Image Classification of Artworks based on their Creation Date

Ludwig Maximilian University of Munich
Faculty of the History and the Arts

Seminar: Automatische und manuelle Bildadressierung in der Kunstgeschichte
Lecturers: Dr. Hubertus Kohle and Stefanie Schneider
Summer Semester 2017

Ivan Bilan
Degree course: Computational Linguistics, M.Sc.
Semester: 8
Student number: 10975006
Fischbachauerstr. 69
81539 Munich
ivan.bilan@campus.lmu.de

Contents

1. Introduction

Recent advancements in the field of Deep Learning have drawn the attention of many researchers in the area of image recognition. Often current neural network models display very good accuracy in recognizing and classifying objects. In this research paper, we will utilize deep learning to categorize artworks based on the date of their creation.

We have access to a dataset of paintings and drawings with information on when they were created. This information is used to first cluster the data into various historical periods. The data is subsequently used to train a neural network model on the basis of Convolutional Neural Networks, a popular deep learning architecture often used for image classification.

Previously researchers already tried to classify artwork, for instance, based on art movements. Icogly et al., in 2004, used common machine learning algorithms to classify images based on whether they belonged to classicism, cubism or impressionism and achieved 90% accuracy. Similarly, we will classify paintings into classes, although now based on their creation date, and to train a model we will use modern advancements in the field of neural networks.

Chapter 2 will give a theoretical overview of various art epochs and styles, and concentrate on outlining key features of the painting styles. Chapter 3 will analyze the dataset of images that were obtained for this research, as well as give an overview of the distribution of images in the dataset based on their creation year.

Chapter 4 will introduce the reader to the high-level concepts that are required to understand how neural networks work and how they differ from previously established machine learning approaches. Chapter 5 will give an overview of the experiments conducted to solve the problem of classifying artworks by their creation date using deep learning. It will also evaluate the effectiveness of the proposed approaches.

2. Classifying Works of Art from a Historical Standpoint

Before analyzing the data and describing the deep learning approach to solving the problem of image classification by creation date, let us look at the various art epochs, styles and movements that have existed in the history of art and how they differ from each other.

Classical antiquity (8th century B.C. to 5th century A.D.) shows simplicity, refinement, symmetry, proportionality, and absence of superfluous details. Best examples of this period can be found in Greek and Roman art. The Greek ornamentation is characterized by spiral, wavy lines, as well as geometric and architectural elements. The Roman art was rather monumental. The strength of the Roman imperial power was embodied by ancient Roman artists in sculptural portraits of emperors in majestic poses.

Byzantine art (5th to 15th centuries) displays deep religiosity with elegant ornamental. The main types of paintings of this period are church mosaic and frescos, as well as icon paintings. Distinctive features of the Byzantine painting are elongation and flatness of the figures, rigorousness of the face, almond-shaped eyes, physical calmness and frequent use of gold. The most common are biblical subjects, ornaments of grapevine, animal figures and birds (Schneider, 2011).

Romanesque art (10th to 13th centuries) – stories from the Gospels and the Bible occupy a major part of paintings of this period. Depicted figures of people are flat, and body proportions are often violated. Ornaments and stylized images of fantastic creatures are also typical for this period (Schneider, 2011).

Gothic art (12th to 15th centuries) – majesty and grandeur with rich decorative detail, and mystical stories can best describe this period. The main types of paintings in the Gothic style are stained-glass windows on religious, historical, literary subjects. Additionally, the interest in ordinary people is increased; detailed reproduction of the smallest elements of plants and animals is also common. (Rudolph, 2011).

Renaissance (13th to 16th centuries) can be described by increased realism of the image (the figures become more vital, the space is more realistic). Shows the expressive plasticity of the figures and reflects the beauty of the human body. This period is also characterized by the spread of the portrait and the formation of the landscape as a separate genre (Hess & Hirschfelder, 2010).

Baroque (end of 16th to mid-18th centuries) reflects aristocracy and monumental solemnity. Paintings show the dynamism of images and use impressive effects with dynamic composition and tension. Often show a fusion of reality and illusion (Hess & Hirschfelder, 2010).

Rococo (first half of 18th century) can be described by elegance, style, and asymmetry of compositions. Often bizarre and fantastic ornamental motifs are part of Rococo. It differs from Baroque primarily by smaller and more complex forms, curved and interlaced lines. The main topics are gallant celebrations, rural motifs, and love stories. Historical, mythological and biblical motifs can also be seen in Rococo, although they mostly deal with love stories.

Classicism (17th - beginning of 19th century) is characterized by harmony, order, clarity of proportions and perfect artistic form. Features of classicism painting are holistic compositions with expressive means and a vivid system of colors. Themes of paintings range from ancient heroes and historical characters to mythological plots. Popular genres of painting during classicism: historical painting, portrait, landscape. *Image 1* depicts the "Oath of the Horatii" by Jacques-Louis David and Anne-Louis Girodet-Trioson as an example of classicism.

Image 1 *"Oath of the Horatii", Example of Classicism*

Romanticism (second half of 18th - first half of 19th century) - a shift from academic canons to lyricism. Depictions of sensuality and emotionality become more common. Climaxes and dramatic moments are often captured in paintings of romanticism. Genres of paintings range from historical canvases, battles, fantastic paintings and marina to portrait and self-portrait, as well as landscape. In the portrait, the focus was on the depiction of bright characters, whereas in the landscape - admiring the power of nature.

Impressionism (1860s – beg. of 20th century) can be described as the transfer of immediate impressions from the surrounding world. Paintings are characterized by fragmentary composition, unexpected points of view and angles, as well as sharp expressiveness of forms. *Image 2* depicts the "A Bar at the Folies-Bergère" by Édouard Manet as an example of impressionism.

Symbolism (end of 19th – beg. of 20th centuries) tries to express the incomprehensible nature of objects and phenomena that give the depicted objects mythological, mystical or esoteric meanings. The main emphasis in the pictures is the content, not the shape and color.

Image 2 *"A Bar at the Folies-Bergère", Example of Impressionism*

Modern art (end of 19th – beg. of 20th centuries) depicts plasticity, smoothness, laconic forms, underlining structural elements and creating unusual decorative effects. Modern art as a style sought to adorn the reality that surrounded man (MacLeod, 2011).

Cubism (beg. of the 20th century) shows images of reality in the form of combinations of geometric shapes (cube, ball, cylinder, cone, etc.) and deformed figures. Cubism is the desire to shred objects into their stereometric components. Figures depicted on canvas are elongated and distorted, often shown in the illusion of three-dimensional space.

Expressionism (**beg. of the 20th century**) is a reflection of the subjective representations and experiences of the author, as a rule, at the moment of extreme spiritual stress, irrationality, increased emotionality, the sharpness of perception and reflection.

Futurism (**beg. of the 20th century**) is a denial of realism, the proclamation of the ideas of urbanism and science; attempts to convey the dynamism of life and man's impressions. Shows the image of new beauty - the beauty of the modern world, which is represented by factories, railways, cars, planes and other images of technology.

Surrealism (early 20th century) conveys the rejection of the image of objective reality, the use of illusions, the paradoxical combination of forms, distorted proportions of objects and figures, the unnatural combination of objects.

Minimalism (second half of 20th century) - minimal transformation of materials used in the process of creativity; simplicity and uniformity of forms; creative self-restraint of the artist; use of simple geometric forms. Works are completely devoid of decorative details and paintings are often monochrome.

After gaining an understanding of what art epochs and styles existed in the history of art and how they can be characterized, in the next chapter, we will analyze the image examples in our dataset.

3. Data Overview

The original dataset of images obtained for this research was uncategorized and included around 100.000 pictures of various nature. Upon detailed inspection, most of these entries can be categorized under the following: photos of people and architecture, photos of different items (e.g., sculptures, furniture, plates, etc.), scans of book pages that often contain illustration, comics, posters, paintings, and drawings. This research paper mainly deals with classifying paintings and drawings. For this reason, these two categories have been manually separated from the other images.

Around 8.500 paintings have been clustered into one category. Additionally, a more fine-grained distribution of painting was achieved by separating around 1.500 images that are examples of Japanese art, thus leaving 7.000 color paintings representing a sample of western art only. This separation will reduce the complexity of the classification problem, since classifying Japanese art can be viewed as a separate task and ideally would require working on a dedicated neural network model to classify it. *Image 3* represents a subsample of the paintings dataset.

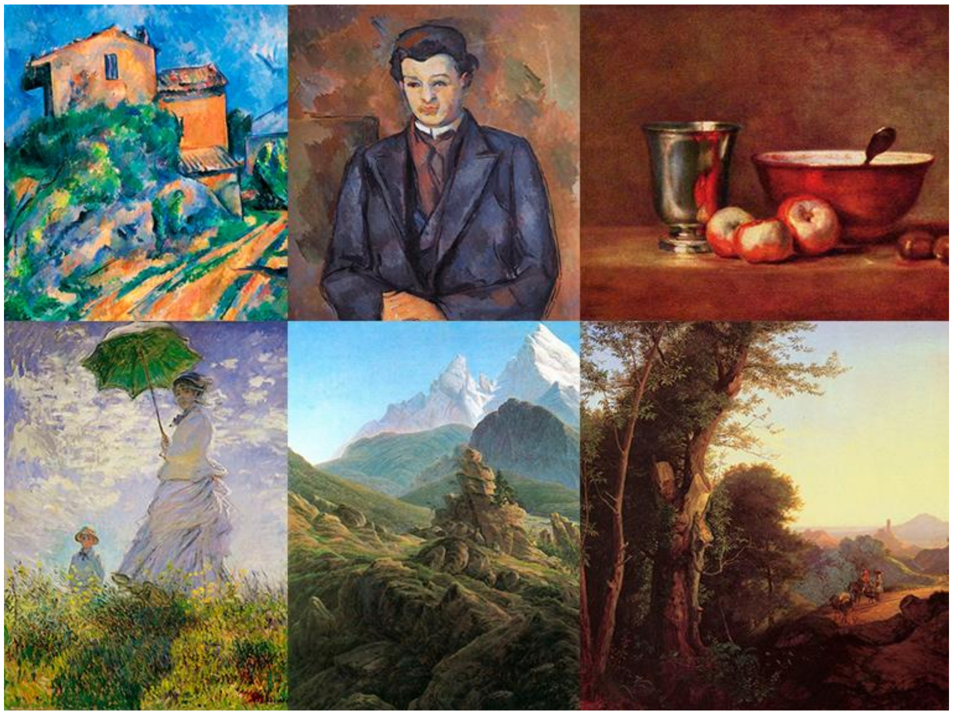

Image 3 *Paintings Dataset Examples*

Drawings are more complicated to separate from the rest of the images in the initial dataset since often there are also drawings present on scanned pages of books or the drawing only constitutes a small part of the actual image. For a clear separation, 7.700 distinct samples of drawings and grayscale paintings were selected in which the artwork itself is the only element present in the image (e.g., no parts of the book's paper, no frames or additional textual description is visible). *Image 4* provides a sample of entries that have been selected.

The dataset is also accompanied by a file containing metadata for the included images. The metadata file includes information about the title of the image, the year it was created in, name and surname of its author, the filename, as well as the location and institution in which it is currently stored. *Table 1* gives a more detailed overview of the metadata where the first row represents the column names, and subsequent rows are examples out of the metadata file. Often various columns are missing and there is no unified format for the creation date. The next subsection will deal with metadata analysis and processing.

Image 4 *Drawings Dataset Examples*

ID	Title	Date Created	Location	Institute	Artist ID	Forename	Surname	path
9548	Das von der Wahrheit erleuchtete Volk ..	1794	Paris	Bibliothèque Nationale	-	-	-	9548.jpg
2015892325	Vase mit einem Vers von Edgar Allan Poe	1898-1900	Hamburg	Museum für Kunst und Gewerbe Hamburg	4353	Emile	Gallé	m57d2eaad3e1e3.jpg
12243	Tulpen	1665/1685	Kopenhagen	Rosenborg	4140	Maria Sibylla	Merian	12243.jpg

Table 1 *Metadata Example*

3.1. Preprocessing Metadata

The provided metadata has various data points, although only some of them are of particular interest and can be used in the analysis. First, filenames are used as a unique identifier of an entry together with the image title. There are around 200.000 data entries in the metadata file. However, there are a lot of duplicate entries. Around 4.000 entries have duplicate file names and around 70.000 entries have duplicate titles. Additionally, around 3.000 images do not have any title. After removing all duplicates, approximately 125.000 entries are left.

To further analyze the data, we also look at the number of authors present in the metadata file. Overall, there are around 16.000 unique authors with around eight images on average per each author. Around 41.000 entries do not have any information about the author.

The most relevant column in the metadata file is the creation date column. It contains information about when the creation of the image started, and also in many cases when it was finished. Most of the entries in the metadata have this information, with only 8.000 entries missing the date. The main problem with this data point is that there is no unified format in the metadata file. The creation date is given in various forms, often even in a textual one. Often only the initial creation date is specified, sometimes a month and the exact day are also given. These are some notable examples of the wide array of the various ways creation date is represented in the metadata file:

- 1633-08-19 - 1633-08-19
- 1929/1932
- 1609 – 1611
- 1432
- um 1901
- ca. 1908
- Januar 1863

- 1791 - 1791-03-01
- -200 –
- early 20th century

To deal with the inconsistent formatting, a programming script was created leveraging regular expressions coupled with an existing Python library for date parsing, Arrow[1], to extract the years and store them in a unified format. Having the creation year represented in a machine-readable format allows for advanced data analysis and makes it possible to distribute the paintings and drawings into various time epochs.

3.2. Dataset Composition

Normalizing the format of the creation date makes it possible to look at the distribution of the artworks in the previously generated paintings and drawings datasets. *Figure 1* shows the distribution of the 6.500 paintings in the first dataset. Another 500 images out of the 7.000 datasets of paintings do not include any information on the creation date. Most of the samples are concentrated around the 1900s, with over 700 samples. Many images were also created between the 1800 and 1900. The third biggest group are images created around the mid-17th century. Such uneven distribution does not allow us to generate classes for our neural network model that would fit right into each art epoch or style. For this reason, we create three relatively even classes that encompass the following time periods (visually represented in *Figure 2*):

I. 1400-1759, 2305 images
II. 1760-1870, 2115 images
III. 1870-present, 1959 images

The first class, 1400 to 1759, includes artworks from the Renaissance period (1400-1600), Baroque (1600-1725), Rococo (1720-1760) and others falling into that historical period. The second class, 1760-1870, would cover, for instance, Romanticism (1800-1850) and Realism (1840-1870). The last class, 1870 until the present time, will cover a more diverse spectrum of art styles, such as Impressionism (1870-1900), Post-Impressionism, Expressionism, Surrealism and many others.

[1] http://arrow.readthedocs.io/en/latest/

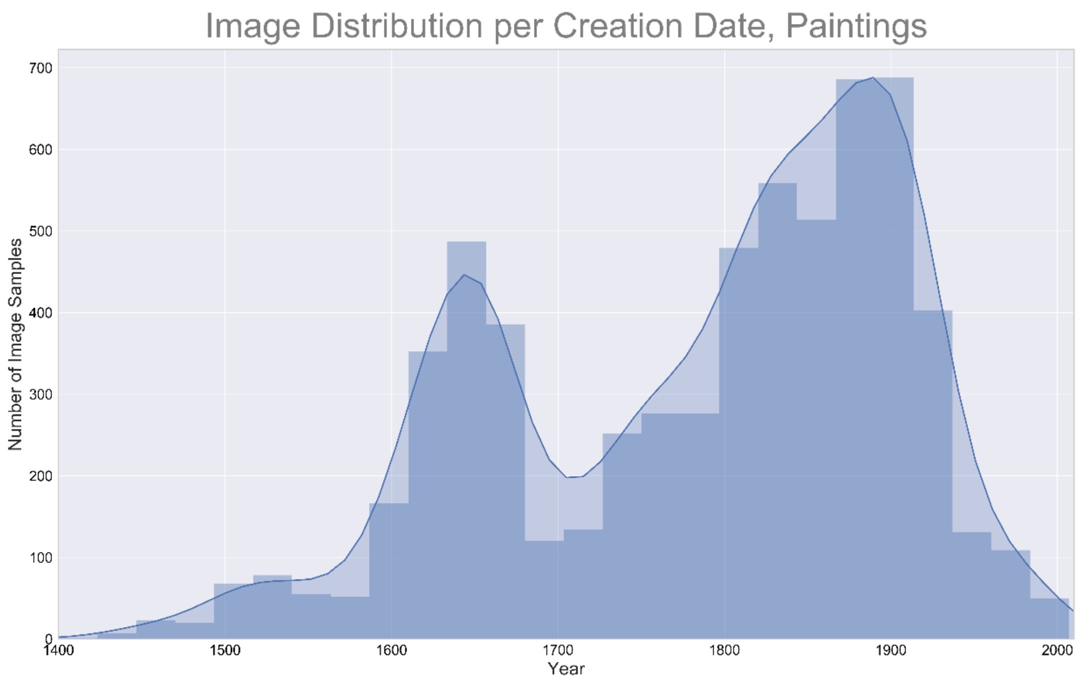

Figure 1 *Image Distribution per Creation Date, Paintings*

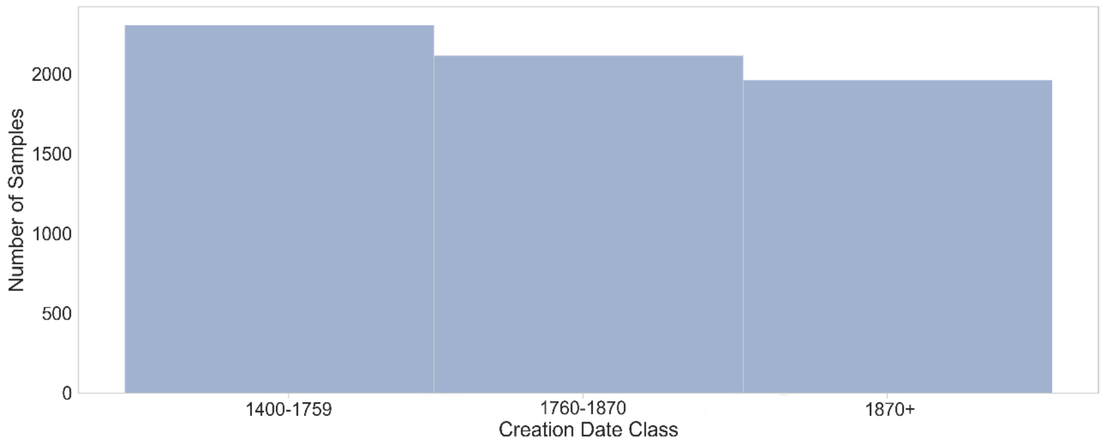

Figure 2 *Paintings Class Distribution*

For the image classification model, we will divide the dataset into two parts. One larger part is used to train a model, usually known as a *training set* in the machine-learning field. A smaller part of the dataset will be used to validate the model. Known as test set, or the validation set, it includes images that the model has not seen during training and is used to check if the trained model can predict the classes correctly. There are various ways of splitting the data into the training and the validation set. Often the data is divided into 50%

training set and 50% validation set. Although, when the dataset is small, 80% to 20%, or also 90% to 10% splits are also common. In our case, the dataset is very small, so we set around 10% of the data aside into the validation set, which is 250 images per each class.

Apart from paintings, we can now also analyze the distribution of the dataset with drawings and grayscale paintings. Out of 7.700 initially extracted images, only 7.300 of them have any information given about their creation date. *Figure 3* shows a visual representation of how the data is distributed based on the creation date of the dataset entry. The largest cluster of images lies around the year 1650, with second largest number of images created shortly before the year 1800. In contrast to the paintings dataset, this is a very different distribution. Thus, the class distribution for image classification also differs from the one we used in the paintings dataset. We will divide the drawings into five classes with a relatively equal number of images (visually represented in *Figure 4*):

I. 1400-1599, 1207 images
II. 1600-1650, 1767 images
III. 1651-1719, 1642 images
IV. 1720-1799, 1466 images
V. 1800-present, 1103 images

The validation set for the drawings also consists of 250 images per each class.

Overall, the number of available images is low in both datasets. Ideally, we would have an equal distribution per each art epoch or at least per century in order to train a stable neural network model. Although, small amounts of data is often the case with image classification problem. The data scarcity problem is present in many image as well as text recognition and processing tasks. The next chapter will give an overview of the basic theoretical background behind the neural networks, deal with Convolutional Neural Networks, transfer learning for image recognition and will serve as a theoretical basis to understanding the conducted experiments to solve the problem of image classification of paintings and drawings based on their year of creation.

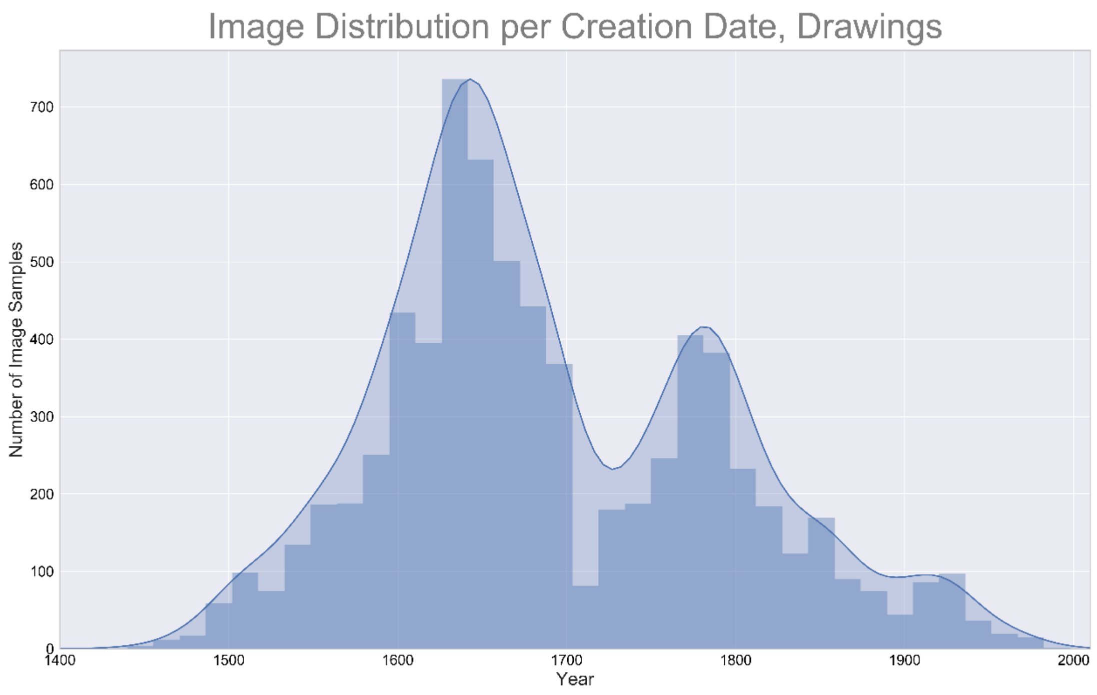

Figure 3 *Image Distribution per Creation Date, Drawings*

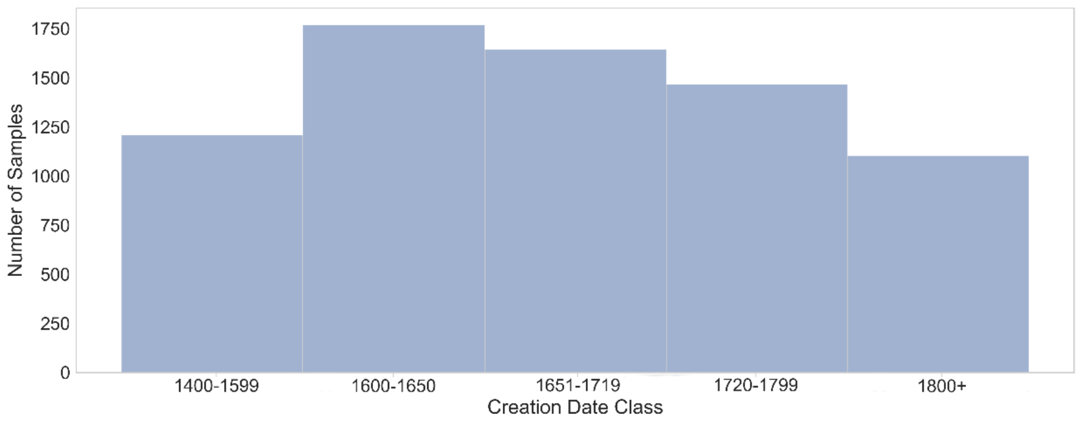

Figure 4 *Drawings Class Distribution*

4. Neural Networks for Image Classification

4.1. From Machine Learning to Deep Learning

In the last few decades most of the research pertaining to machine learning was based on the principle that feature engineering and algorithms that can interpret these extracted features, or in other words patterns in data, are the only way to achieve state of the art performance on classification and regression problems. Support Vector Machines, Decision Trees, and many other algorithms produced impressive results in text and speech processing, as well as image recognition but never actually reaching a reasonable accuracy to be used in everyday life technological solutions.

In parallel, the theoretical basis for the neural network has existed for more than 50 years, with Rosenblatt's Perceptron architecture, a rudimentary mathematical representation of a human brain neuron, first formulated in 1958. Unfortunately, while over the last few decades many researchers promised great future for neural networks they never actually produced any reasonable results. This has drastically changed within the last ten years, mainly because of the Internet which allows researchers to gather big datasets for training, as well as considerable advances in hardware, namely in the area of Graphical Processing Units. Especially after a number of neural network based models won top places in a most popular image recognition competition from Google, ImageNet 2013[2], deep learning became more popular and began to gradually outperform machine-learning based models in various tasks.

On a very high level, computational neural networks try to mimic the neural network in our brain. A neuron in the brain consists of *dendrites* that receive electrical activity, this activity is accumulated in a part of the neuron called *soma*, it is then activated by an *axon* and is sent out to the next neuron. Neurons form neural networks in which they pass these electrical stimuli between each other and process them. In deep learning, neural networks are represented in mathematical terms. An input x to a neuron has variable strength that can be measured by a numeric weight w, various inputs are then summed up to produce a so called *logit*, transformed into a new signal using a function of a form $y = f(logit)$ and sent

[2] http://www.image-net.org/challenges/LSVRC/2013/results.php

to the next neuron (Buduma & Lacascio, 2017). *Figure 5* gives a visual representation of this process (Buduma & Lacascio, 2017).

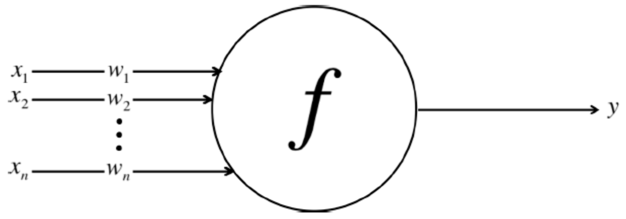

Figure 5 *Neuron in a neural network*

A more detailed mathematical overview is out of the scope for this research paper, but the main takeaway is that most of the research in the field of deep learning is initially based on the findings of how the brain of mammals works.

4.2. Deep Learning with Convolutional Neural Networks

Various computational architectures try to mimic the neural network in our brain. For instance, Recurrent Neural Networks, Long Short-term Memory (LSTM), Gated Neural Networks, Adversarial Neural Networks and more. Most of them deal with specific problem sets. For instance, LSTMs are often used to deal with sequential data, such as human language, or a sequence of images in a film. Convolutional Neural Networks (CNNs), on the other hand, are successfully used to solve various image recognition problems.

Previously, using machine-learning algorithms, the image recognition models concentrated on processing hand-coded features the researchers developed. For instance, when recognizing human faces, the model would look for areas where light intensity differs, since the area where the eyes are is usually much darker than the upper part of the cheeks. Neural networks eliminate the need for feature engineering since they can deduct their own features based on the input, an image, and the output, the object designation.

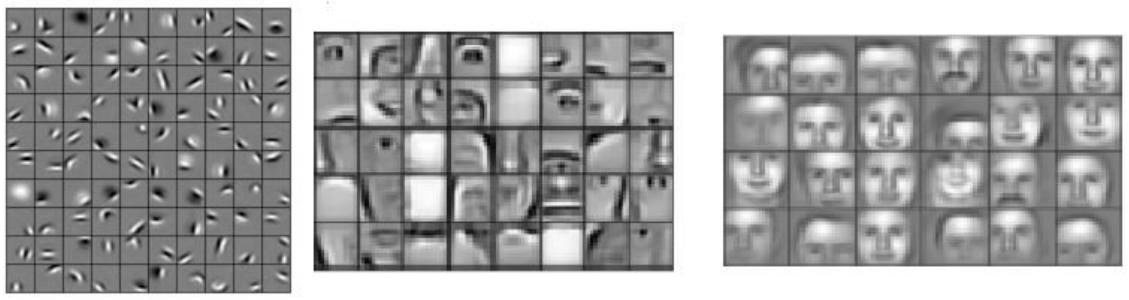

Figure 6 *Face Recognition Features in a Neural Network*

Figure 6, as illustrated by Lee (2011), shows how a neural network model automatically learns how to detect a face in a picture by first recognizing outlines of various parts of the face and then forming more high-level representations. It also gives a visual representation of how a Convolutional Neural Network automatically extracts features from an image.

CNNs can represent images in a vector form using such techniques as convolution and max pooling. It represents pixels, or more precisely, a set of pixels of the image as higher-order vector representations. It takes into account the height, width, as well as the number of channels the image has (i.e., RGB with three channels). All of these serve as the initial input which afterward is processed in a convolution layer where the higher-order representation is built. Next, max pooling layer is used to progressively reduce the dimensionality of the inputs in order to avoid overfitting. While the mathematical processes behind the CNN architecture are beyond the scope of this research paper, *Figure 7* gives an intuitive visual explanation of how CNNs would classify an image by transforming the pixels into numeric vectors (Patterson & Gibson, 2017).

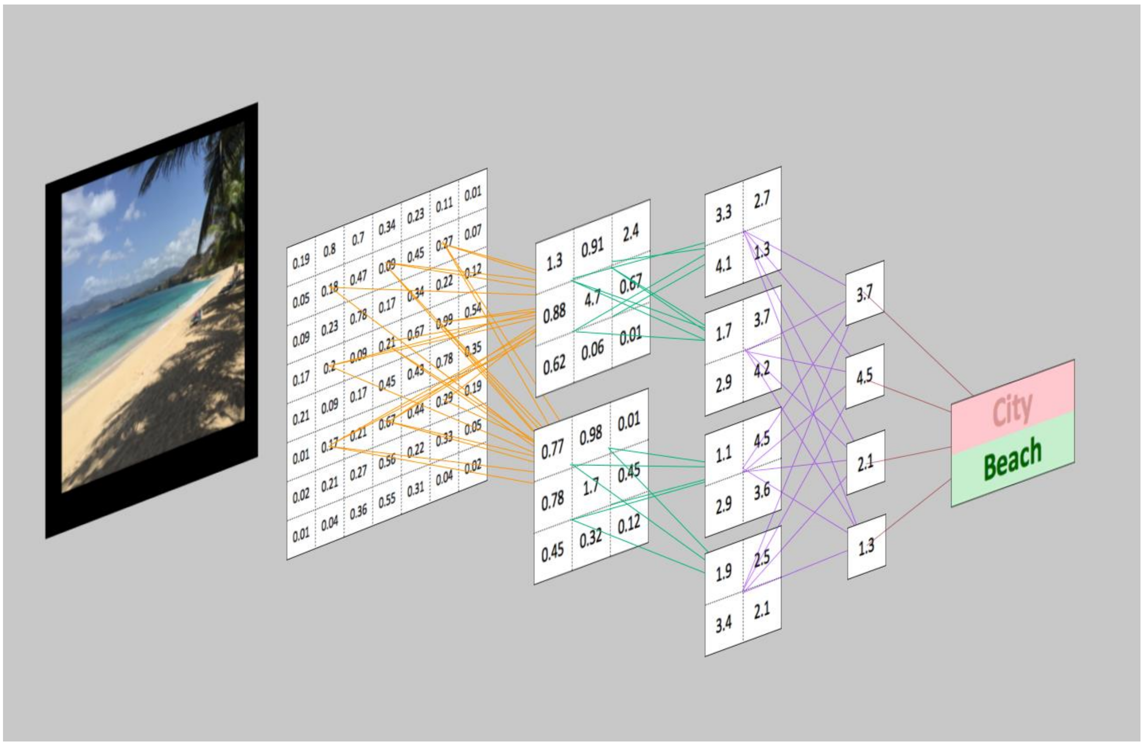

Figure 7 *Visual Representation of how CNNs Classify Images*

As previously discussed, we want to simulate how neural networks work in our brain. Just like a neuron in the brain, layers of neurons in deep learning models need to propagate the accumulated stimuli to the next layer of neurons. In mathematical terms, this can be achieved by applying a scalar to scalar function known as an activation function. There are various approaches from a mathematical perspective: linear, sigmoid, tanh and softmax functions are some of the examples. In our models, we use Rectified Linear (ReLu) activation function, which only allows the signal to propagate if the input is higher than zero. ReLu performs very well in most of the deep learning problems, but there is no fully proven theory on why it performs better than other activation functions, and thus its effectiveness remains a mystery for the researchers (Patterson & Gibson, 2017)

Neural network models often tend to learn too much from the training set, and are, in the end, not able to generalize on unseen data. When this happens, the model is said to overfit. To avoid overfitting, various regularization techniques from machine learning can be used, such as L1 or L2 regularization. Deep learning also has unique approaches to regularization, as for instance, dropout technique first formulated by Srivastava et al. 2014. When using dropout, we deactivate given percentage of neurons at random during training. This allows the model to learn even if parts of it are deactivated, so that the activated neurons can compensate for the missing ones by adapting their weights accordingly. During test time, all neurons are activated again. To sum up, this approach creates a number of smaller neural networks in which parts of the network are deactivated and combines them into one network at the end of training. *Figure 8* illustrates how the neural network changes when dropout is applied (Srivastava et al., 2014).

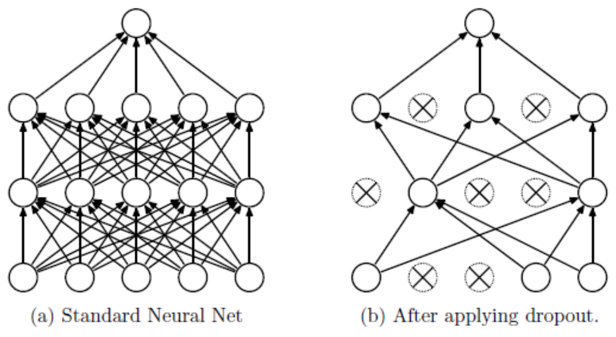

Figure 8 *Neural Network Model with Dropout*

To evaluate neural network models, the metric known as *accuracy* is used. It shows how many examples from the validation set were classified correctly. In addition to accuracy, another metric to evaluate if the neural network model is properly learning or not is used and it is known as *loss*. Loss function can show the researcher whether the model is learning from the data by observing and analyzing the errors it makes during training. The ultimate goal of the model is to find the best parameters, weights, and biases, to minimize the loss during training. The lower the loss, the closer the model is to reaching an ideal input to output mapping (Patterson & Gibson, 2017).

4.3. Transfer Learning

How accurate a neural network model is, often depends on how much data is available for training and what computational resources we have access to. The more layers in a network, the more computational power is required. The fewer images we have in our dataset, the less our model is able to generalize to unseen examples. To alleviate both of these problems, we can take use of transfer learning.

Transfer learning is a technique that allows us to partially reuse an existing complex neural network that was trained on a large dataset. The effectiveness of transfer learning is widely discussed in the deep learning scientific community, some notable papers on this topic are the ones by Yosinski et al. (2014) and Razavian et al. (2014). In short, we can take an existing network, leave the weights that it has learned during gradient descent optimization on specific layers and then remove some parts of the model and substitute them with custom layers and a classifier that can be used to solve the classification problem at hand.

There are various complex neural network implementations that are available online. The most widely used architecture for transfer learning is a VGG model published by Simonyan and Zisserman in 2014 which consists of over 20 layers of convolution. This model is impossible to train from scratch on desktop hardware. It was developed for Google's Large Scale Visual Recognition Challenge 2014[3] and won the first place. The model was trained on 14 million images and was able to classify 1.000 different classes (mostly objects, like dogs, cars, humans, etc.) with quite a high accuracy of 90%. In transfer learning, we can leverage the lower layers of the VGG model to reuse the weights it has learned when

[3] http://www.image-net.org/challenges/LSVRC/2014/

learning to detect the edges and outlines of various objects it has seen in the large 14 million images dataset.

Both the implementation of Convolutional Neural Networks and mechanisms for transfer learning are available in various deep learning frameworks.

4.4. Deep Learning Frameworks

There is a number of frameworks available that allow programmers and researchers to work with neural networks on a more higher level without dealing with much mathematical and often also complex programming side of the deep learning.

Most of the frameworks available today are implemented in the Python programming language. Some of the most used frameworks are Theano[4], Tensorflow[5], PyTorch[6] and Keras[7]. While Theano and Tensorflow are more low-level frameworks that require quite an in depth knowledge of programming as well as a thorough understanding of neural networks, Keras is a more high-level framework that can allow building neural architectures with less complexity. It follows three central framework architectural principles: modularity, each model in Keras is a sequence of graph-based models; minimalism, all operations and the interface is kept short and self-describing; extensibility, Keras can be easily extended with new functionality (Pal & Gulli, 2017). To elaborate more on the modularity principle, we first need to understand that in lower level frameworks, data flow graphs, which are sequences of computations in deep learning models, require a lot of technical understanding to build properly. In Keras, these computational sequences are often represented in a few short lines of code. *Figure* 9[5] shows a visual representation of a data flow graph in Tensorflow.

[4] http://deeplearning.net/software/theano
[5] https://www.tensorflow.org/
[6] http://pytorch.org
[7] https://keras.io/

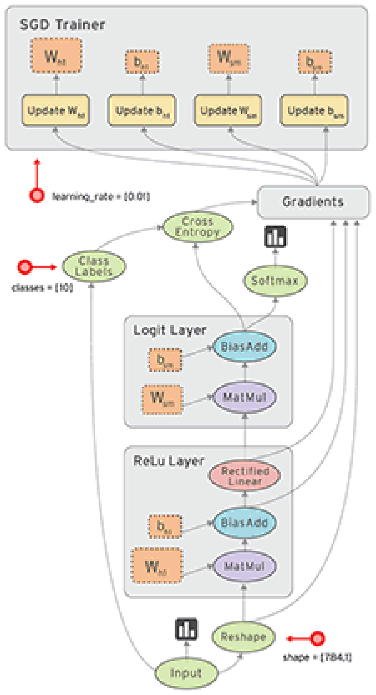

Figure 9 *Data Flow Graph Example*

In the following chapter, the neural models created to solve the artworks image classification problem will be discussed. These models are implemented using Keras framework. Moreover, all source code of the experiments is available as an attachment to this research paper.

5. Implementation and Evaluation

To classify the paintings and drawings into the previously defined classes, we will use Keras as the main deep learning framework and run the training process on a desktop computer using a Graphical Processing Unit (GPU) with 6 gigabyte of processing memory (VRAM), namely on an NVIDIA 980ti GPU. Until recently, training a complex neural network architecture on a desktop computer was impossible, and only recent advancements in hardware allow this to happen. Although, neural network architectures become more and more complex and currently the best solution is still to run the computation in the cloud or on a supercomputer with a cluster of GPUs. Usually, very deep neural networks would run for weeks before they converge. This chapter will show how we can still train and achieve a reasonable accuracy when solving the problem of classifying artwork by creation date using just a desktop computer.

5.1. Custom CNN Architecture

At first, we try to solve the classification problem using a custom CNN architecture, which we have come up with during numerous iterations. After trying out several models, the best custom model architecture can achieve 70% accuracy. It consists of three convolutional layers with 16, 64 and 64 neurons respectively and a max-polling layer between each of them with the 2x2-filter size. The last dense layer consists of 64 neurons and is regularized with *0.01* L2 regularization and a 50% dropout rate. All layers are activated with RELU activation function and the images are resized to 150x150 pixels.

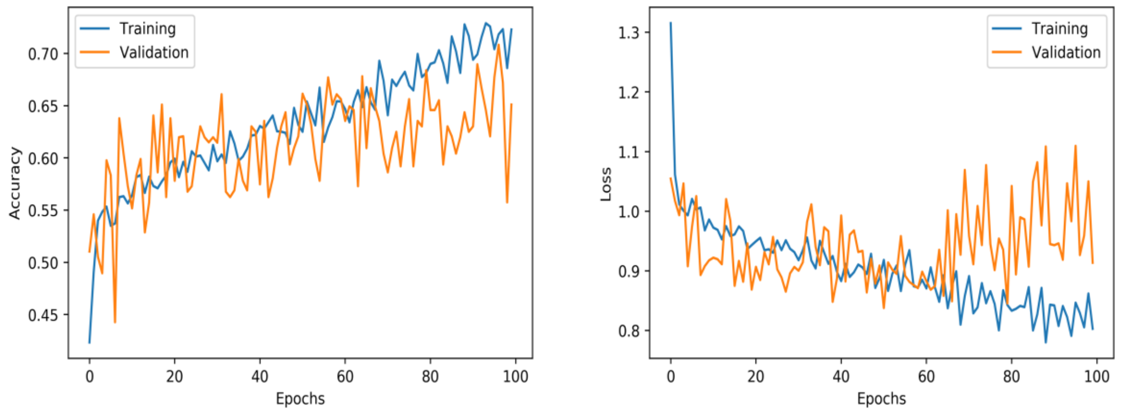

Figure 10 *Classifying Paintings, Accuracy and Loss Graph*

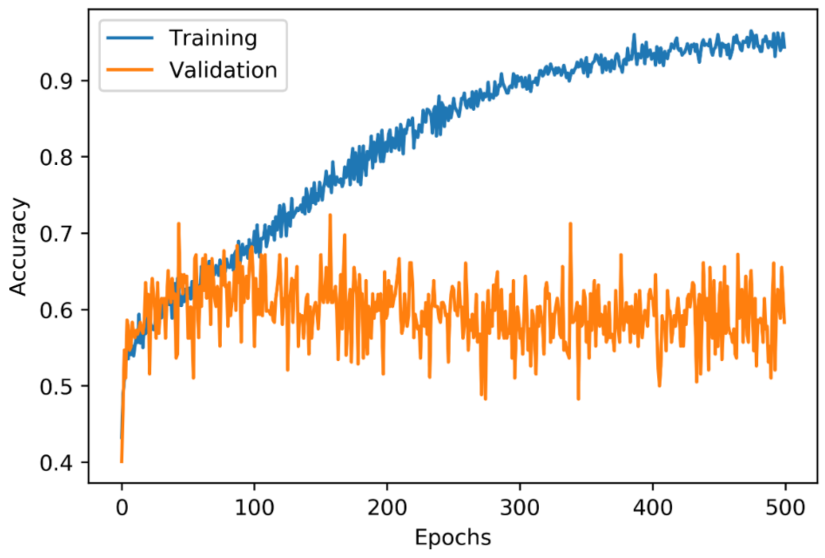

Figure 11 *Classifying Paintings with 500 Training Epochs*

Figure 10 illustrates how the accuracy and the loss are updated during 100 epochs of training. Overall, various models were overfitting and this final iteration remains relatively stable with a slightly higher accuracy on the training set than on the validation set. At the same time, the validation loss does not diverge too much from the training loss.

If the model trains for more than 100 epochs, the accuracy on the training set gradually reaches 100% but remains stagnant at around 70% on the validation set, as illustrated in *Figure 11*. Applying regularization that is more aggressive prevents the model from learning and produces worse results.

Since the training dataset is rather small, we have moved another 100 images per class from the validation set to the training set to see if the accuracy improves if the amount of data increases. Overall, the accuracy remains around 70%, but at one point, it actually reaches 100% accuracy as seen in *Figure 12*. This, however, could be a random outcome since the model does not converge afterward.

The performance on the drawings dataset, with a similar but a slightly tuned CNN architecture is considerably lower. Here we only reach 55% accuracy. *Figure 13* shows how accuracy and loss change during training.

Next, we will look at how we can improve the accuracy using transfer learning.

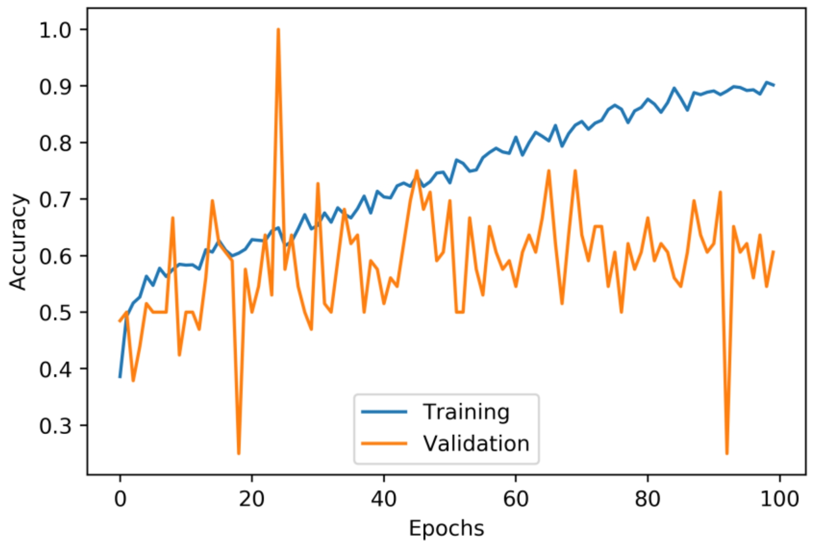

Figure 12 *Smaller Validation Set Reaches 100% Accuracy*

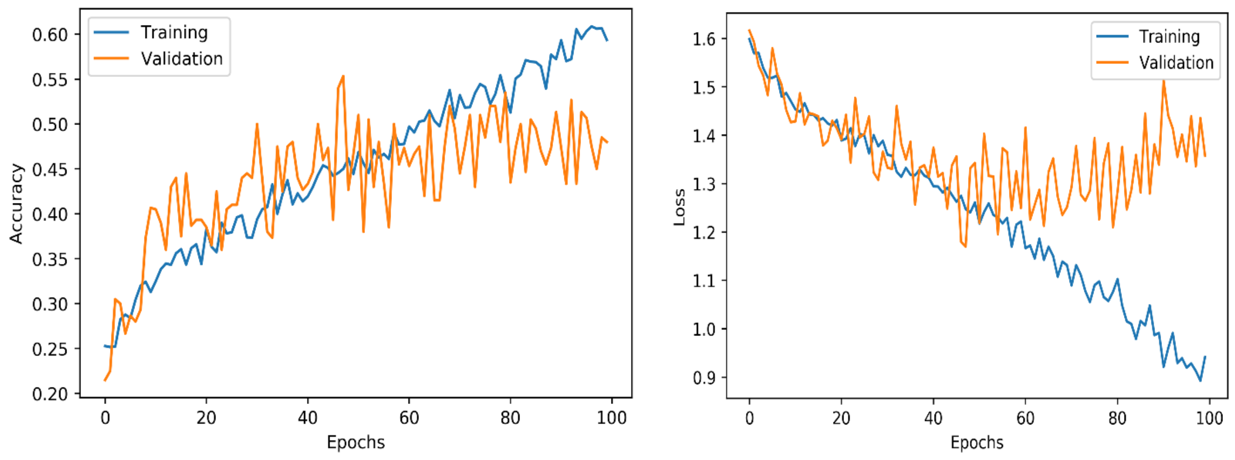

Figure 13 *Classifying Drawings, Accuracy and Loss Graph*

5.2. Augmenting Transfer Learning Models

In order to improve the accuracy, instead of continuing the experiments with the custom CNN architectures we use an existing VGG19 model (Simonyan & Zisserman, 2014) and adapt it to our classification problem. Unfortunately, it is not possible to run the VGG architecture itself and let it learn all the hyperparameters from our dataset due to the lack of computational power on the desktop PC we use. We can, however, reuse most of the weights the model learned on the ImageNet dataset of 14 million images and augment some of the layers to learn from our dataset only.

We leave the weights of the VGG19 from the first third of the architecture and let the rest of the architecture learn from the dataset we have. We also augment the VGG architecture with two dense layers of 1024 neurons each in the bottom part of the architecture and train a Stochastic Gradient Descent classifier separately for paintings and drawings datasets.

Using transfer learning, we can improve the accuracy on the paintings dataset from 70% to 75.7% (see *Figure 14*) and on the drawings dataset, from 55% to 62% (see *Figure 15* for detailed loss and accuracy visualization).

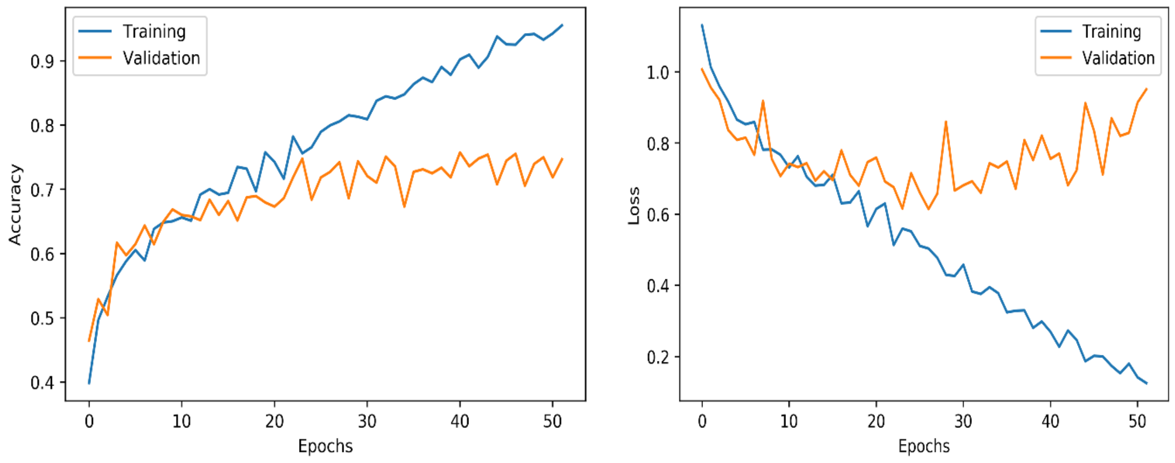

Figure 14 *Classifying Paintings with VGG19, Accuracy and Loss Graph*

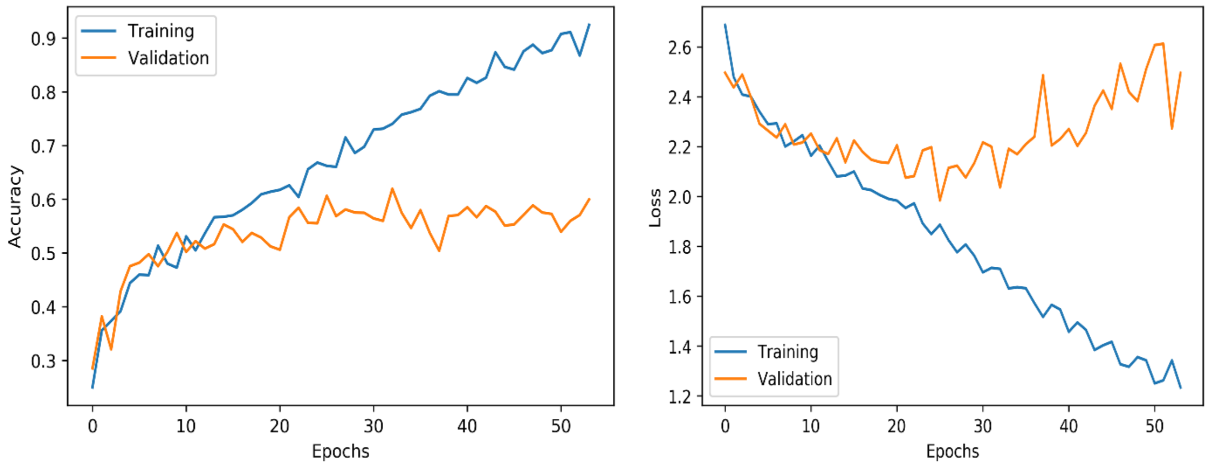

Figure 15 *Classifying Drawings with VGG19, Accuracy and Loss Graph*

6. Conclusion and Future Work

In this research paper, we looked at the problem of automatically categorizing artworks based on their creation date. We obtained a dataset of around 7.000 color paintings from various historical periods, as well as a separate dataset of 7.500 drawings and grayscale paintings.

Through the use of Convolutional Neural Networks and transfer learning from a pre-trained VGG19 model we could achieve 76% accuracy on classifying paintings into three historical periods (years 1400-1759, 1760-1870 and 1870 to present day) and 62% accuracy on classifying drawings into five historical periods (1400-1599, 1600-1650, 1651-1719, 1720-1799, 1800-present day).

Such classification is a complex problem since the dataset has a small number of image examples. Additionally, we have trained the neural network models on a desktop computer and are prevented from building more complex CNN architectures due to the limited computational power. Working on this classification problem on a more specialized deep learning hardware could potentially improve the accuracy, although, ideally, a bigger dataset could contribute to better results even more.

As future work, findings in the field of neural style transference, first presented by Gatys et al. in 2015, where a neural network is able to capture the artistic style of an image and transfer it onto another image, could be used to build a more advanced CNN architecture, which could also capture style of the image. Although, the theory and computational implementation of such approach would require extensive research and experiments.

References

Buduma, N., & Lacascio, N. (2017). *Fundamentals of Deep Learning.* O'Reilly Media, Inc.

Chollet, F. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Gatys, L., Ecker, A., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. *ArXiv e-print .*

Hess, D., & Hirschfelder, D. (2010). *Barock. Renaissance. Aufklärung. Kunst und Kultur vom 16. bis zum 18. Jahrhundert.* Germanisches Nationalmuseum Abt. Verlag.

Hodge, S. (2014). *50 Schlüsselideen Kunst.* Springer-Verlag.

Icoglu, O., Gunsel, B., & Sariel, S. (2004). Classification and Indexing of Paintings Based on Art Movements. *Signal Processing Conference,.* Vienna: IEEE.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. (2011). Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks. *Communications of the ACM, 54*, pp. 95-103.

MacLeod, G. (2011). Modernism and the Visual Arts. *The Cambridge Companion to Modernism*, 245-267.

Pal, S., & Gulli, A. (2017). *Deep Learning with Keras.* Packt Publishing.

Patterson, J., & Gibson, A. (2017). *Deep Learning.* O'Reilly Media, Inc.

Razavian, A., Azizpour, H., Josephine, S., & Carlsson, S. (2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition. *ArXiv e-prints.*

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, p. 386.

Rudolph, C. (2011). *A Companion to Medieval Art: Romanesque and Gothic in Northern Europe (Vol. 12).* John Wiley & Sons.

Schneider, N. (2011). *Geschichte der Kunsttheorie: Von der Antike bis zum 18. Jahrhundert.* UTB GmbH.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*, 1929-1958.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deepneural networks? *In Advances in Neural Information Processing Systems 27 (NIPS '14)*. NIPS Foundation.