Learning to automatically recognize artists by their artworks

Master thesis

M.R.M. Teeuwen
ANR 715746

Data Science: Business and Governance
Faculty of Humanities
Department of Communication and Information Sciences
Tilburg University

Supervisor: Prof. dr. E. O. Postma
Second reader: Dr. W. Huijbers

August 2018

Preface

Before you lies the thesis "Learning to automatically recognize artists by their artworks". It has been written to fulfil the graduation requirements of the Communication and Information Sciences program, master track Data Science: Business and Governance, at Tilburg University. I was engaged in writing this thesis from December 2017 to August 2018.

I would like to thank my supervisor, Prof. dr. E. O. Postma, and my second reader, Dr. W. Huijbers, for their guidance and support during the process.

I hope you enjoy your reading.

Margot Teeuwen
Tilburg, August 2018

Abstract

Traditionally, the analysis of an artwork is performed by art experts. However, analytical methods have advanced over the past years. With the rise of computers and digital reproductions of works of art, it becomes possible to automatically attribute artworks. Because technologies have changed, digital computer storage and computational power is wider available, and there are more scientists with an analytical background, computational techniques to automate artist attribution are more widely applied than before.

Since the early 1990s, several convolutional neural network architectures have been proposed for image recognition tasks. Within the topic of artist attribution, the neural network learns to identify marks that are seen as characteristics of a specific artist. In this thesis a new approach is presented that makes use of a different convolutional neural network than ever used before.

The aim of this study is to examine whether the results of an experiment in artist attribution differ, when using a recent and well-developed convolutional neural network. To this end, the research question is as follows:

> *To what extent can the results of an experiment in automatic artist attribution be improved by using a current network architecture?*

To compare results, the dataset that is used for the experiments in this thesis is the same as the dataset used by Van Noord, Hendriks and Postma (2015): the Rijksmuseum Challenge dataset.

The research question is answered through several experiments which are executed by using the Inception V3 Network in Python. The results of these experiments indicate that the results of an experiment in artist attribution improve when using a current network architecture.

Further research could be undertaken to identify whether using a heterogeneous dataset, or a dataset where no visual marks, like different perspectives in the images, are present, influences the results of an experiment in artist attribution.

Contents

Chapter 1. Introduction

The first part of this chapter contains a general introduction into the landscape of automated artist attribution. Section 1.2 consists of the formulation of the research problem. Section 1.3 consists of the scientific and practical relevance of this thesis. Finally, section 1.4 describes the structure of this thesis.

1.1 Automated artist attribution

Over the past years, the technologies in image data acquisition have developed. Therefore, museums started to collect digital libraries of images of their collections. Due to more collaborations between image analysis researchers and art historians, technology developers are able to focus on image analysis tasks (Johnson et al., 2008). For various artworks, either the author is unknown or there is a continuing discussion about the authenticity of the attributions (Van Noord, 2018). Traditionally, the analysis of an artwork is performed by human art experts (Berezhnoy, Postma, & Van den Herik, 2006). To establish the cultural, historical, and economic value of an artwork, identifying the author of an artwork is important. In order to establish the value of an artwork, art experts need to have a certain amount of knowledge. Art experts acquire this knowledge by analysing artworks and their descriptions of the relevant aspects (Van Noord, Hendriks, & Postma, 2015). According to Johnson et al. (2008), "the problem of artist identification seems ripe for the use of image processing tools". When experts identify the artist of an artwork, they use not only their current understanding of the routines of the artist. Experts combine this knowledge by examining the presence of the "handwriting" of the artist, and with comparisons of a variety of technical data (Johnson et al., 2008). A handwriting could be, for example in the case of Van Gogh, the brushwork. The analysis of a digital representation of an artwork could help the art expert in the process of attribution (Johnson et al., 2008). The application of computational techniques to analyse artworks already existed years ago. However, only recently, analytical methods have produced a relevant impact on the ability to contribute to the analysis of an artwork. This is due to a historical division between science and humanities. Only since several years, interaction between these two domains occurs more often (Barni, Pelagotti, & Piva, 2005).

With the development of computer techniques and the rise of high-resolution digital reproductions of artworks attempts were made to automate the attribution of artworks (Johnson et al., 2008; Van Noord et al., 2015; Li, Yao, Hendriks, & Wang, 2015; Elgammal, Kang, & Den Leeuw, 2017). To identify the artist of an artwork, machine learning algorithms may be helpful because they can do this automatically. Collaboration between art experts and conservators already established the feature engineering for recognizing Van Gogh and other artists of his time as the original maker of their works of art. According to Van Noord et al. (2015), "this highlights the value of automatic approaches as a tool for art experts".

The issue of automated artist attribution has been addressed extensively, see Chapter 3. As stated by Van Noord et al. (2015), "convolutional neural networks have not yet been applied for automated artist attribution". They were the first to use convolutional neural networks to automatically attribute works of art. In their work they applied "PigeoNET", a variant of AlexNET, the network responsible for the breakthrough ImageNet performance (Krishevsky, Sutskever, & Hinton, 2012). Since the publication of Van Noord et al.'s study, neural network architectures have been considerably improved in terms of efficiency and performance. Chapter 2 provides an overview of AlexNET and these improved network architectures. In this thesis, one of the improved network architectures will be applied to the task of author attribution to determine if and to what extent it outperforms AlexNET. The conditions for the experiments in this thesis are the same as in the study by Van Noord et al. (2015), however slight differences in the experiment have been made for an optimal result, see Chapter 4.

1.2 Problem formulation

To detect the characteristics that determine the touch of an artist, Van Noord et al. (2015) presented a specific approach. This approach trains a convolutional neural network on a substantial set of digitalized imitations of various artworks. Hereby, he network is encouraged to discover visual features that are distinguishing for a particular artist (Van Noord et al., 2015). Ultimately, the task of automatic artist attribution is performed. By studying artworks that are representative of the artist, the distinguishing characteristics of that artist can be recognized. However, according to Van Noord et al. (2015), the absence of certain methods, in particular the automatization and determination of which criteria make an artwork representative and obtaining a dataset of decent size containing different images, is difficult. Therefore, Van Noord et al. (2015) suggest that to avoid the need for a decent sample, a big sample has to be taken. In this case this means that a substantial dataset is required, which includes many images per artist. Therefore, the dataset used for the experiment in the paper by Van Noord et al. (2015), and the dataset used for the experiments in this thesis, is the Rijksmuseum Challenge dataset. This dataset contains 122.039 digital photos of works of art made by 6.629 artisans, all represented in the Rijksmuseum in Amsterdam, the Netherlands (Van Noord et al., 2015).

Given that during the past years several improved network architectures have been proposed, the question arises whether the results of automatically recognizing artists by their artworks can be improved by using a different network than AlexNET. Therefore, the problem statement of this thesis is formulated as follows:

> *To what extent can the results of an experiment in automatic artist attribution be improved by using a current network architecture?*

1.3 Scientific and practical relevance

This research aims to make a contribution to the line of work reported by Johnson et al. (2008), Hughes, Graham, and Rockmore (2010), Van Noord et al. (2015), Li et al. (2015), Elgammal et al. (2017), who present an approach that tried to automatically attribute artworks to an artist.

The practical relevance of this research can be found in the fact that there are many paintings of which the author is still unidentified. For example, May 2018, when a new Rembrandt painting was discovered, after it already had been sold by auction house Christie's as a painting made by a painter "close to Rembrandt". The painting never was officially attributed to Rembrandt by Christie's (Pinedo & Ribbens, 2018), however a Dutch art dealer drew 15 curators and art historians into this situation, who assure the painting's authenticity as a Rembrandt (Rea, 2018). The research in this thesis can help to automatically attribute an artist to a painting, which will be helpful for auction houses, museums, but also the individual collector. Moreover, the identification of author-specific characteristics supports art historical investigations and may inform restauration efforts.

1.4 Thesis outline

This thesis comprises seven chapters, each divided in several subchapters. Chapter 1 contains the introduction of this research, the problem formulation, the scientific and practical relevance of this thesis, and the structure. Chapter 2 gives an overview of several convolutional neural networks and explains why the Inception Network is used in the experiments as a suitable representative of an improved network architecture. Chapter 3 discusses the related work in the field to place the thesis in a broader context. Chapter 4 describes the method used for this research. Chapter 5 contains the results of the experiments. Chapter 6 contains the discussion, which consists of the limitations of this research and recommendations for future work. Finally, Chapter 7 contains the conclusion.

Chapter 2. Convolutional Neural Networks

This chapter is introduced by a description of what convolutional neural networks actually are, followed by an analysis of several convolutional neural network architectures. Except from LeNet-5 and the Inception Network, which is a renewed version of GoogLeNet, the convolutional neural networks that are mentioned in this thesis are all architectures made for the ImageNet Large Scale Visual Recognition Competition (ILSVRC) and are all top competitors. The six network architectures described in this thesis, can be compared to each other because they all entered the same competition, the ILSVRC. Except from LeNet-5, which is mentioned because it was the first convolutional neural network, a breakthrough in the history of network architectures, all the networks named in this thesis have been built with the same goal: winning the ImageNet Large Scale Visual Recognition Competition. Furthermore, in this chapter the choice for the convolutional neural network used for the experiments in this thesis is explained, which is made based on two criteria: performance and complexity, in terms of number of parameters.

2.1 Structure of a Convolutional Neural Network

In the course of time, the convolutional neural network (CNN) accomplished several successes in computer vision tasks. A convolutional neural network, which is inspired by neuroscience, shares many characteristics with the visual system of the human brain (Liang & Hu, 2015).

A convolutional neural network contains several layers which consist of small computational units. Within these units, visual information is processed in a hierarchical and feed-forward manner. Each layer of units extracts a particular feature from the input image and is an assemblage of image filters. The output of a layer contains feature maps. Feature maps are versions, each filtered in a different way, of the input image. Gatys, Ecker, and Betghe (2015) argue that “higher layers in the network capture the high-level content in terms of objects and their arrangement in the input image, but do not constrain the exact pixel values of the reconstruction”. On the contrary, restorations of the bottom layers duplicate the precise pixel rates of the authentic picture. By adapting the filters of a CNN, the network might identify distinguishing features of an artist. These filters can be adjusted until an appropriate configuration is found. Besides information on the input images and the labels, this process does not need any prior knowledge. In the case of the experiments in this thesis, the label is the artist who created the artwork (Van Noord et al., 2015).

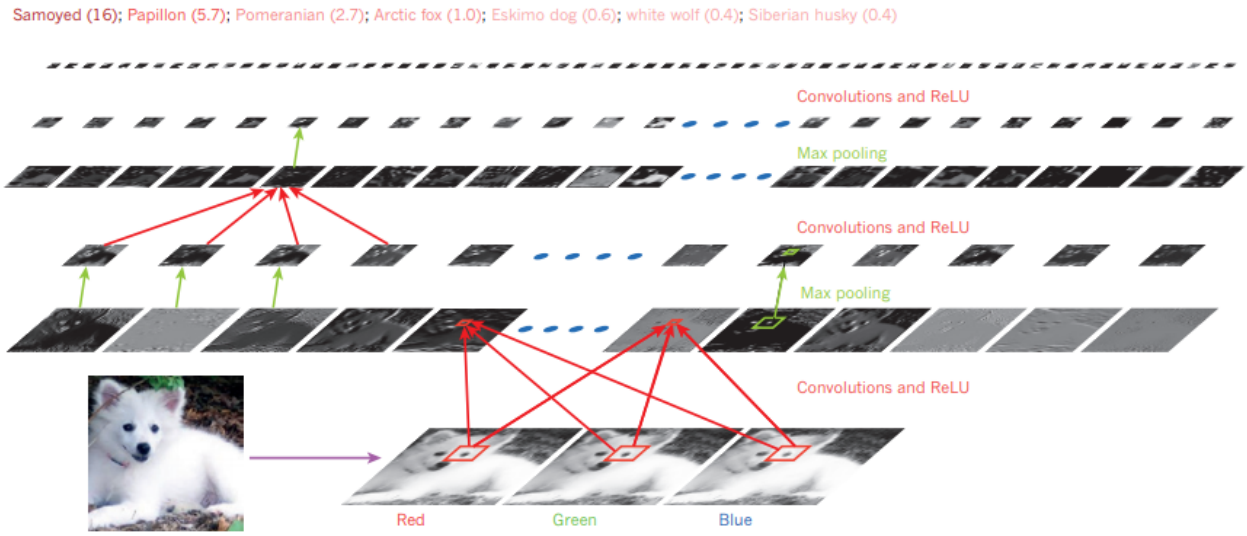

Figure 1. Inside a convolutional neural network. Reprinted from "Deep Learning", by Y. LeCun, Y. Bengio, and G. Hinton, 2015. *Nature, 521,* p. 438. Copyright 2015 by Macmillan Publishers Limited.

As to be seen in Figure 1 (LeCun, Bengio, & Hinton, 2015), the outputs of each horizontal layer are adapted to the picture of a Samoyed dog. "Each rectangular image is a feature map which corresponds to the output for one of the learned features of the image positions" (LeCun et al., 2015).

The architecture of a convolutional neural network consists of several stages. LeCun et al. (2015) describe a convolutional neural network as follows:

> The first few stages are composed of two types of layers: convolutional layers and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a filter bank. The result of this local weighted sum is passed through a non-linearity such as ReLU. All units in a feature map share the same filter bank. Different feature maps in a layer use different filter banks. The role of the pooling layer is to merge semantically similar features into one, the role of the convolutional layer is to detect local conjunctions of features from the previous layer.

With artist attribution, the network learns to identify the characteristics of an artist. As mentioned before, a convolutional neural network contains various layers. "The first layer is directly applied to images, subsequent layers to the responses generated by previous layers" (Van Noord et al., 2015). The layers of filters are called "convolutional layers". This is due to convolution being practiced to spread the filters to a picture, or to the outcome of a preceding layer. Within a convolutional layer the weights are shared, which is an advantage compared to a traditional neural network layer. "This allows the adaptive filters to respond to characteristic features irrespective of their position or location in the input" (Van Noord et al., 2015). The convolutional layers are succeeded by several so-called "fully-connected layers". These layers convert the intensity and the existence of the outcomes of the filters to a single confidence

rate per artist. “This score is high whenever the responses for filters corresponding to that artist are strong, the score is low when the responses are weak or non-existent. Thus, an unseen artwork can be attributed to an artist for whom the score is the highest” (Van Noord et al., 2015).

2.2 Historical overview of Convolutional Neural Networks

After the first application of a CNN in the early 1990s, computational hardware started to improve in capability and convolutional neural networks became more popular as an efficient learning approach (Alom et al., 2018). Therefore, new convolutional neural networks came to existence. After a description of the very first convolutional neural network, LeNet-5, in subchapter 2.2.1, the subsequent networks are explained in the subchapters thereafter.

2.2.1 LeNet-5

Since the early 1990s, several times convolutional networks have been applied. First of all, in the 1990s, LeNet was proposed. However, the algorithm was hard to implement until around 2010, due to restricted memory capacity and computation competence (Alom et al., 2018). Therefore, in 1998, LeCun, Bottou, Bengio and Haffner proposed a new architecture, known as LeNet-5. According to LeCun et al. (1998), “the ability of multi-layer neural networks trained with gradient descent to learn complex, high-dimensional, non-linear mappings from large collection of examples, makes them candidates for image recognition tasks”. In order to make sure there is a degree of shift, scale, and distortion invariance, convolutional neural networks combine three architectural concepts. These three concepts are “local receptive fields, shared weights, and spatial or temporal sub-sampling” (LeCun et al., 1998).

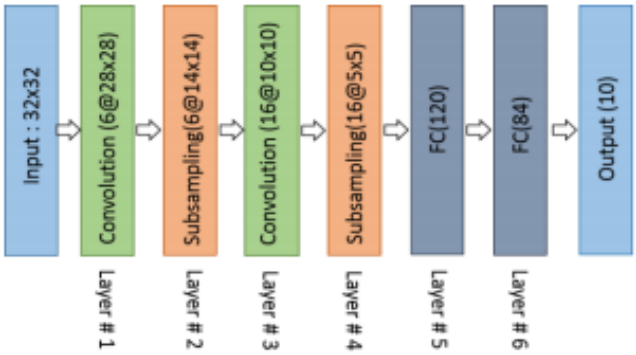

Figure 2. Architecture of LeNet-5. Reprinted from “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches” by Alom et al., 2018, retrieved from https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf Copyright 2018 by Alom et al.

As shown in Figure 2 (Alom et al., 2018), “the basic configuration of LeNet-5 consists of two convolutional layers, two sub-sampling layers, two fully connected layers, and an output layer with Gaussian connection. The input is a 32 x 32 pixel image” (LeCun et al., 1998).

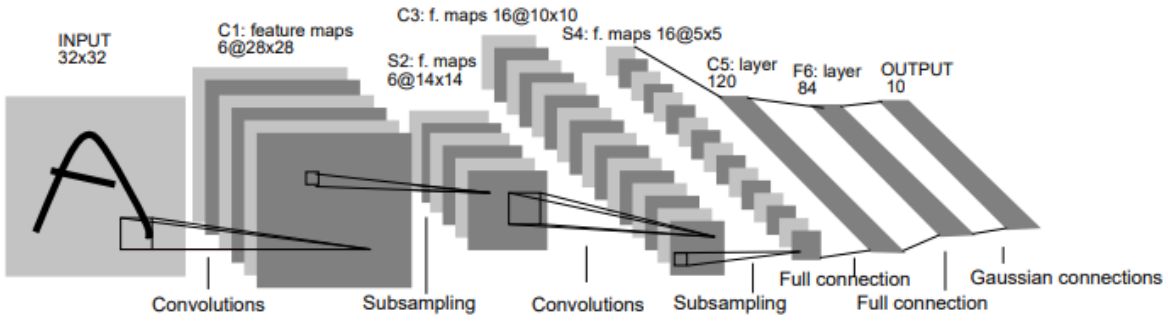

Figure 3. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical. Reprinted from "Gradient-based Learning Applied to Document Recognition" by Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, 1998, *Proceedings of the IEEE, 86,* p. 2285. Copyright 1998 by LeCun et al.

LeCun et al. (1998) built LeNet-5 to recognize characters, see Figure 3 (LeCun et al., 1998). The input of the network is an image of characters that is size-normalized and centered. According to LeCun et al. (1998), "receives each unit in a layer inputs from a set of units located in a small neighbourhood in the previous layer". The units in a layer are organised in so-called "planes". Every unit in each plane has an identical assemblage of weights. The outputs of the units in a plane are addressed as a "feature map". Every unit in a feature map performs the same activity on divergent components of the picture. A thorough convolutional layer contains various feature maps. Every feature map has a distinctive weight vector, which means that different features can be derived at every position. An example of a complete convolutional layer is shown in the first layer of LeNet-5 in Figure 3:

> Units in the first hidden layer of LeNet-5 are organised in six planes, each of which is a feature map. A unit in a feature map has 25 inputs connected to a 5 by 5 area in the input, called the receptive field of the unit. Each unit has 25 inputs, and therefore 25 trainable coefficients plus a trainable bias (LeCun et al., 1998).

2.2.2 AlexNET

In 2012, Krizhevsky et al., proposed AlexNET, which is a more advanced model in comparison to LeNet-5. With AlexNET they won the 2012 ImageNet challenge for visual object recognition. "AlexNET accomplished state-of-the-art recognition accuracy compared to the traditional machine learning and computer vision approaches" (Alom et al., 2018). This was a big development within the field of machine learning and computer vision for recognition and classification tasks and demarcated the march of convolutional neural networks in computer vision (Alom et al., 2018).

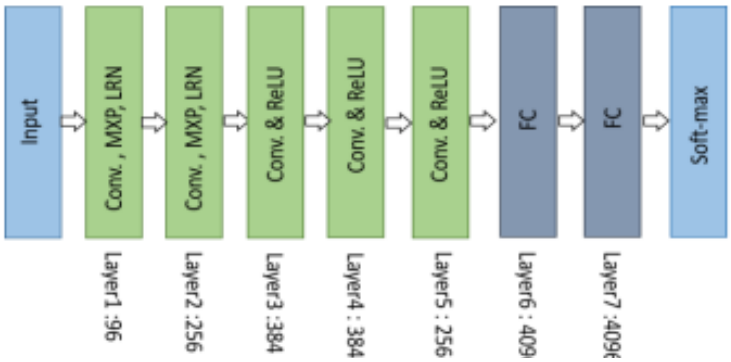

Figure 4. Architecture of AlexNET. Reprinted from "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches" by Alom et al., 2018, retrieved from https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf Copyright 2018 by Alom et al.

As shown in Figure 4 (Alom et al., 2018), AlexNET is a convolutional neural network which consists of eight layers. According to Krizhevsky et al. (2012), AlexNET consists of "five convolutional and three fully connected layers". After every convolutional and fully connected layer a ReLU is applied to add non-linearity. This intensifies the speed. Before the first and the second fully connected layer dropout is applied to deal with overfitting. The network has approximately 60 million parameters (Krizhevsky et al., 2012).

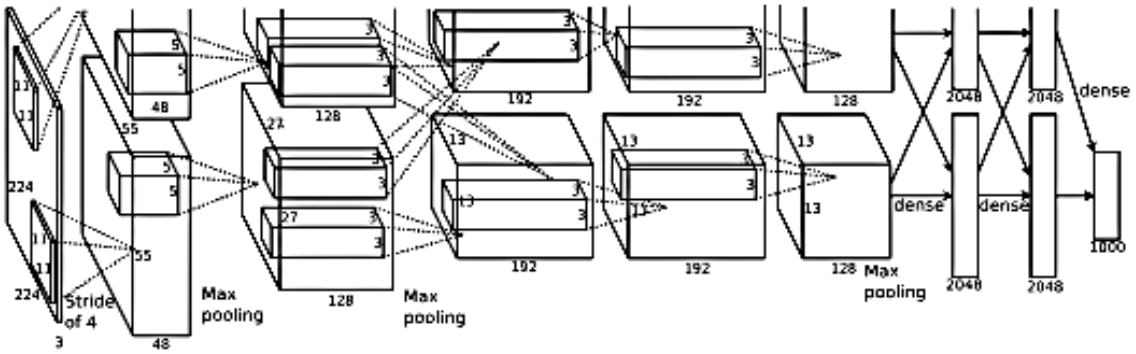

Figure 5. An illustration of the architecture of AlexNET. Reprinted from "ImageNet Classification with Deep Convolutional Neural Networks" by A. Krishevsky, I. Sutskever, and G. E. Hinton, 2012, *Proceedings of the 25th International Conference on Neural Information Processing Systems, 1,* p. 1102. Copyright 2012 by Curran Associates Inc.

The result of the final fully connected layer produces a distribution over the 1000 class labels, as shown in Figure 5 (Krizhevsky et al., 2012), which is fed to a 1000-way softmax. Krizhevsky et al. (2012) describe the construction of AlexNET as follows:

> The kernels of the second, fourth, and fifth convolutional layer are linked to all kernel maps in the second layer. The neurons in the fully connected layers are linked to all neurons in the preceding layer. Response-normalization layers follow the first and second convolutional layers. Max-pooling layers follow the response-normalization layers and the fifth convolutional layer. The first convolutional layer filters the 224 x 224 x 3 input image with 96 kernels of size

11 x 11 x 3 with a stride of 4 pixels. The input of the second convolutional layer is the output of the first convolutional layer and filters it with 256 kernels of size 5 x 5 x 48. The third, fourth, and fifth convolutional layers are connected to each another without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size 3 x 3 x 256 and is connected to the outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size 3 x 3 x 192, and the fifth convolutional layer has 256 kernels of size 3 x 3 x 192. The fully connected layers have each 4096 neurons.

2.2.3 ZFNet

The winner of the ImageNet Large Scale Visual Recognition Competition in 2013 was ZFNet, designed by Zeiler and Fergus. This network achieved a top-5 error rate of 14.8%. This was achieved by adjusting the hyper parameters of AlexNET, while maintaining the same structure with additional deep learning elements (Das, 2017). The difference between the approach used by Krizhevsky et al. (2012) and the approach by Zeiler and Fergus (2013) is that the sparse connections applied in the layers are replaced with dense connections (Zeiler & Fergus, 2013). The ZFNet model by Zeiler and Fergus (2013) was trained on the same dataset as AlexNET, the ImageNet 2012 training set.

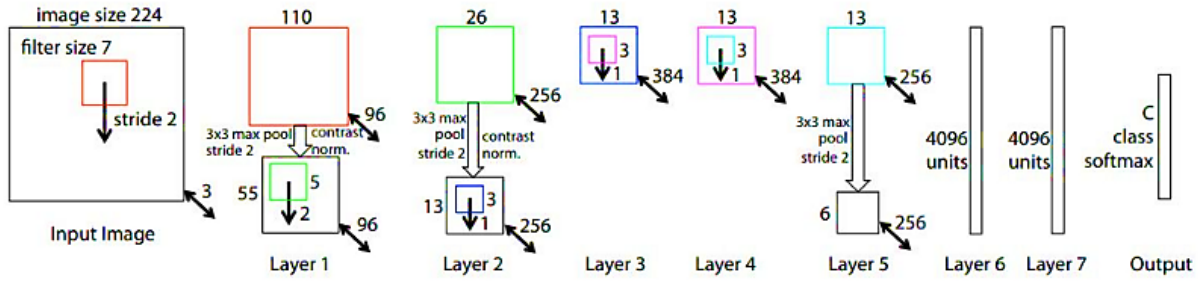

Figure 6. Architecture of ZFNet. Reprinted from "Visualizing and Understanding Convolutional Networks" by M. D. Zeiler and R. Fergus, 2013, retrieved from https://arxiv.org/pdf/1311.2901.pdf Copyright by Zeiler and Fergus.

As shown in Figure 6, the input of the ZFNet model is a 224 by 224 crop of an image. According to Zeiler and Fergus (2013):

> This is convolved with 96 different first layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function, this is not shown in the figure, (ii) pooled (max within 3 x 3 regions, using stride 2), and (iii) contrast normalised across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2, 3, 4, 5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form. The final layer is a *C*-way sofmax function, *C* being the number of classes. All filters and feature maps are square in shape.

2.2.4 GoogLeNet

The winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014 was GoogLeNet. GoogLeNet is also known as Inception V1, from Google. GoogLeNet is a network introduced by Christian Szegedy of Google. The model aims to reduce computation complexity compared to the traditional convolutional neural network (Alom et al., 2018). GoogLeNet uses a convolutional neural network inspired by LeNet, but implements a new element; the inception module. This inception element is based on several small convolutions to reduce the quantity of required parameters (Das, 2017). The proposed method of Szegedy (2015) was to incorporate so-called inception layers. These layers have fluctuating receptive fields, which are constructed by divergent kernel sizes (Alom et al., 2018). GoogLeNet was designed to be computational efficient and practical. This means that "inference can be run on individual devices, including those with limited computational resources" (Szegedy et al., 2015).

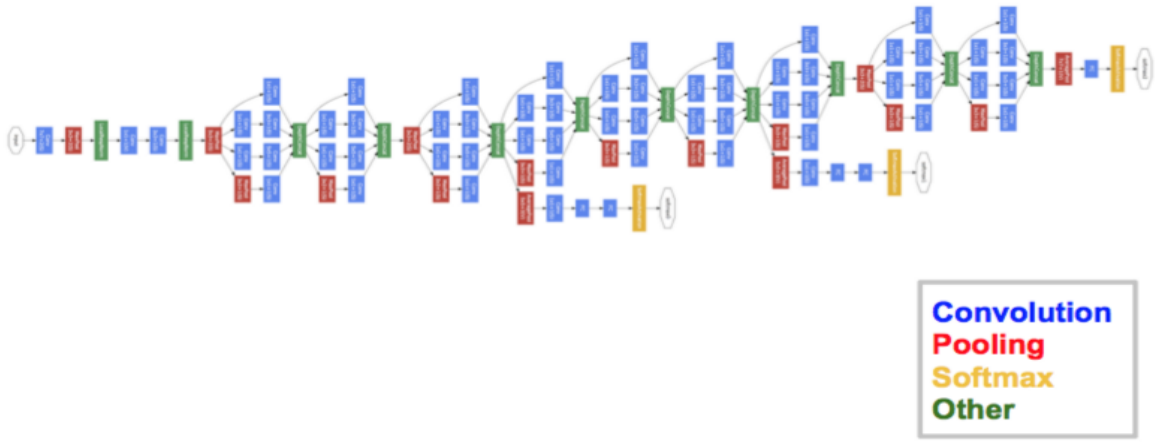

Figure 7. Architecture of GoogLeNet. Reprinted from *Medium* website, by S. Das, 2017, retrieved from https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5 Copyright 2017 by Medium.

As shown in Figure 7, GoogLeNet consists of 22 layers, adding up just the layers which include parameters, which makes it the biggest network to date. However, "the number of parameters used by GoogLeNet is lower than the prior network AlexNET" (Alom et al., 2018), from 60 million (AlexNET) to 4 million (Das, 2017).

2.2.5 VGGNet

The runner-up at the ILSVRC in 2014 is VGGNet, a network developed by Simonyan and Zisserman. As shown in Figure 10 (Das, 2017), this network contains 16 convolutional layers and is attractive because it has a uniform structure. VGGNet has only 3 x 3 convolutions, which is comparable to AlexNET, however VGGNet has a large number of filters. VGGNet consists of 138 million parameters (Das, 2017).

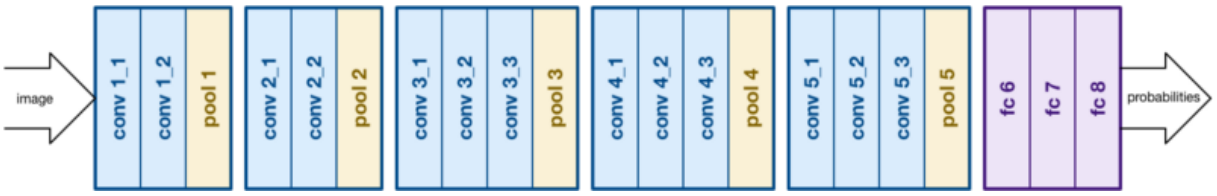

Figure 8. Architecture of VGGNet. Reprinted from *Medium* website, by S. Das, 2017, retrieved from https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5 Copyright 2017 by Medium.

The input to VGGNet is an image of size 224 x 224 RGB. This image is processed throughout a bundle of convolutional layers, in which filters are practiced with little receptive fields: 3 x 3. "A stack of convolutional layers is followed by three fully connected layers. The first two fully connected layers have 4096 channels each, the third layer contains 10000 channels. The final layer is the soft-max layer" (Simonyan & Zisserman, 2015).

2.2.6 Inception Network

GoogLeNet was the initial version of this architecture, see subchapter 2.2.4, but "subsequent manifestations have been called Inception vN, where N refers to the version number put out by Google" (Rosebrock, 2017). The Inception V3 architecture origins from a later publication by Szegedy, Vanhoucke, Ioffe and Shlens (2015). This publication introduces several modernisations to the inception module (Rosebrock, 2017).

The Inception architecture was first introduced by Szegedy et al. (2015). According to Rosebrock (2017), the goal of the inception module is as follows:

> The inception module aims to act as a multi-level feature extractor by computing 1 x 1, 3 x 3 and 5 x 5 convolutions within the same module of the network. The outputs of these filters are stacked along the channel dimension and before being fed into the next layer in the network.

Szegedy et al. (2015) state that "the inception architecture allows for increasing the number of units at each stage without an uncontrolled blow-up in computational complexity at later stages". This can be achieved "by using dimensionality reduction prior to expensive convolutions with larger patch sizes. Furthermore, the design follows the intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from the different scales simultaneously" (Szegedy et al., 2015).

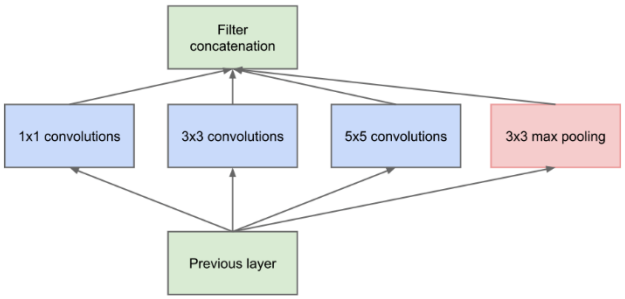

Figure 9. Inception module, naïve version. Reprinted from "Going Deeper with Convolutions" by Szegedy et al., 2015, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 4. Copyright 2015 by Szegedy et al.

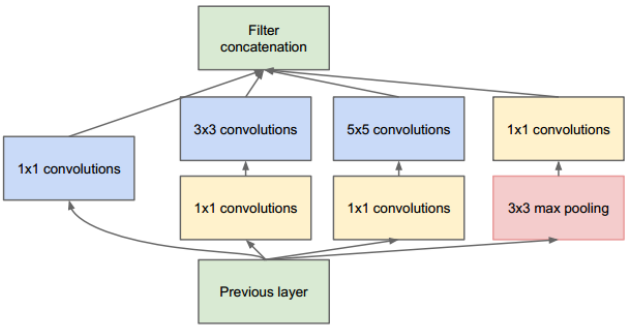

Figure 10. Inception module with dimensionality reduction. Reprinted from "Going Deeper with Convolutions" by Szegedy et al., 2015, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* p. 4. Copyright 2015 by Szegedy et al.

The main idea of the Inception architecture, as shown in Figure 8 (Szegedy et al., 2015), is to "consider how an optimal local sparse structure of a convolution vision network can be approximated and covered by readily available dense components" (Szegedy et al., 2015). According to Szegedy et al. (2015), there is one problem with the Inception module as mentioned, the naïve form: "5x5 convolutions can be expensive on top of a convolutional layer with a large number of filters. This becomes more visible once pooling units are added: the number of output filters equals to the number of filters in the previous stage". This may lead to a growth in the number of outputs, which makes it an inefficient architecture that eventually leads to a computational blow up within a few phases. This potential threat of computational blow up leads to another idea of the Inception architecture: the module with dimensionality reduction, as shown in Figure 9 (Szegedy et al., 2015). According to Szegedy et al. (2015):

> This module is based on embeddings: even low dimensional embeddings might contain a lot of information about a relatively large image patch. However, embeddings represent information in a dense, compressed form, which makes it harder to process. The representation should be kept sparse at most places and only compress the signals whenever they have to be aggregated. That means, 1 x 1 convolutions are used to compute reduction before the expensive 3 x 3 and 5 x 5 convolutions.

2.3 Determining the best deep learning architecture

Over the years, the quality of network architectures significantly improved by utilizing deeper and wider networks. Architectural improvements in deep convolutional architecture can be used for improving performance for other computer vision tasks that rely on high quality, learned visual features. Furthermore, improvements in the quality of the network resulted in new application domains for convolutional networks in cases where, for example, AlexNET features could not compete with hand

engineered, crafted solutions (Szegedy et al., 2015). An overview of the various convolutional neural network architectures that have been proposed over the years, is demonstrated in Table 1 (Das, 2017).

Table 1
Summary table

Year	Convolutional Neural Network	Developed by	Place ILSVRC	Top-5 error rate	Number of parameters
1998	LeNet	LeCun et al.			60 thousand
2012	AlexNET	Krizhevsky et al.	1st	15.3%	60 million
2013	ZFNet	Zeiler & Fergus	1st	14.8%	
2014	GoogLeNet	Google	1st	6.67%	4 million
2014	VGGNet	Simonyan & Zisserman	2nd	7.3%	138 million
2015	Inception Network	Szegedy et al.		5.6%	4 million

Note. Adapted from *Medium* website, by S. Das, 2017, retrieved from https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5 Copyright 2017 by Medium.

The networks mentioned in Table 1 (Das, 2017) all have their advantages and disadvantages. For example, although VGGNet has the advantage of having a simplistic architecture, this comes at a high cost. The fact is that evaluating this particular network requires a lot of computation. Therefore, GoogLeNet was designed in 2014 to perform well even under strict constraints on memory and computational budget. As to be seen in Table 1, GoogLeNet and the Inception Network employ around 4 million parameters, which is much less than its predecessor AlexNET, which uses around 60 million parameters, but also less than VGGNet, which uses around 138 million parameters. The computational cost of the Inception Network is also lower than its predecessors. This makes it feasible to use Inception Networks in big data scenarios, where a lot of data needs to be processed at reasonable cost or scenarios where memory or computational capacity is limited (Szegedy et al., 2015).

The network used for the experiments in our thesis, has been selected on two criteria. Namely, the quantity of parameters which reside within the network, and the error rate. Therefore, the network that has been chosen for this experiment, is the Inception Network: the network with the smallest number of parameters (4 million), and the network with the lowest error rate (5.6%) compared to the other convolutional neural networks. The experiment in our thesis goes further than the experiment by Van Noord et al. (2015), who use a relatively old network, i.e. AlexNET, and uses a new and better, with less parameters, convolutional neural network: The Inception Network.

Chapter 3. Related work on automated artist attribution

This chapter describes related work in the field in order to place this thesis in a broader context and to state how this research relates to other studies completed in the field. This chapter is divided in two sections. Section 3.1 introduces the subject on a more general level and mentions how the field of using automated methods to analyse art and cultural traditions has developed in the course of time. Section 3.2 discusses related work on automated artist attribution in a chronological structure.

3.1 Computational techniques over the years

In order to authenticate and data artworks, art historians make use of a variety of methods. Examples of these methods are documentary research or categorizing painting styles and techniques. However, over the years, art analysts became more and more engrossed in automated analysis strategies. According to Li et al. (2011), "some of them believe that computers can extract certain patterns from images more thoroughly than is possible when through manual attempts., can process a larger number of paintings and are less subjective". In recent years, several researches on studying art and cultural heritages by means of computational techniques have emerged (Li et al., 2011). An example of these computational techniques is the discovery of x-rays, shortly after the 19th century. Researcher started to use these rays to reveal underdrawings and pentimenti, an alteration in a painting. Later, other techniques, such as infra-red photography and reflectography were used to achieve comparable results. With these methods, the output image is interpreted by an art expert. However, some of the image interpretation relies nowadays on algorithms developed from computer vision. Computers are able to analyse certain aspects of perspective, lighting, colour, or brushstrokes better than a trained art expert or artist (Stork, 2009).

3.2 Overview of automated artist attribution

Sablatnig, Kammerer, and Zolda (1998) are one of the first researchers to investigate the personal style of an artist. They argue that it is challenging to attribute artworks to an artist: "methods like X-ray and infra-red diagnosis, or digital radiography do not relate characteristics of an artwork to a specific artist and his personal style" (Sablatnig et al., 1998). To be able to examine this individual way of expression, the authors examined the "structural signature" relying on brushstrokes in portrait miniatures. Therefore, they developed a system that recognizes portrait miniatures by means of a computer-aided classification. This system facilitates a semi-automatic classification based on brushstrokes. The classification is separated into three aspects: "colour, shape of region, and structure of brushstrokes" (Sablatnig et al., 1998). In 1999, Taylor, Micolich, and Jonas analysed the drip paintings of Jackson Pollock using fractal techniques. Pollock dripped paint from a can onto a vast canvas rolled out across the floor of his barn. Taylor et al. (1999) found, through the use of computer techniques, that Pollock's seemingly random splatters of paint actually have characteristic fractal dimension values. These fractal dimension values increase slightly over the course of Pollock's career (Taylor et al., 1999). Lyu, Rockmore, and Farid (2004) describe a computational tool to authenticate artworks. They focus on paintings and drawings,

which are represented by digital scans of high quality of original works. “The technique looks for consistencies or inconsistencies in the first- and higher-order wavelet statistics collected from artworks” (Lyu et al., 2004). They apply their analysis to an assemblage of 13 drawings which have once been attributed to Pieter Bruegel the Elder. Furthermore, they demonstrate a “many-hands” inquiry of a portrait by Perugino, a painter from the Renaissance. Thereafter, a summary is given of the techniques used for the inquiry. This summary also contains a description of the underlying statistical model (Lyu et al., 2004). According to Johnson et al. (2008), “image processing tools are aimed at helping art historians currently in the earliest stages of development”. Partly, this results from data not being widely accessible. To stimulate the evolution of methods like these, the Van Gogh Museum and Kröller-Müller Museum in the Netherlands published a dataset of 101 gray-scale scans of paintings in their collections, which is accessible to researchers in the field of image processing from various universities (Johnson et al., 2008). This dataset is used by Johnson et al. (2008) to analyse the brushstrokes of the paintings of Vincent Van Gogh. The analysis of brushstrokes consists of different steps. Firstly, when evaluating the brushwork of a painting, it is important to decide which part of the painting should be reduced, since they may not be painted by the author itself. The next step is to describe the characteristics of the original brushwork, which are observed across the remainder of the painting. An important characteristic to pay attention to when examining a painting is the frequent use of specific brushstroke styles. For example, the “elbow-strokes” or the “brickwork” patterns of Van Gogh (Johnson et al., 2008). This analysis has been done on a small scale of just 101 images with full resolution reproductions as input (Strezoski & Worring, 2017). Therefore, Johnson et al. (2008) conclude that brushstroke analysis is helpful in artist attribution, but perfect results have not been obtained yet. Using a wider range of analysis tools, better results can be achieved. This can be obtained by using richer representations of the paintings, and more nuanced mathematical models. Hughes et al. (2010) describe a technique for the quantification of styles of art which uses a model that includes sparse coding. According to Hughes et al. (2010), “sparse coding models can be trained to represent any image space by maximizing the kurtosis of a representation of a randomly selected image from that space”. The authors use this technique to distinguish an assemblage of drawings by Pieter Bruegel the Elder from an assemblage of imitations of Bruegel’s paintings (Hughes et al., 2010). A few years later, in 2014, Mensink and Van Gemert introduced the Rijksmuseum Challenge: Museum-Centered Visual Recognition. This research contains a contest for classification and content-based retrieval of artworks. The dataset used for this research is a dataset of art objects, all of which are displayed in the Rijksmuseum in Amsterdam, the Netherlands. The artworks in this dataset origin from aged periods to the nineteenth century. Mensink and Van Gemert (2014) propose four challenges: “(i) predict the artist, (ii) predict the art-type, (iii) predict the used material, and (iv) predict the creation year”. One of the challenges is to predict the artist of an artwork given a particular image. According to Mensink and Van Gemert (2014), “this is a multi-class problem where each object has a single creator. Performance is measured as the weighted mean class accuracy. This ensures that the classification performance of an artist with only a few works accounts as much as an artist with more artworks”.

3.3 The Van Noord et al. (2015) study

The Rijksmuseum Challenge dataset is not only used by Mensink and Van Gemert (2014), but also by Van Noord et al. (2015), who perform artist attribution using their own subsets with a convolutional neural network named PigeoNET. Van Noord et al. (2015) argue that "to ensure that the visual characteristics on which the task is solved by PigeoNET make sense, human experts are needed to assess the relevance of the acquired mapping from images of artworks to artists". Van Noord et al. (2015) also mention that "although the Rijksmuseum Challenge dataset is the largest available dataset containing digital reproductions of artworks, it does suffer from two limitations". Firstly, they mention that is not clear in what way the "controlled conditions" were determined for various works of art. Each differentiation in the way the photo is made, for example the perspective or type of camera, may be picked up by PigeoNET. The second limitation involves the labelling of works of art. The Rijksmuseum Challenge dataset just mentions one artist, where the Rijksmuseum catalogue mentions numerous contributions. This might cause doubt about whether the attribution of works of art in the Rijksmuseum challenge dataset has been executed correctly (Van Noord et al., 2015). Furthermore, Van Noord et al. (2015) state that "the number of artists and the number of examples per artist have a very strong influence on the performance". Therefore, they suggest that in order to improve the performance, the dataset has to be expanded.

Chapter 4. Method

This chapter describes the method used for this research. This chapter is divided in four sections. Section 4.1 describes the dataset in detail, section 4.2 consists of a description of wat has been done on pre-processing of the data, section 4.3 consists of a description of the actual implementation, and section 4.4 consists of a description of the experimental procedure and evaluation criteria.

4.1 Dataset description

The dataset that is used for the task of automatically recognizing artists by their artworks, is the Rijksmuseum Challenge dataset (Mensink & Van Gemert, 2014). This dataset contains 112.039 digital photos of works of art by 6.629 artisans which are all shown in the Rijksmuseum in Amsterdam, the Netherlands. According to Van Noord et al. (2015), "all artworks were digitised under controlled settings". This dataset contains 1.824 contrasting categories of works of art and 406 annotated materials, like paper, canvas, porcelain, iron and wood (Van Noord et al., 2015). "The artworks in this dataset date from ancient times, medieval ages and the late 19th century" (Mensink & Van Gemert, 2014).

Van Noord et al. (2015) defined two types of subsets for the purpose of their experiment:

> Type A (for "All") and type P (for "Prints"). For the heterogeneous subset of at least 256 artworks of type A, Table 2 provides a more detailed listing which specifies the three most outstanding types: Prints, Drawings, and Other. The Other category includes a variety of different artwork types, including 35 paintings.

As follows from table 2 (Van Noord et al., 2015), the most common artwork in the Rijksmuseum Challenge dataset is prints. This approach by Van Noord et al. (2015) has been used for this thesis.

Table 2
List of the 34 artists with at least 256 artworks and the distribution of artworks over main types (prints, drawings, and other).

#	Name	Prints	Drawings	Other
1	Heinrich Aldegrever	347	27	
2	Ernst Willem Jan Bagelaar	400	27	
3	Boëtius Adamsz. Bolswert	592		
4	Schelte Adamsz. Bolswert	398		
5	Anthonie Van Den Bos	531	3	
6	Nicolaes De Bruyn	515	2	
7	Jacques Callot	1,008	4	1
8	Adriaen Collaert	648	1	
9	Albrecht Dürer	480	9	2
10	Simon Fokke	1,177	90	
11	Jacob Folkema	437	4	3
12	Simon Frisius	396		
13	Cornelis Galle I	421		
14	Philips Galle	838		
15	Jacob De Gheyn II	808	75	10
16	Hendrick Goltzius	763	43	4
17	Frans Hogenberg	636		4
18	Romeyn De Hooghe	1,109	5	5
19	Jacob Hourbraken	1,105	42	1
20	Pieter De Jode II	409	1	
21	Jean Lepautre	559		1
22	Caspar Luyken	359	18	
23	Jan Luyken	1,895	33	
24	Jacob Ernst Marcus	372	23	2
25	Jacob Matham	546	4	
26	Meissener Porzellan Manufakter			1,003
27	Pieter Nolpe	344	2	
28	Crispijn Van De Passe I	841	15	
29	Jan Caspar Philips	401	17	
30	Bernard Picart	1,369	132	3
31	Marcantonio Raimondi	448	2	
32	Rembrandt Harmensz. Van Rijn	1,236	119	29
33	Johann Sadeler I	578	1	
34	Reinier Vinkeles	573	50	

Note. Reprinted from "Towards Discovery of the Artist's Style: Learning to Recognise Artists by their Artworks" by N. Van Noord, E. Hendriks, and E. Postma, 2015, *IEEE Signal Processing Magazine,* p. 50. Copyright 2015 by IEEE.

4.2 Pre-processing of the data

The subset as shown in Table 2 is used for our experiments. Conform the procedure by Mensink and Van Gemert (2014) and Van Noord et al. (2015), the dataset is arbitrarily separated in three different sections: the train set, validation set, and test set. These three sections have been constructed with three different goals in mind: to train classification models, to tune hyper parameters of the models, and to compute the functioning of the models. The train set contains 17.026 images, the validation set contains 2.436 images, and the test set contains 4.850 images.

This study uses the ImageNet pre-trained model Inception V3 from Keras. Keras is a software environment, written in Python, for performing deep learning experiments. Keras is designed as component of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Keras focuses on empowering fast experimentation. The primary author of Keras is François Chollet, a Google engineer (Keras Documentation, n.d.).

The pre-processing procedure of the data is conform the procedure by Krizhevsky et al. (2012), which is also used by Van Noord et al. (2015). The images are down-sampled to a rigid resolution of 256 x 256. When the input is rectangular of shape, the picture is resized such that the shorter side is of length 256. Thereafter, the central 256 x 256 patch is cropped out from the resulting image (Krizhevsky et al., 2012).

To prevent the data from overfitting, a data augmentation procedure is applied, conform the procedure used by Van Noord et al. (2015). This procedure resides of random crops and horizontal reflections. To create a bigger sample size, horizontal reflections were applied. This procedure doubles the quantity of training data. We experimented with various settings to the basic architecture. We examined the inclusion of dropout layers, varied the number of densely (fully) connected layers on top of the pre-trained Inception V3 architecture, retrained different parts of the Inception V3 architecture, and used a two-phase or single-phase training, as shown in Table 3.

4.3 Description of the actual implementation

The experiments are executed using the programming language Python. To complete the task of automated artist attribution, Keras is used together with TensorFlow in Python. Within Keras, the convolutional neural network Inception V3 is used, from which the fully connected layer at the output is removed.

4.4 Description of the experimental procedure and evaluation criteria

As Van Noord et al. (2015) mention, “the objective of the artist attribution task is to identify the correct artist for each unseen artwork in the test set”. Therefore, the functioning of the model is assessed by

means of the accuracy rate. Where Van Noord et al. (2015) evaluate their network by taking five patches, one in the centre of the 256 x 256 crop, and four in the corner, the experiments in our thesis use single random crops for evaluation. The motivation for doing so, is that it better reflects the way the network is trained.

To perform the task of artist attribution, six experiments have been executed, see Table 3, to ultimately achieve the highest test performance. The models used for the experiments consists of several layers. On top of the pre-trained base model, a global average pooling layer is added, after which a fully connected layer is added (dense relu). These two layers are equivalent for all six experiments. In experiment 1 and 2, a dropout layer is added to reduce overfitting.

The first four experiments are trained in two phases, accordingly to the Keras documentation on the Inception V3 architecture[1]. The first phase consists of 100 epochs, of which only the top is trained. In this phase all the convolutional Inception V3 layers are frozen. After this phase, the top layers are well trained, which means the convolutional layers from the Inception V3 architecture can be fine-tuned. The bottom layers are frozen, and the remaining top layers are trained. The second phase consists of also of 100 epochs, of which the first 249 layers are fixed (200 layers in the fourth experiment), and the rest is trainable.

The fifth and sixth experiment are executed in a different manner compared to the first four experiments, and thus differ from the Keras documentation. These last two experiments are trained in one phase (single-phase training). In the fifth experiment, the first 100 layers are fixed, the rest is trainable. In the sixth experiment, the first 50 layers are fixed, the rest is trainable. As follows from Table 3, the less layers that are fixed, the higher the test performance. However, due to computer capacity it was not feasible to train with more fixed layers.

[1] See https://keras.io/applications/#usage-examples-for-image-classification-models ("Fine-tune InceptionV3 on a new set of classes")

Chapter 5. Results

Within this chapter, the results that follow from the experiments that were executed in order to answer the research question of this thesis are examined.

Table 3 shows the results of the task of artist attribution. Six experiments, each with different settings, described in section 4.4, are executed in our thesis to ultimately achieve the highest test performance. The first four experiments are based on two phases of training, the last two train (almost) the entire Inception network in one 100-epoch phase. The results of the experiment by Van Noord et al. (2015) are also mentioned in Table 3, however the layers and what kind of training they performed is unknown.

Table 3

Overview of the six experiments executed in our thesis.

Experiment	**Layers**	**Initialisation/training**	**Test Performance**
1	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Dropout (0.5) 5. Softmax	Two-phase training 1: 100 epochs, only top 2: 100 epochs, first 249 layers fixed, rest trainable. SGD (lr=0.01, momentum=0.9)	0,802
2	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Dropout (0.5) 5. Dense relu (1024) 6. Dropout (0.5) 7. Softmax	Two-phase training 1: 100 epochs, only top 2: 100 epochs, first 249 layers fixed, rest trainable. SGD (lr=0.01, momentum=0.9)	0,787
3	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Softmax	Two-phase training 1: 100 epochs, only top 2: 100 epochs, first 249 layers fixed, rest trainable. SGD (lr=0.01, momentum=0.9)	0.792
4	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Softmax	Two-phase training 1: 100 epochs, only top 2: 100 epochs, first 200 layers fixed, rest trainable. SGD (lr=0.01, momentum=0.9)	0,830
5	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Softmax	Single-phase training First 100 layers fixed, rest trainable	0,857
6	1. Inception V3 base 2. Global Average Pooling 3. Dense relu (1024) 4. Softmax	Single-phase training First 50 layers fixed, rest trainable	0,868
Van Noord et al. (2015)			0,783

As follows from Table 3, the test performances of the experiments increase when going further with the experiments. The final experiment, experiment 6, performs therefore better than the first experiment. Moreover, it can be said that training in one phase results in a better test performance, than training in two phases.

All configurations result in an accuracy score of 78.7% or more. The accuracy score for the artist attribution task, executed by Van Noord et al. (2015), with PigeoNET, a variant of AlexNET, is 78.3%. Moreover, the sixth experiment shows an accuracy rate of 86.8%, which is a big improvement compared to the accuracy rate of Van Noord et al. (2015). This means that executing the task of artist attribution with a current convolutional neural network improves the accuracy of an experiment in author attribution considerably. There is only one caveat, which is the slightly different way of evaluating the performance. As mentioned in section 4.4, we use randomly selected patches, rather than taking the average performance of the central patch and the four corner patches, to evaluate the performance. This may have affected the performance slightly.

68	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2
0	251	2	0	0	1	0	0	0	2	1	1	0	1	0	2	0	0	0	0	0	2	2	3	0	0	1	3	2	0	1	1	0	1
0	5	61	0	0	0	0	0	0	4	4	1	0	1	0	3	1	0	0	0	0	3	0	2	0	0	1	0	0	0	1	1	0	0
0	1	1	49	0	2	0	0	1	3	0	0	0	0	2	2	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0
0	1	0	0	46	6	0	1	0	1	0	0	2	0	0	0	0	0	0	1	0	1	3	2	0	0	7	0	2	0	0	0	2	0
0	0	0	0	1	141	0	0	2	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	7	2
0	0	0	0	0	0	150	7	0	1	0	0	0	1	0	1	0	2	0	0	3	0	0	0	1	1	1	0	0	1	0	0	0	0
0	0	0	1	0	4	4	121	0	1	0	0	0	1	0	0	2	2	0	0	7	0	2	0	0	0	2	1	0	0	4	0	1	0
0	0	0	0	0	1	0	0	109	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	224	0	1	1	0	0	5	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	1
0	2	3	0	1	0	0	0	0	0	221	1	3	0	0	1	1	0	0	0	0	0	1	2	0	0	1	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	77	0	0	0	2	3	0	0	0	0	0	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	1	64	0	0	0	0	2	0	0	1	0	1	3	0	0	0	0	0	0	0	0	1	0
1	0	0	1	2	0	0	0	0	2	0	0	0	101	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	54	24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	2	0	1	1	0	1	1	0	7	0	0	0	0	12	320	1	1	0	0	0	0	1	0	1	1	0	3	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	68	0	0	0	0	1	0	4	0	2	0	2	0	2	0	0	0	0
0	0	0	0	1	1	1	3	0	0	0	0	3	0	0	0	0	100	0	0	7	0	1	1	1	0	3	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	0	0	0	0	1	0	0	0	0	1	1	2	3	0	0	0	47	0	0	0	2	1	0	1	1	3	0	0	1	0	0
0	0	0	2	0	0	2	1	1	0	0	1	0	0	0	0	0	6	0	0	142	0	0	0	1	0	8	0	0	0	0	0	0	2
0	11	6	1	0	0	0	0	0	0	2	0	0	0	0	1	2	0	0	0	0	51	0	3	0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1	3	0	0	2	0	0	2	0	2	0	0	3	0	108	1	0	0	0	0	1	0	0	0	0	1
1	8	5	2	1	0	1	0	0	4	5	0	0	0	0	7	1	1	0	0	0	2	1	268	0	0	3	0	0	1	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	86	0	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	267	0	0	0	2	0	0	0	0
0	0	0	1	1	3	0	3	0	0	0	0	2	0	0	0	0	2	0	0	6	1	1	0	1	0	97	0	0	0	0	0	6	1
0	6	1	0	0	0	0	0	0	0	1	0	0	0	0	0	7	0	0	0	1	0	0	1	0	1	0	105	0	1	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	61	0	0	0	1	0
0	3	0	1	0	0	2	2	0	1	0	7	0	0	0	3	1	0	0	0	0	0	0	1	0	3	0	0	0	71	0	0	0	0
0	0	0	0	0	1	1	3	0	0	0	0	0	0	0	0	0	1	0	0	4	0	1	0	0	0	4	0	0	0	98	0	2	0
0	1	0	2	0	0	2	1	0	3	0	0	0	1	0	2	0	0	0	0	0	1	2	1	1	1	2	1	0	1	0	176	1	0
0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	126	0
7	0	0	0	0	0	0	4	0	3	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	5	0	0	0	0	0	2	0	91

Figure 11. Confusion matrix for all artists with at least 256 training examples of all artwork types. The rows represent the actual artists and the columns the artist estimates. This confusion matrix represents experiment 6.

Figure 11 displays a visual representation of the confusion matrix for that part of the dataset that contains at least 256 examples of all types of works of art, see Table 2 for this subset. This confusion matrix represents experiment 6, which is the experiment with the highest test performance. The rows and columns correlate to the artisans mentioned in Table 2. The rows symbolize the actual artists, the columns the artist estimates by the Inception V3 network. From top to bottom, the matrix has 34 rows,

which is in accordance with the subset used for the experiments, see Table 2. The diagonal line represents correct attributions. The confusion matrixes for the other five experiments are included in the appendix. In order to make the confusion matrixes readable, which was not the case in the Python files, they are converted to csv-files. As a consequence, the coloured diagonal line, which is normally visible in the Python files, disappears.

As shown in Figure 11, the highest number of correct attributions for the subset mentioned in Table 2, is 320. This number corresponds with the 16th artist in the subset mentioned in Table 2, which is Hendrick Goltzius. The lowest number of correct attributions is 47, which corresponds to Pieter De Jode II, the 20th artist in the subset mentioned in Table 2. These results differ from the results by Van Noord et al. (2015). Meissener Porzellan Manufakter was noticed to have the best artist-specific accuracy, the worst artist-specific classification accuracy was assigned to Schelte Bolswert (Van Noord et al., 2015).

Chapter 6. Discussion

In this chapter, the results of the experiments are evaluated with regard to the problem statement listed in the introduction. Section 6.2 consists of the limitation of this research, and section 6.3 introduces ideas for future work on the topic of automated artist attribution.

6.1 Goal of the experiment

The goal of the experiments in our thesis was to investigate to what extent using a current convolutional neural network improves the results of an experiment in artist attribution. In order to compare results with the research executed by Van Noord et al. (2015), the same dataset is used: the Rijksmuseum Challenge dataset.

Based on logical reasoning, the expectations of the experiments executed in our thesis were that the results would be improved when using a current network architecture. This is not only because network architectures have advanced over the past years, but also because several new network architectures have been proposed. As mentioned in section 2.3, the network used for the experiments, has been selected based on two criteria: the quantity of parameters in the network, and the error rate. Therefore, the network used for these experiments, is the Inception Network. In particular, the Inception V3 network.

The results of the experiments show that the accuracy score for the artist attribution task has improved by using a current network architecture, the Inception V3 network (2015). The highest accuracy score achieved out of the six experiments executed for our thesis, is 86.8%, instead of 78.3% when using PigeoNET, a network based on the older AlexNET (2012), as Van Noord et al. (2015) did. This result has been achieved by using almost the same approach by Van Noord et al. (2015), however we used a current convolutional neural network. Moreover, as mentioned in section 4.4, another difference between the approach by Van Noord et al. (2015) and our approach, is the way of evaluation. Where Van Noord et al. (2015) evaluate their network by taking five patches, one in the centre of the 256 x 256 crop, and four in the corner, the experiments in our thesis use single random crops for evaluation. The motivation for doing so, is that it better reflects the way the network is trained. Thus, however there is a slight difference in approach, there is also an improvement in results, compared to the results of Van Noord et al. (2015).

6.2 Limitations of the research

Firstly, the dataset that has been used for the experiments in our thesis is a heterogeneous dataset. The Rijksmuseum Challenge dataset contains images of various artworks, like paintings, but also porcelain, and wood. It is possible to identify the characteristics of an artist by examining works of art which are created by that artist. However, according to Van Noord et al. (2015), “obtaining such a large sample of

images is problematic, given the lack of (automatic) methods and criteria to determine whether an artwork is representative". A common method to avoid the requirement to have a decent sample is to use a big sample. This means, a collection of many images, and many images per artist. A dataset that matches with this description, is the Rijksmuseum Challenge dataset. Although the dataset is large, it does contain a great variety of artworks. This might influence the results of the experiment. Therefore, it might be better to use a homogeneous dataset for a similar experiment.

Secondly, as Van Noord et al. (2015) also mention, given the wide variety of types of artwork in the Rijksmuseum Challenge dataset, it is not clear in what way the "controlled conditions" are defined for various works of art. Every deviation in how the picture is made, like perspective or camera type, may be picked up by the network and influence the results. The ideal dataset for an experiment in automated artist attribution would therefore be a dataset without any visual marks. However, it is hard to create such a dataset on a scale like the Rijksmuseum Challenge dataset.

Thirdly, it is a fact that a computer maps pragmatically on authorship. They are trained to execute this task. Therefore, when changing for example pixels in an image, the computer will see this as a different image and subsequently not attribute the image to the right artist, as before the changing of pixels. Thus, the computer classifies differently than a human being. A human being would be able to see whether it is the same image, even when the pixels have been changed. Therefore, fully relying on a computer system with a task like author attribution, can be dangerous. Involvement of a human art expert is still required.

6.3 Future work

When taking the limitations of this research into consideration, there are a few recommendations to make for potentially future work. First of all, as mentioned in section 6.2, it might be useful to execute this experiment of automated artist attribution with a homogeneous dataset, instead of a heterogeneous dataset, as used for the experiments in our thesis. An example of a heterogeneous dataset is the dataset Painter by Numbers[2], which only exists of paintings, instead of other forms of artworks.

A second recommendation for future work can be made also within the topic of the dataset that has been chosen for the experiment of automated artist attribution. As mentioned in section 4.1, the dataset that is used for the experiments in our thesis, the Rijksmuseum Challenge dataset, is a dataset with images that show visual marks, like different perspectives. These differences in the images might be picked up by the network, and therefore may influence the results of the experiments. Therefore, it might be better to use a dataset with images where no visual marks are present.

[2] Available on Kaggle.com

Chapter 7. Conclusion

In this thesis a learning system is evaluated to assess to what extent the results of an experiment in automated artist attribution can be improved by using a current network architecture. The results of the experiments in this thesis are compared to the results by Van Noord et al. (2015), who used a less modern network architecture, PigeoNET, based on AlexNET. Six experiments have been executed to ultimately achieve the highest test performance. The approach for these experiments is the same as the approach used by Van Noord et al. (2015), however the way of evaluating is different. In this thesis, randomly selected patches are used, rather than taking the average performance of the central patch and the four corner patches, to evaluate performance.

Although this difference in approach may have affected the results of the experiments slightly, it can be said that the outcomes of the task of automatically recognizing the artist of an artwork show that a current convolutional neural network performs better than a less modern network architecture (86.8% accuracy when using the Inception V3 Network from 2015, 78.3% accuracy when using PigeoNET, a network based on AlexNET from 2012). Concluding, using a current convolutional neural network produces better results than an older network, which means that using a current network architecture produces a profitable way for future automated evaluation of works of art.

References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Hasan, M., Van Esesn, B. C., . . . Asari, V. K. (2018). The History Began from AlexNET: A Comprehensive Survey on Deep Learning Approaches. *Computer Vision and Pattern Recognition.* Retrieved from https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf

Barni, M., Pelagotti, A., & Piva, A. (2005). Image Processing for the Analysis and Conservation of Paintings: Opportunities and Challenges. *IEEE Signal Processing Magazine, 22,* 141-144. doi: 10.1109/MSP.2005.1511835

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

Das, S. (2017). CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more. Retrieved from https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

Elgammal, E., Kang, Y., & Den Leeuw, M. (2017). Picasso, Matisse, or a Fake? Automated Analysis of Drawings at the Stroke Level for Attribution and Authentication. *Image and Video Processing.* Retrieved from https://arxiv.org/pdf/1711.03536.pdf

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

Hughes, J., Graham, D., & Rockmore, D. (2010). Quantification of Artistic Style trough Sparse Coding Analysis in the Drawings of Pieter Bruegel the Elder. *Proceedings of the National Academy of Sciences of the United States of America, 107,* 1279-1283. doi:10.1073/pnas.0910530107

Johnson, C. R. Jr., Hendriks, E., Berezhnoy, I. J., Brevdo, E., Hughes, S. M., Daubechies, I., . . . Wang, J. Z. (2008). Image Processing for Artist Identification: Computerized Analysis of Vincent van Gogh's Painting Brushstrokes. *IEEE Signal Processing Magazine, 25*, 37-48. doi: 10.1109/MSP.2008.923513

Keras Documentation. (n.d.). Retrieved from https://keras.io/

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

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE, 86*, 2278-2324. doi: 10.1109/5.726791

Li, J., Yao, L., Hendriks, E., & Wang, J. Z. (2011). Rhythmic Brushstrokes Distinguish van Gogh from His Contemporaries: Findings via Automated Brushstroke Extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*, 1159-1176. doi:10.1109/TPAMI.2011.203

Liang, M., & Hu, X. (2015). Recurrent Convolutional Neural Network for Object Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. doi:10.1109/CVPR.2015.7298958

Lyu, S., Rockmore, D., & Farid, H. (2004). A Digital Technique for Art Authentication. *Proceedings of the National Academy of Sciences of the United States of America, 101,* 17006-17010. doi:10.1073/pnas.0406398101

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

Pinedo, D., & Ribbens, A. (2018, May 15). Onbekend schilderij van Rembrandt ontdekt. *NRC Handelsblad.* Retrieved from https://www.nrc.nl/nieuws/2018/05/15/onbekende-rembrandt-ontdekt-a1602960

Rea, N. (2018, May 16). An art dealer claims he's discovered a previously unknown Rembrandt. Where'd he find it? At Christie's. *ArtnetNews.* Retrieved from https://news.artnet.com/art-world/undiscovered-rembrandt-hermitage-amsterdam-1286810

Rosebrock, A. (2017). *ImageNet: VGGNet, ResNet, Inception, and Xception with Keras*. Retrieved from https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/

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

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Vision and Pattern Recognition.* Retrieved from https://arxiv.org/abs/1409.1556

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

Strezoski, G., & Worring, M. (2017). OmniArt: Multi-task Deep Learning for Artistic Data Analysis. Retrieved from https://arxiv.org/abs/1708.00684v1

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going Deeper with Convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. (2015). Rethinking the Inception Architecture for Computer Vision. *Conference on Computer Vision and Pattern Recognition.* doi: 10.1109/CVPR.2016.308

Taylor, R. P., Micolich, A. P., & Jonas, D. (1999). Fractal Analysis of Pollock's Drip Paintings. *Nature, 399*, 422-423. doi:10.1038/20833

Van Noord, N. (2018). *Learning Visual Representations of Style* (Doctoral dissertation).

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*, 46-54. doi:10.1109/MSP.2015.2406955

Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *Computer Vision and Pattern Recognition.* Retrieved from https://arxiv.org/abs/1311.2901

Appendix

Appendix 1. Confusion matrix experiment 1

For experiment 1 there is no confusion table.

Appendix 2. Confusion matrix experiment 2

67	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
0	231	8	0	0	0	0	2	0	2	1	0	1	0	0	5	1	1	0	0	2	4	1	9	0	0	1	0	1	1	2	3	0	1
1	1	48	0	0	1	1	0	0	4	6	0	1	3	0	4	1	1	0	1	2	0	1	11	0	0	0	0	0	0	0	1	0	0
0	3	0	43	0	3	0	2	1	3	0	0	2	0	0	1	0	1	0	0	2	0	1	1	0	0	3	0	0	0	0	0	0	0
0	1	1	0	36	8	1	0	0	0	0	0	5	0	0	1	0	0	0	0	3	1	1	3	0	0	3	0	3	0	2	0	5	1
0	0	0	0	0	135	0	1	1	0	0	1	0	1	1	1	0	0	0	0	6	0	3	1	1	0	3	0	0	0	2	1	4	0
0	0	0	0	0	0	149	7	0	3	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	3	0	0
1	1	0	0	0	4	3	102	0	1	1	0	1	1	0	1	0	12	0	1	8	0	1	1	0	1	3	0	1	0	4	0	5	0
0	0	0	5	0	3	2	1	99	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
1	2	0	4	0	1	0	3	1	184	0	1	1	2	2	17	0	1	0	1	7	2	1	0	1	0	2	0	1	0	0	2	0	0
0	2	0	0	0	0	0	2	0	0	222	0	5	0	0	0	0	1	0	0	0	0	0	5	0	0	1	1	0	0	0	0	0	0
0	0	1	0	0	0	1	1	0	1	4	66	0	0	0	1	1	0	0	0	0	1	0	2	0	4	0	0	0	2	0	4	0	0
0	0	1	0	0	1	0	1	1	0	1	0	61	0	0	1	0	3	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	100	0	2	0	0	0	1	0	0	0	2	0	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	1	0	3	0	1	0	0	38	29	0	0	0	1	0	1	0	2	0	1	0	0	0	0	0	0	0	0
1	1	0	2	0	3	0	5	1	9	1	1	1	1	12	291	1	0	0	3	3	0	1	6	0	3	2	1	0	1	2	2	1	0
0	1	0	0	0	1	0	0	0	0	0	5	0	0	0	2	57	0	0	1	1	3	0	3	0	1	0	2	1	3	1	0	0	0
0	0	1	0	0	3	0	6	0	0	0	0	1	0	1	0	0	100	0	1	4	0	1	2	0	0	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	0	49	2	0	0	1	0	0	3	0	3	0	1	0	0	0
1	0	0	1	0	6	4	9	0	0	1	0	3	0	0	0	0	6	0	0	121	0	2	1	0	0	6	0	1	0	3	0	1	0
1	3	2	1	0	0	0	0	0	1	4	0	1	2	0	6	0	0	0	0	1	49	0	4	0	0	1	0	0	0	1	2	0	0
0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	4	0	2	0	1	3	0	104	0	0	0	3	0	3	0	0	0	1	0
0	7	10	0	0	1	3	3	3	3	6	2	0	3	3	13	1	7	0	1	1	4	1	226	1	3	2	0	1	0	1	3	1	2
5	0	0	1	1	4	0	0	0	0	0	0	0	1	0	3	0	2	0	0	1	0	1	1	64	0	2	0	0	0	0	1	2	2
0	2	1	0	0	0	2	2	0	1	2	1	0	0	0	3	0	0	0	1	0	0	0	0	0	259	0	0	0	0	0	1	0	0
0	0	0	1	0	22	0	5	0	1	0	0	1	0	0	2	0	1	0	0	10	0	0	3	0	0	70	0	1	0	3	0	3	2
0	8	2	0	0	0	0	2	0	1	3	0	0	0	1	8	0	0	0	0	0	1	0	5	0	1	0	89	0	1	1	1	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0	0	1	0	0	1	0	0	0	0	59	0	0	0	0	0
0	3	0	0	0	0	1	0	0	1	1	8	0	0	0	3	2	0	0	0	0	0	0	6	0	5	0	0	0	65	0	0	0	0
0	0	0	1	0	1	0	2	0	0	0	0	0	1	1	3	0	0	0	0	4	0	1	2	0	0	1	0	1	0	97	0	0	0
0	0	1	3	0	1	1	2	2	6	0	3	0	1	1	5	1	0	0	1	2	0	2	4	0	1	1	0	0	0	0	160	1	0
0	1	0	0	0	10	1	3	0	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	6	0	0	0	3	0	103	1
10	0	0	0	0	0	1	6	0	0	0	0	0	0	0	2	0	1	0	0	1	0	4	4	1	3	2	0	1	0	1	1	0	78

Figure 12. Confusion matrix for all artists with at least 256 training examples of all artwork types. The rows represent the actual artists and the columns the artist estimates. This confusion matrix represents experiment 2.

Appendix 3. Confusion matrix experiment 3

67	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	3
0	219	3	0	0	1	0	1	0	5	1	2	1	3	0	8	1	0	0	0	0	9	2	11	0	3	0	1	1	1	1	0	2
2	5	42	0	1	1	1	0	0	3	6	1	0	2	0	5	0	0	0	0	0	4	1	10	0	0	2	0	1	0	0	1	0	0
0	2	0	41	0	3	1	3	1	1	0	0	1	0	0	4	0	1	0	1	1	0	0	0	0	0	2	1	0	0	0	2	0	1
0	0	0	0	49	5	0	2	0	0	1	0	2	0	0	1	0	1	0	0	2	1	2	0	0	1	2	0	4	0	2	0	0	0
0	0	0	1	3	129	0	2	1	1	0	0	0	0	0	3	0	1	0	0	0	1	1	1	1	0	6	0	0	0	4	0	7	0
0	0	0	0	0	1	144	12	0	1	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	1	1	0	0	2	1	2	0	0
0	0	0	1	3	4	2	116	0	0	0	0	1	0	0	1	1	4	0	0	4	0	0	2	0	0	2	0	0	0	5	1	4	2
0	0	0	4	0	1	1	2	97	2	0	0	0	0	0	3	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	2	0	0
1	2	0	2	0	3	0	4	1	177	0	1	1	1	2	26	0	0	0	0	0	2	3	2	0	3	3	0	0	0	2	0	0	1
0	1	1	0	0	0	0	3	0	0	216	0	4	0	0	2	0	1	0	0	0	1	1	7	0	0	2	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	1	66	0	0	0	1	1	0	0	0	0	0	0	5	0	6	0	0	0	4	1	2	0	0
0	0	0	0	2	0	0	1	0	0	2	0	61	0	0	1	0	2	0	0	0	0	0	1	0	1	1	0	2	0	0	0	0	0
1	0	0	1	0	0	0	2	0	1	0	0	0	99	0	1	0	0	0	1	0	0	0	0	1	0	0	0	2	0	0	1	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	0	33	37	0	0	0	1	0	0	0	1	0	1	0	1	0	0	3	0	0	1
0	2	0	2	1	0	1	3	1	12	0	1	1	1	4	299	0	0	0	2	2	1	1	6	0	5	1	2	2	0	0	1	2	2
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	65	0	0	1	0	1	2	4	2	4	0	0	0	1	1	0	0	0
1	0	0	0	2	3	0	10	0	0	0	0	3	0	0	0	0	89	0	0	3	0	0	2	0	0	2	0	2	0	5	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	1	0	2	5	0	2	0	46	0	0	2	2	0	2	1	0	0	0	2	0	0	0
1	0	0	1	2	7	5	5	0	0	2	0	0	1	0	0	0	4	0	0	119	0	5	4	0	1	5	0	1	0	1	1	0	1
0	7	2	0	0	1	0	2	0	0	2	1	0	0	0	6	1	0	0	0	0	48	1	5	1	0	1	0	1	0	0	0	0	0
0	0	0	1	3	1	1	2	1	0	0	0	1	0	0	1	0	0	0	1	5	0	98	1	0	0	4	0	3	0	0	0	1	1
1	7	4	0	2	1	2	2	1	0	5	0	2	3	1	11	4	3	0	1	6	2	2	239	0	3	1	1	1	0	2	1	2	2
4	0	0	0	0	3	2	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	73	0	0	0	0	1	1	1	0	1
0	0	1	0	0	0	2	1	0	0	0	1	0	0	1	4	1	0	0	0	0	0	0	2	0	261	0	0	0	0	0	1	0	0
0	0	0	0	7	12	2	7	0	0	0	0	4	0	0	1	0	0	0	0	3	0	0	1	1	0	74	0	2	0	4	0	6	1
0	4	0	0	0	0	0	1	0	1	2	0	0	0	0	6	1	0	0	1	0	1	0	4	0	1	0	102	0	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	59	0	1	0	0	0
0	1	0	1	0	0	2	1	0	0	0	8	0	0	0	6	0	0	0	0	0	0	0	2	0	11	0	0	0	62	0	1	0	0
0	0	0	0	0	2	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	3	0	0	0	104	0	1	0
0	4	1	4	1	4	1	1	0	2	0	1	1	0	0	8	1	0	0	2	1	0	1	3	0	1	1	1	0	0	0	160	0	0
0	0	0	0	1	6	1	4	0	0	0	0	0	1	0	2	0	0	0	0	2	0	0	1	1	0	7	0	0	0	2	0	105	0
8	0	0	0	0	1	2	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	4	2	3	2	0	0	0	0	0	0	88

Figure 19. Confusion matrix for all artists with at least 250 training examples of all artwork types. The rows represent the actual artists and the columns the artist estimates. This confusion matrix represents experiment 3.

Appendix 4. Confusion matrix experiment 4

66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	4	0	0	0	0	1	0	0	2
0	241	2	0	1	0	0	0	1	3	2	2	1	0	0	4	0	0	1	1	0	3	0	8	0	2	0	3	0	0	0	1	0	1
0	1	50	0	1	0	0	0	0	4	6	0	1	1	0	5	0	0	0	2	0	1	1	11	0	0	1	1	0	0	0	2	0	0
0	4	0	38	0	1	1	2	3	2	0	0	0	0	1	4	0	0	0	1	2	0	0	1	0	0	1	1	1	0	0	2	1	0
0	1	0	0	43	9	0	3	0	0	1	0	2	1	0	1	0	0	0	0	2	0	3	3	1	0	1	0	2	0	0	1	1	0
0	0	0	1	3	133	0	0	1	1	0	0	1	0	1	2	0	0	0	0	5	1	3	1	1	0	3	0	0	0	0	1	4	0
0	0	0	0	0	1	153	2	0	2	0	0	0	0	0	0	0	2	0	1	3	0	0	1	0	0	1	0	0	1	0	0	0	2
0	1	0	0	3	3	13	102	0	1	0	0	0	0	0	2	0	7	0	0	7	1	1	1	0	1	1	0	0	0	6	1	1	1
0	1	0	1	0	2	0	0	104	2	0	0	0	0	0	2	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	1	0	0	1	0	2	207	1	0	1	1	1	9	0	0	0	1	0	1	0	2	0	2	0	0	3	0	1	0	0	1
0	0	0	0	1	0	1	0	0	0	220	0	3	0	0	0	0	1	0	0	1	0	1	10	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	1	71	0	0	0	1	1	0	0	0	0	0	0	4	0	4	0	0	0	4	1	0	0	0
0	0	0	0	0	1	1	0	0	0	3	0	57	0	0	0	0	0	0	0	4	0	1	3	0	0	1	0	3	0	0	0	0	0
1	0	0	0	1	0	1	0	0	2	0	0	0	98	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	3	0	1
0	2	0	0	0	0	0	0	0	0	0	0	0	0	44	30	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0
0	3	0	2	0	0	1	0	1	9	1	0	0	0	9	301	1	0	0	2	0	1	2	7	1	4	1	3	0	2	0	4	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	1	0	0	0	1	0	9	1	2	1	3	0	2	1	1	0	0
1	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	100	0	1	7	0	0	2	3	0	5	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	2	0	0	0	50	1	0	0	2	0	1	2	1	0	0	0	2	0	0
0	1	0	0	0	1	5	0	0	0	1	0	0	1	0	1	1	6	0	0	143	0	2	1	0	0	3	0	0	0	0	0	0	0
0	7	2	0	0	0	0	0	0	1	5	0	0	0	0	4	0	0	0	2	1	51	2	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	3	0	0	0	0	3	0	112	1	1	0	1	0	0	0	0	0	0	0
0	6	3	0	1	2	2	1	0	0	3	0	0	0	1	10	2	3	0	0	0	2	1	270	0	1	0	1	0	1	0	2	0	0
2	0	0	0	0	2	1	0	0	0	0	0	0	1	0	3	0	1	0	0	0	0	0	0	0	76	1	0	0	0	0	0	0	4
0	1	1	0	0	0	1	0	0	0	0	2	0	0	1	1	0	0	0	0	0	0	0	0	1	0	266	0	0	0	0	0	1	0	0
1	0	0	0	5	10	1	3	0	0	1	0	0	0	0	1	0	1	0	0	9	1	1	1	1	0	82	0	1	0	1	1	4	0
0	5	0	0	0	0	0	1	0	1	2	0	0	0	1	4	0	0	0	0	0	0	0	3	0	3	0	103	0	0	0	1	0	0
0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	58	0	0	0	0	0
0	3	0	0	0	0	2	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	9	0	1	0	64	0	3	0	0
0	0	0	0	0	2	1	1	0	0	0	0	0	0	0	0	0	1	0	0	3	0	0	0	0	0	3	0	0	0	104	0	0	0
0	1	0	1	0	2	3	0	0	1	0	1	1	1	1	3	0	0	0	1	2	0	1	1	0	1	1	1	0	0	0	174	2	0
0	0	0	0	1	11	1	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	4	1	0	0	7	0	0	0	0	0	104	0
9	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	4	4	1	0	0	0	0	2	0	90

Figure 14. Confusion matrix for all artists with at least 256 training examples of all artwork types. The rows represent the actual artists and the columns the artist estimates. This confusion matrix represents experiment 4.

Appendix 5. Confusion matrix experiment 5

70	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2
0	258	1	0	1	0	1	0	0	1	0	0	1	1	0	5	0	0	0	0	0	2	2	1	0	1	0	1	0	0	0	1	0	0
0	6	52	0	0	0	0	0	0	6	5	1	0	1	0	3	0	0	0	2	0	2	1	8	0	0	0	0	0	0	0	1	0	0
0	2	0	46	0	1	0	2	3	4	0	0	0	0	0	4	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0	0
1	2	1	0	48	5	0	2	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	3	0	0	1	0	2	0	0	1	3	0
0	0	0	1	2	140	0	2	2	0	0	0	0	1	0	1	0	0	0	0	2	0	1	1	1	0	2	0	0	0	1	0	5	0
0	0	0	0	0	0	152	10	2	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
0	1	0	1	0	3	4	121	1	4	0	0	0	0	0	1	0	5	0	0	4	0	2	1	0	1	0	0	0	0	2	0	2	0
0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	216	0	1	1	1	0	8	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	4	0	1
0	3	1	0	2	0	1	1	0	1	218	0	1	0	0	1	0	1	0	1	1	1	0	5	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	77	0	0	1	0	0	0	0	0	0	0	0	1	0	2	0	0	0	3	0	1	0	0
0	0	0	0	2	1	0	1	1	0	1	0	58	0	0	0	0	3	0	0	1	0	0	1	0	1	2	0	2	0	0	0	0	0
0	0	0	1	1	0	0	0	0	2	0	0	0	102	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	1	0	0	0	0	44	27	0	0	0	0	0	0	1	0	1	3	0	1	0	0	0	0	0	0
0	5	0	1	0	0	1	0	0	7	1	0	0	1	7	323	2	0	0	0	0	0	0	1	0	2	1	1	0	1	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	69	0	0	2	1	1	0	1	0	1	0	1	0	2	0	0	0	1
1	0	0	0	1	0	2	11	0	0	0	0	0	0	0	0	0	98	0	0	7	0	0	1	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	2	0	0	0	0	0	0	2	5	0	0	0	52	0	0	1	0	0	0	1	0	2	0	0	1	0	0
0	0	0	1	0	2	4	7	0	1	0	0	0	0	0	0	0	4	0	0	140	0	0	0	0	0	4	0	0	0	2	0	0	1
0	10	6	2	0	0	0	0	0	1	3	0	0	0	0	2	0	0	0	0	0	54	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	1	2	0	0	1	0	0	2	1	1	0	1	3	0	106	0	0	0	0	0	0	0	1	0	0	1
0	8	2	0	1	1	1	1	0	7	4	0	0	1	2	7	0	0	0	1	3	2	0	265	0	1	0	0	0	1	1	2	0	1
2	0	0	0	1	2	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	80	1	0	0	0	0	0	1	0	0
0	2	1	0	0	0	0	1	0	0	0	3	0	0	0	3	0	0	0	0	0	0	1	0	0	263	0	0	0	0	0	1	0	0
0	0	0	2	2	4	0	10	0	0	0	0	1	2	0	0	0	1	0	0	6	0	0	4	2	0	78	0	1	0	2	3	7	0
0	5	2	0	1	0	0	3	0	0	1	0	0	1	0	4	3	0	0	0	0	1	0	1	0	0	0	101	0	1	0	0	0	0
0	2	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	1	0
0	2	0	0	0	0	1	1	0	2	1	3	0	0	2	4	2	0	0	0	0	0	1	2	1	6	0	0	0	67	0	0	0	0
0	1	0	0	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	2	0	0	0	104	0	1	1
0	3	0	1	3	0	2	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0	1	0	1	2	1	0	0	0	0	177	1	1
0	2	0	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	122	1
8	0	0	0	0	0	3	6	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2	0	92

Figure 15. Confusion matrix for all artists with at least 256 training examples of all artwork types. The rows represent the actual artists and the columns the artist estimates. This confusion matrix represents experiment 5.