Zisserman, 2014). The thesis tests 1024, 2048, 4096 single and double layers. The double layer models restrict itself to the same amount of connections and dropout, for instance two dense layers of 2048 connections and 0.5 dropout. The search always starts with 0.5 dropout and checks validation accuracy to determine if the dropout or number of neurons must increase or decrease. The dropout is adjusted according to the under- or overfitting of the validation-set. The fifth convolution layer of the model from Lecoutre, Nevregergne and Yger is concatenated with the Keras ImageNet pre-trained VGG-5, and the fifth, fourth and third Resnet-layer, to see if their results can be improved by adding extra information from the ImageNet pre-trained VGG and Resnet-network (2017). While previous studies demonstrated that early fusion is as successful as late fusion our study also investigates the concatenated vectors in combination with late fusion, more specifically concatenating the output of the softmax vector of all the complex models and training a FC-layer with softmax on the vectors.

All models are evaluated using the validation-set accuracy. The single feature vectors are compared to the results single feature vectors of previous studies, to determine if our study is able to replicate findings of previous studies. Moreover, to investigate if adding an additional FC-layer improves results of previous studies. Only the best concatenated models are evaluated on the test-set.

Additionally, confusion matrices are created for these models to inspect the confusions and differences between confusions of the models.

The implementation is written in Python, using the Keras framework with Tensorflow as a backend. The experiments were run on the freely available Tesla K80 GPU of the Google Colaboratory Jupyter notebook environment (https://colab.research.google.com/).

Section 5: Experiments and results

5.1 RQ1: Single layer feature vector with softmax FC-layer

Table 3 shows the results of the simple model with one sofmax FC-layer and indicates that the single layers of the Resnet-network are better feature vectors as previous studies also showed. For the VGG-network the FC1-layer is the most successful, although the representation is much bigger than the 5th-layer (4096 versus 512 dimensional). Additionally, Table 3 shows the tremendous drop in accuracy between the 5th and the 4th layer of the VGG-network. Interestingly, the same does not hold for the Resnet-network for which the 4th layer is an improvement over the 5th layer. Also, the 3rd layer shows good results, improving the best layer of the VGG-network. Table 3 shows the best scores for both models and indicates that the single layers of the Resnet-network are better feature vectors. Additionally, the accuracy scores of the single layer feature vectors moderately improve previous studies, which are shown in Table 2. The VGG-FC2 layer is marginal improvement upon the first model of Kedia, although they used 27 instead of 25 styles and classified different styles (201). Interestingly, the accuracy score of the Resnet-5 layer is a considerable improvement upon the first model of Lecoutre, Nevregergne and Yger which use the exact same dataset (2017).

Table 3. Overview of the best results for single feature vectors using simple (only softmax FC-layer) and complex models (4096-FC layer + dropout + softmax FC-layer). The finetuned Resnet-5 is obtained from Lecoutre, Nevregergne and Yger (2017)

Architecture	Acc. of simple model	Acc. of complex model
VGG-1000	30.4%	-
VGG-FC2	47.3%	-
VGG-FC1	48.5%	53.5%
VGG-5	45%	55.8%
VGG-4	14%	-
Resnet-1000	30%	-
Resnet-5	53%	61.6%
Resnet-4	56%	62.7%
Resnet-3	48%	57.4%
Finetuned Resnet-5	61.5%	61.4%

5.2 RQ2: Single layer feature vector with additional FC-4096 layer

To investigate the performance of more complex models an additional 4096-dimension FC-layer with dropout is added. The dropout differs per model ranging from 0.1 to 0.8 and can be found in Table 4.

The results are illustrated in Table 3. The results clearly outperform the simple models and previous studies, while all the accuracy scores increases tremendously. The VGG-5 layer outperforms the VGG-FC1 layer, which is bigger and has lower accuracy. The Resnet-4 layer is again slightly better than the 5th layer with half the size (1024 versus 2048). We also tested the model 5th-layer of the fine-tuned model of Lecoutre, Nevregergne and Yger and adding a more complex on that feature vector did not improve their model with a score of 61.5%.

Table 4. Dropout used for different feature vectors of complex models

Layer	Resnet-5	Resnet-4	Resnet-3	VGG-5	VGG-FC1	finetuned Resnet-5
Dropout	0.8	0.4	0.1	0.8	0.8	0.1

5.3 RQ3: Concatenated feature vectors of different layers of different pr-trained CNNs

Table 5 shows the accuracy scores of the early fusion concatenated models with a single FC-4096-layer and 0.4 dropout, while models of different complexity and dropout did not improve the validation accuracy and only improved the training accuracy. The early fusion concatenated feature vectors slightly decrease the performance of the better performing single feature vectors, although only for the combined Resnet-3 + VGG-5 feature vector the accuracy slightly improves. Concatenating the softmax prediction layers (late fusion) and of all the models and training a softmax-layer on that vector results in an increase in performance with 64.1% outperforming all early fusion models. The ‘Resnet-5-fine-tuned’ in Table 2 refers to the fifth convolutional layer of Lecoutre, Nevregergne and Yger (2017). Concatenating the fine-tuned layers with different layers of the ImageNet pre-trained networks of VGG and Resnet increases performance and outperforms the original fine-tuned model, although the third convolutional layer of Resnet only slightly improves the accuracy of the vector. All the models use one 4096-FC layer and 0.7 dropout.

Table 5. Results of concatenated feature vectors trained with a FC-4096 layer, dropout of 0.7 and a softmax layer. Boldfaced are improvements upon previous results.

Architecture	Accuracy
Resnet5 + VGG-5	60,4%
Resnet5 + VGG-FC1	51,8%
Resnet5 + Resnet4	61,7%
Resnet5 + Resnet3	60,5%
Resnet4 + VGG-5	59,5%
Resnet4 + VGG-FC1	54,2%
Resnet4 + Resnet3	63,0%
Resnet3 + VGG-5	57,5%
Resnet-5-fine-tuned + VGG-5	**63.7 %**
Resnet-5-fine-tuned + Resnet-5	**63.3%**
Resnet-5-fine-tuned + Resnet-4	62.3 %
Resnet-5-fine-tuned + Resnet-3	62.0 %

To determine if our models can generalize to unseen data the best models are evaluated on the test-set and the results are shown in Table 6. The results in Table 6 indicate that our best models can generalize to unseen data, while the accuracy is stable. Additionally, the normalized confusion matrices of the best performing single layer models (Resnet-4) and concatenated vectors (Resnet-5-fine-tuned + VGG-5 and Resnet-5-fine-tuned + Resnet-5) models are shown in the Appendix (Figure 9 and 10). All the confusion matrices seem to confuse the same styles. Better models are simply less likely to make mistakes, but do not make different mistakes. The biggest confusions of the best model (Resnet-5-fine-tuned + VGG-5) are between Post-Impressionism and Impressionism (19%), between Mannerism and Baroque (16%) and between Romanticism and Realism (16%). Interestingly, all these styles flourished centuries after each other, therefore these confusions seem to be reasonable. Ukiyo-e is the most distinctive style, which is predicted correctly 90% of the times.

Table 6. Test-set results of best models. Boldfaced are improvements of previous results.

Architecture	Accuracy
Resnet-4	**63.1%**
Resnet-5-fine-tuned + VGG-5	**63.6%**
Resnet-5-fine-tuned + Resnet-5	62.4%

Section 6. General discussion and conclusion

Our study investigated if concatenating feature vectors of different layers of different ImageNet pre-trained Resnet50 and VGG19 improves performance in comparison to single layer feature vectors. Our results indicate that concatenating different layers trained on the same task does not improve performance, but results in a small decline in performance. Therefore, the combined feature vectors of different layers of the Imagenet pre-trained VGG and Resnet are not capable in capturing the larger variance in presentation of content within paintings.

As previous studies already showed the feature vector of different layers of Resnet-50 outperform the VGG-19 layers. Training a complex model instead of a simpler traditional classifier improves the performance for all feature vectors for painting style classification. Combining the ultimate layers or the fifth convolutional layers of Resnet and VGG does not increase performance. While, the Resnet fifth convolutional layer is the better feature vector, makes the same mistakes but smaller mistakes and is also the bigger feature vector it presumably recognizes all of the objects of VGG. However, it is more likely that both models do not really capture objects, but capture the feeling of the painting by combining several objects into a score for instance the painting is 0.14 human-like, 0.09 sheep-like and 0.06 sheep-like. The Resnet fifth convolutional layer could than probably still capture similar feelings as the VGG-network.

The biggest difference between the Resnet and VGG network is that the decline in performance of the lower layers of Resnet-50 is much smaller than that of VGG-19. The increased performance of the lower layers of the Resnet-network is probably caused by the residual function. The Resnet-50 network is better at capturing lower level features in comparison to the VGG-19 network which is oriented towards classifying entire objects. While, the Resnet fourth convolutional is oriented towards object parts and already achieves good performance, it is expected that combining this layer with the VGG fifth convolutional layer that is oriented at entire objects increases performance. The latter should be expected since VGG and Resnet could potentially both capture different local optima, thereby detect different object or content presentation that is present within paintings. Moreover, both feature vectors focus on different content (object parts and objects), which could also complement each other. However, combining the lower layers features of the Resnet fourth and fifth convolutional layer with the VGG-5 layer does not improve performance of the individual layers. A possible explanation is that, while the lower levels features of Resnet are better representations of object parts or textures, these representations are only useful in paintings in which the entire object is also detectable. Another, possible explanation is that object parts or textures might contradict evidence from entire objects in predicting certain styles. The latter could be the case if a large proportion of the paintings from a certain style, for instance Realism, portray a certain object category e.g. humans. The network could be capable of capturing the specific texture, but still ‘decides’ that without a human the painting style is not Realism.

This study focused on the actual predictive power of combining both CNNs it could have benefited from exploring what both CNNs actually do with the paintings and what the actual predictions are. While the prediction or the softmax output layers of VGG-19 and Resnet-50 are poor feature vectors it is interesting to investigate the correlations between both layers to strengthen the thesis that both CNNs capture the same variance. Additionally, these feature vectors could be explored more thoroughly to investigate if the CNNs are actually able to detect objects within paintings or solely capture the feeling of the painting. Furthermore, the activations of different layers of both networks could be explored more thoroughly by for instance the deconvolutional operation of Zeiler and Fergus (2004). Besides looking at the mistakes of the single CNNS, the mistakes of the combined CNNs could also be explored in more detail. For instance investigating the correlations between predictions of the different combined models to support the thesis that all of the models make the same mistakes. While the latter is also explored within the confusion matrix, a correlation matrix that correlates predictions of certain layers with each other could explore this issue more thoroughly.

While concatenating feature vectors of pre-trained CNNs that are trained on the same task declined performance, vectors that are trained on different tasks do complement each other. Concatenating the fine-tuned fifth convolutional layer of Lecoutre, Nevregergne and Yger with the ImageNet pre-trained vectors of different layer of Resnet-50 and VGG-19 does improve accuracy. Kedia (2017) also demonstrated that fine-tuning in combination with multitask learning increases the information that can be extracted from the paintings. Therefore, future research could improve accuracy by utilizing the date or eras of the paintings in a multitask classification task, while our current model especially confuses closely related eras. Alternatively, the genre labels of the Wikiart-dataset can be utilized in this learning scheme. Moreover, future research could apply a weighted loss-function that accounts for the inequality in the amount of instances per styles, while Kedia (2017) showed that using such a loss-function increases accuracy for unequally distributed datasets. Increasing the training dataset by adding the Ergsart-dataset is also an option. Another alternative would be to use different CNN architecture(s), while architectures improve very fast.

Appendix

Table 1. Overview of painting classifications. Some studies classify several categories, acc. % (accuracy percentage) and database size are reported in the same order. Studies that do not report accuracy are not included for comprehensibility of the table.

Study	Classifies	Database size	Acc. %	Classifier	Important feature(s)
Keren et al., 2002	5 painters	<180	85	Naive Bayes	DCT transform coefficients
Widjaja et al., 2003	4 painter	100	86	DAGSVM	hue
Gunsel et al., 2005	5 styles, 12 painters	290	100, 100	SVM	6-d content-representative feature set
Siddique et al., 2009	6 styles	489	82.4	AdaBoost	MR8 filter bank
Zujovic et al., 2009	5 genres	353	68.3	AdaBoost	pyramid, HSV histograms
Ivanova et al., 2012	18 painters, 10 styles	600	71, 83	vcluster	MPEG-7 color descriptors
Shamir et al., 2010	3 styles, 9 painters	517	91	two-stage	-
Agrawal, & Jou, 2012	5 painters	750	65	Naive Bayes	color histogram
Arora, & Elgamal, 2012	7 genre	490	65.4	SVM	Classeme
Khan et al., 2014	13 genre, 91 painters	4266	53.1, 62.2	SVM	CN-SIFT
Karayev et al., 2014	25 styles	85000	44.1 AP	NN	MC-bit
Bar et al., 2014	27 styles	40724	43	k-NN	CNN. PiCode
Agarwal et al., 2015	6 genres, 10 styles	1800, 3000	84.56, 62.37	libsvm $\chi2$	SIFT, HOG
Florea et al, 2015	12 styles	7224	55	SVM	pLBP, CSD, GIST
Saleh, & Elgammal, 2015	27 styles, 23 painters, 10 genres	78449, 63691, 18599	45.97, 63.06, 60.28	LMNN	Classeme, PiCode
Tan et al., 2016	27 styles, 23 painters, 10 genres	78449, 20000, 65000	54.5, 74.14, 76.02	Fine-tuned AlexNet	-
Lecoutre, Negrevergne, & Yger, 2017	25 styles	82133	62.8	Fine-tuned ResNet50	-
Balakrishan, Rosston, & Tang, 2017	20 painters	20000	90	Fine-tuned ResNet18	-
Kedia, 2017	27 styles	82133	65.4	fully fine-tuned VGG-19	-

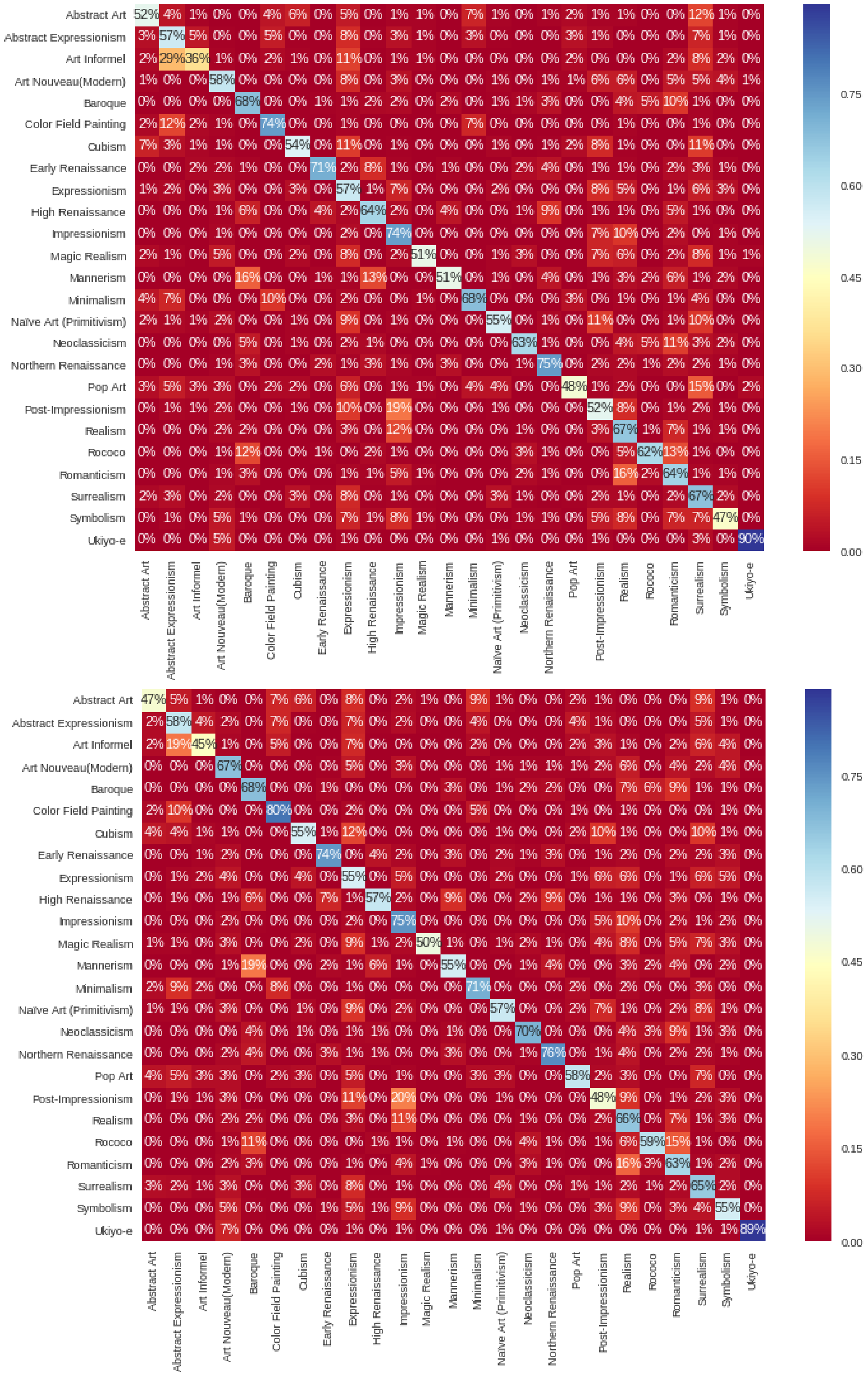


Figure 9. The normalized confusion matrices. Top is the concatenated vectors of Resnet-5-fine-tuned + Imagenet pretrained VGG-5. Top is the concatenated vectors of Resnet-5-fine-tuned + Imagenet pretrained Resnet-5. The vertical axis is the true style, while the horizontal is the prediction. The fine-tuned model is trained by Lecoutre, Negrevergne, & Yger (2017)

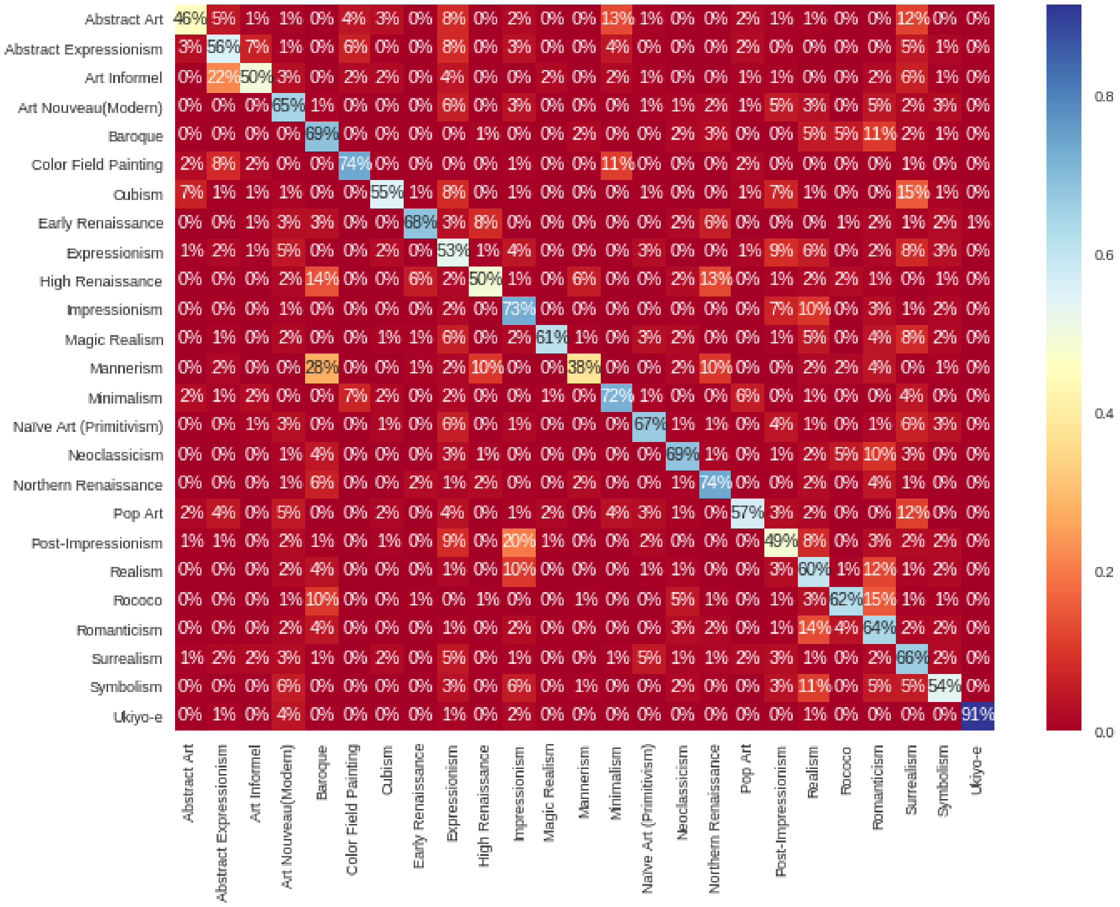


Figure 10. Normalized confusion matrix of the Resnet-4 convolutional layer.

Literature

Agarwal, S., Karnick, H., Pant, N., & Patel, U. (2015, January). Genre and style based painting classification. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on* (pp. 588-594). IEEE.

Arora, R. S. (2012). *Towards automated classification of fine-art painting style: A comparative study* (Doctoral dissertation, Rutgers University-Graduate School-New Brunswick).

Balakrishan T, Rosston S, Tang E (2017) Using CNN to classify and understand artists from the Rijksmuseum. Stanford technical report, pp 1–8

Bar, Y., Levy, N., & Wolf, L. (2014, September). Classification of artistic styles using binarized features derived from a deep neural network. In *Workshop at the European Conference on Computer Vision* (pp. 71-84). Springer, Cham.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798 1828.

Berezhnoy, I., Postma, E., & Van den Herik, J. (2006). Computer Analysis of Van Gogh's Complementary Colours. *Pattern Recognition Letters, 28,* 703-709. doi: 10.1016/j.patrec.2006.08.002

Carneiro, G., da Silva, N. P., Del Bue, A., & Costeira, J. P. (2012, October). Artistic image classification: An analysis on the printart database. In *European Conference on Computer Vision* (pp. 143-157). Springer, Berlin, Heidelberg.

Choromanska, Anna, et al. "The loss surfaces of multilayer networks." *Artificial Intelligence and Statistics*. 2015.

Crowley, E. J., & Zisserman, A. (2014, September). In search of art. In *Workshop at the European Conference on Computer Vision* (pp. 54-70). Springer, Cham.

Crowley, E., & Zisserman, A. (2014, September). The State of the Art: Object Retrieval in Paintings using Discriminative Regions. In *BMVC*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). Ieee.

Donahue, Jeff, et al. "Decaf: A deep convolutional activation feature for generic visual recognition." *International conference on machine learning*. 2014.

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

Fernie, E. (Ed.). (1995). *Art history and its methods: A critical anthology* (p. 223). London: Phaidon.

Gatys, L. A., Ecker, A. S., & Betghe, M. (2015). A Neural Algorithm of Artistic Style. *Computer Vission and Pattern Recognition*. Retrieved from https://arxiv.org/abs/1508.06576

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770 778).

Icoglu, O., Gunsel, B., & Sariel, S. (2004, September). Classification and indexing of paintings based on art movements. In *EUSIPCO* (pp. 749-752).

Ivanova, K., Stanchev, P., Velikova, E., Vanhoof, K., Depaire, B., Kannan, R., & Markov, K. (2012). Features for art painting classification based on vector quantization of mpeg-7 descriptors. In *Data Engineering and Management* (pp. 146-153). Springer, Berlin, Heidelberg.

Jou, J., & Agrawal, S. (2012). Artist identification for renaissance paintings.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2013). Recognizing image style. *arXiv preprint arXiv:1311.3715*.

Kedia, M. (2017). Fine-grained painting classification (Doctoral dissertation).

Keren, D. (2002). Painter identification using local features and naive bayes. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (Vol. 2, pp. 474-477). IEEE.

Khan, F. S., van de Weijer, J., & Vanrell, M. (2010). Who painted this painting? In *2010 CREATE Conference* (pp. 329-333).

Khan, F. S., Beigpour, S., Van de Weijer, J., & Felsberg, M. (2014). Painting-91: a large scale database for computational painting categorization. *Machine vision and applications*, *25*(6), 1385-1397.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM, 60,* 84-90. doi: 10.1145/3065386

LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks* 3361.10 (1995): 1995.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature, 521*. 463-444. doi: 10.1038/nature14539

Lecoutre, A., Negrevergne, B., & Yger, F. (2017, November). Recognizing Art Style Automatically in painting with deep learning. In *Asian Conference on Machine Learning* (pp. 327-342).

Lombardi, T. E. (2005). *Classification of Style in Fine-art Painting* (pp. 4908-4908). New York, NY: Pace University.

Lyu, S., Rockmore, D., & Farid, H. (2004). A digital technique for art authentication. *Proceedings of the National Academy of Sciences*, *101*(49), 17006-17010.

Mensink, T., & Van Gemert, J. (2014). The Rijksmuseum Challenge: Museum-Centered Visual Recognition. *ACM International Conference on Multimedia Retrieval (ICMR).*

Van Noord, N., Hendriks, E., & Postma, E. (2015). Toward Discovery of the Artist's Style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, *32*(4), 46-54.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717-1724).

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

Polatkan, G., Jafarpour, S., Brasoveanu, A., Hughes, S., & Daubechies, I. (2009, November). Detection of forgery in paintings using supervised learning. In *Image Processing (ICIP), 2009 16th IEEE International Conference on* (pp. 2921-2924). IEEE.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and system magazine*, *6*(3), 21-45.

Sablatnig, R., Kammerer, P., & Zolda, E. (1998). Hierarchical Classification of Paintings Using Face- and Brush Stroke Models. *Fourteenth International Conference on Pattern Recognition, 1,* 172-174. doi:10.1109/ICPR.1998.711107

Saleh, B., & Elgammal, A. (2015). Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813).

Siddiquie, B., Vitaladevuni, S. N., & Davis, L. S. (2009, December). Combining multiple kernels for efficient image classification. In *Applications of Computer Vision (WACV), 2009 Workshop on* (pp. 1-8). IEEE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Shamir, L., Macura, T., Orlov, N., Eckley, D. M., & Goldberg, I. G. (2010). Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. *ACM Transactions on Applied Perception (TAP)*, *7*(2), 8.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2015.7298594

Stork, D. G. (2009). Computer Vision and Computer Graphics Analysis of Paintings and Drawings: An Introduction to the Literature. *Thirteenth International Conference on Computer Analysis of Images and Patterns,* 9-24. doi: 10.1007/978-3-642-03767-2_2

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

Widjaja, I., Leow, W. K., & Wu, F. C. (2003, September). Identifying painters from color profiles of skin patches in painting images. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (Vol. 1, pp. I-845). IEEE.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (pp. 3320-3328).

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Zujovic, J., Gandy, L., Friedman, S., Pardo, B., & Pappas, T. N. (2009, October). Classifying paintings by artistic genre: An analysis of features & classifiers. In *Multimedia Signal Processing, 2009. MMSP'09. IEEE International Workshop on* (pp. 1-5). IEEE.