Image Classification of Artworks based on their Creation Date

Slot F+TF || Group number: 7

Guided By

Prof. Santanu Mandal

Group Members:

1. Abhay Chaudhary	19BCE7290
2. Anish Kumar	19BCE7272
3. Sarthak Bansal	19BCE7321
4. Sankalp Gupta	19BCE7596
5. Vaibhav Sharma	19BCE7619

Section and slot: F-MAT1003-TH-ONL004

Index

S.No.	Topic	Page No.
1	Abstract	4
2	Introduction	5
3	Classifying Works of Art from a Historical Standpoint	5-8
4	Data Overview	9-10
5	Pre-processing Metadata	11
6	Dataset Composition	11-14
7	Neural Networks for Image Classification	14-15
8	Deep Learning with Convolutional Neural Networks	15-17
9	Transfer Learning	17
10	Deep Learning Frameworks	18
11	Implementation and Evaluation	19-21
12	Custom CNN Architecture	22-24
13	Augmenting Transfer Learning Models Conclusion and Future Work	24-25
14	Methodology	26
15	Expected Result	26
16	References	27

List of Images

S.No.	Topic	Page No.
1	“Oath of the Horatii”, Example of Classicism	6
2	“A Bar at the Folies-Bergère”, Example of Impressionism	7
3	Paintings Dataset Examples	9
4	Drawings Dataset Examples	10

List of Figures

S.No.	Topic	Page No.
1	Image Distribution per Creation Date, Paintings	12
2	Paintings Class Distribution	12
3	Image Distribution per Creation Date, Drawings	13
4	Drawings Class Distribution	14
5	Neuron in a neural network	15
6	Face Recognition Features in a Neural Network	15
7	Visual Representation of how CNNs Classify Images	16
8	Neural Network Model with Dropout	17
9	Data Flow Graph Example	18
10	Classifying Paintings, Accuracy and Loss Graph	22
11	Classifying Paintings with 500 Training Epochs	22
12	Smaller Validation Set Reaches 100% Accuracy	23
13	Classifying Drawings, Accuracy and Loss Graph	24
14	Classifying Drawings with VGG19, Accuracy and Loss Graph	24
15	Classifying Drawings with VGG19, Accuracy and Loss Graph	25

Abstract

In the past couple of decades, millions of photographs have been digitised. It has offered art critics and historians new possibilities. The lack of metadata and the semantic details used to identify these images, however, makes it challenging to find and access these art pictures. Finding similar artwork is a challenging job unless one recognises the exact title and creator. This problem is tackled by work in this project through the creation of unattended computer-vision methods that automatically construct metadata from art. More than 300,000 art photographs from our multimedia archive come from three sources: the Metropolitan Museum of Art, WikiArt and media. To achieve sustainability, we plan to create an immersive framework to discover the functionality extracted and to establish a suggestion program for art historians, academics and art lovers.

Introduction

Recent success has been observed by many scholars in the area of picture detection in the field of deep learning. Current neural network models also demonstrate very high performance in target detection and classification. We will use deep learning in this Research Paper to identify artworks on the date of formation.

We have access, when created, for a dataset of paintings and drawings. This knowledge is used first to divide the details into various phases of history. The data is formerly used to train an image recognition model for the neural network based on Convolutional Neural Networks, a typical deep research architecture.

Researchers have historically tried, for example, to identify art dependent on social trends. Based on how traditional, cubist or impressionist pictures are used, Icogly et al. used regular machinery learning algorithms in 2004 to identify them with 90% accuracy. They would still divide paintings into groups, but we can now use recent advances in neural networks to construct a layout dependent on their development date.

Chapter 2 provides a comparative perspective of various art ages and techniques and reflects on the main aspects of the forms of painting. Chapter 3 analyses the data collection of photos obtained for this analysis and offers a description of the distribution of images in the dataset depending on their year of existence.

Chapter 4 provides the reader with the essential principles essential for knowing how neural networks operate and how they vary from the machine learning methods previously embraced. Chapter 5 describes the studies that have been carried out to address the issue by utilising profound research in the classification of artworks. The feasibility of the suggested solutions would also be measured.

Classifying Works of Art from a Historical Standpoint

Let us yield a look at the different art epochs, trends and revolutions of art history and how they differ from each added before examining the data and explaining a deep-seated learning method to solving the question of image classification per year.

Classical antiquity (8th century B.C. to 5th century A.D.)

It displays usability, complexity, order, proportionality and lack of superfluous detail. Greek and Roman art are better examples of this era. Spiral, wavy lines, as well as geometric and architectural elements, characterise the Greek ornamentation. Roman art has become somewhat historic. In sculptural portraits of emperors with heroic poses, ancient Roman artists reflected the power of the Roman empire.

Byzantine art (5th to 15th centuries)

Plays deep religiosity of exquisite decoration. Church mosaics and frescos, as well as famous paintings, are the key styles of paintings of this era. The Byzantine painting’s defining features are elongation and flattening, the rigour of the nose, almond-shaped vision, emotional composure, and repeated gold usage. Biblical topics, grapevine ornaments, animal figures and birds are the most popular (Schneider, 2011).

Romanesque art (10th to 13th centuries)

Gospel tales and the Bible also dominated much of this period's art. The characters portrayed are smooth, and the proportion of the body is always damaged. Often typical for this era are ornaments and stylised pictures of incredible monsters (Schneider, 2011).

Gothic art (12th to 15th centuries)

Magnificence and beauty of vibrant décor with magical stories that can better characterise this time. The primary styles of Gothic paintings are stained glass windows on theological, historical and literary topics. Throughout fact, curiosity throughout average citizens is increased; the smallest plant and animal components are more widely replicated with information. (Former, 2011).

Renaissance (13th to 16th centuries)

More picture realism (figures are more vitally important; the room becomes more realistic) can be represented. Shows the figures' imaginative plasticity and represents the human body's elegance. The proliferation of the image and the emergence of the countryside as a separate genre often define this era (Hess & Hirschfelder, 2010).

Baroque (end of 16th to mid-18th centuries)

This represents historic solemnity and nobility. Maley shows the image dynamism and uses dynamic composition and tension with impressive effects. A combination of reality and delusion also exists (Hess & Hirschfelder, 2010).

Rococo (first half of 18th century)

Elegance, elegance, and composition asymmetry can be represented. Unusual and beautiful ornamental motifs also surround rococo. This varies mostly from smaller, more intricate forms, bent and interlaced lines from Baroque. Profitable markets, agricultural motifs and love tales are main topics of concern. Rococo is often used with geographical, mythological and biblical motifs, but they deal primarily with love tales.

Classicism (17th - the beginning of 19th century)

The unity, order, relative consistency and complete creative form define it. Classicism art is defined by cohesive designs of descriptive means and a bright colour palette. Painting styles range from historical figures and villains to mythological plots, in classicism common styles of art: historical art, portrait painting, the landscape. Figure 1 demonstrates the "Oath of the Horatii" as an example of classicism, by Jacques-Louis David and Anne-Louis Girodet-Treason.

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

Romanticism (second half of 18th - first half of 19th century)

A transition to lyricism from formal canons. Sensuality and emotionality are becoming more common. The romantic portraits are often depicted climaxes and tense moments. Landscapes of all styles feature traditional canvases, battle, beautiful landscapes and the marina, people, self-portraits and scenery. In the picture, the emphasis was on presenting vibrant people, while in the countryside, the beauty of nature was appreciated.

Impressionism (the 1860s – beg. of 20th century)

The communication from the external environment can be represented as immediate experiences. Paintings are distinguished by a fragmentary framework, unusual perspectives, angles and intense expressiveness of shapes. Picture 2 demonstrates an example of impressionism as “A Bar in the Folies-Bergère” by Édouard Manet.

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 colour.

Image 2 “A Bar at the Folies-Bergère”, Example of Impressionism

Modern art (end of 19th – beg. of 20th centuries)

Represents plasticity, smoothness, short forms, shows structural features and produces remarkable cosmetic results. Modern art as a form sought to decorate the nature of the human being (MacLeod, 2011).

Cubism (beg. of the 20th century)

Shows real-life pictures, such as variations of geometric forms and deformed figures (cube, ring, circle, cone, etc.). The ability to break artefacts into their stereometric parts is Cubism. The characters portrayed on paper, sometimes in a three-dimensional space distortion, are elongated and blurred.

Expressionism (beg. of the 20th century)

This represents the author’s descriptive perceptions and observations, typically in the moment of intense metaphysical tension, irrationality, heightened feelings, the sharpness of consciousness and thought.

Futurism (beg. of the 20th century)

It an effort to express the dynamism of creation and the emotions of the human person, to reject reality, to declare urbanism and research concepts. The uniqueness of the industrial world is demonstrated by warehouses, ships, automobiles and other technological photos. The picture reveals the modern elegance of this planet.

Surrealism (early 20th century)

This refers to the denial of the objective truth view, the use of delusions, the paradoxical mixture of shapes, representations of artefacts and figures, and a strange object mix.

Minimalism (second half of 20th century)

Limited material alteration used in creativity; flexibility and continuity of shapes; the artist’s artistic self-content; the usage of exact geometric types. Job without decorative details, most mostly monochrome paintings.

In this next chapter, we should examine the image examples in our dataset, providing an appreciation of which art periods and types occurred in art history and how they can be described.

Data Overview

The initial image dataset collected for this work has not been labelled and includes around 100,000 photographs of different forms. Upload the following: portraits of the persons and art, images from various objects (i.e. paintings, furniture, cards, etc.) scans of book pages that often include sketches, comics, flyers, videos, and drawings after careful examination. This article reflects on the naming of paintings and sketches. Then, the two groups were extracted manually from the other images.

Approximately 8,500 paintings were placed under one category. However, by splitting around 1500 pictures, which are representations of Japanese art, a more precisely dispersed distribution of the paintings was obtained and 7,000 colour paintings are depicted as a study of West Art only. Such distinction would reduce the difficulty of the classification issue since the classification of Japanese art should be deemed a different activity, and a dedicated neural network model will preferably be categorised. Figure 3 is a sub-sample of the dataset of paintings.

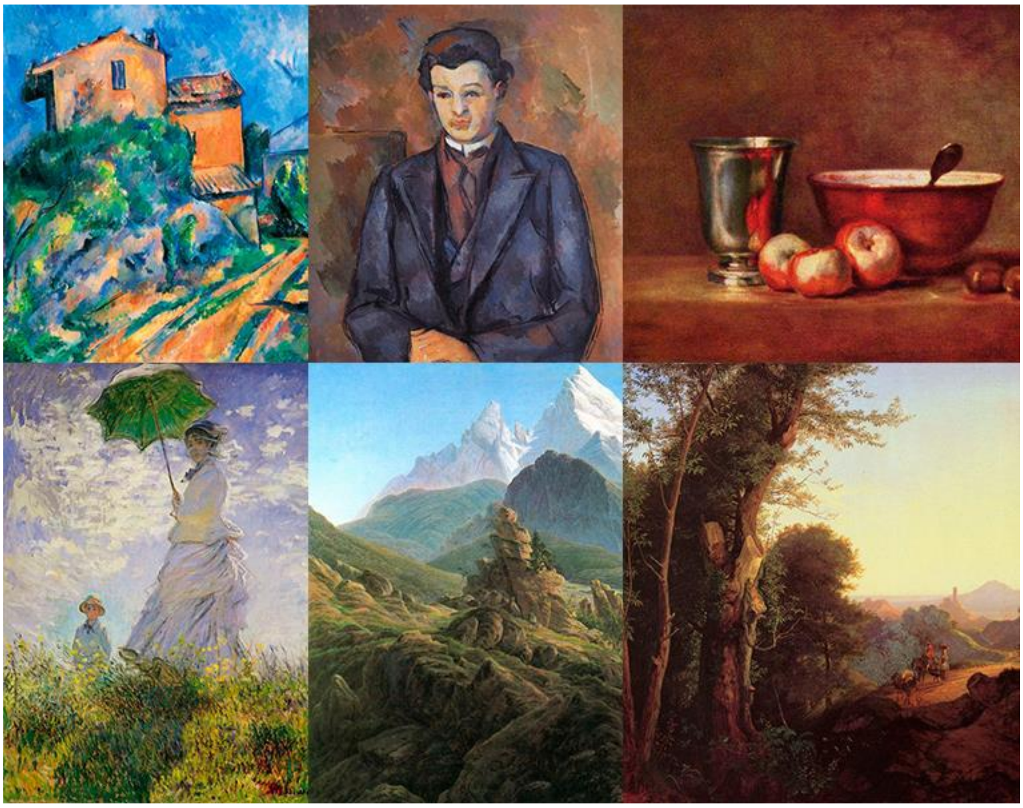

Image 3: Paintings Dataset Examples

Drawing is more difficult to distinguish from other images in the initial dataset because drawings can often also be seen on scanned book pages, or only a small proportion of the actual picture is made of drawing. For a clear separation of the drawings and grayscale paintings 7,700 individual samples have been selected, in which the work of art itself is the only item present in the image (e.g. no parts of the book, no frames or further textual description are visible). Photo 4 displays a curated selection of entries.

A package with metadata for the photos used is also included with the data collection. The metadata file includes details on the picture description, the year in which it was made, its author's name and surname, its file name and the place and organisation it is currently housed. Table 1 provides a more comprehensive description of the metadata in which the first row includes column name, and samples of metadata file are given in the following tables. There are often many columns absent, so the formation date is not consolidated. Metadata research and preparation are covered in the next paragraph.

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*

Pre-processing Metadata

The metadata given contains many data points, but only several of them are of particular significance and can be used for research. The first is to use filenames along with the picture title as a specific identification for an object. The metadata file includes nearly 200,000 data entries. There are also double submissions. They are others. Around 4,000 entries have double-file titles, and over 70,000 entries have double-file names. Moreover, there is no title to around 3,000 pictures. Roughly 125,000 entries are left following the elimination of all duplicates.

We also evaluate the number of writers in the metadata file to further analyse the results. In all, there are around 16,000 specific writers with an average of around eight pictures per book. There is no detail regarding the author in about 41.000 articles.

The section in the metadata file most important is the date formation tab. It includes details about when the picture was made, and also when it was finished. This detail is contained in much of the documents in the database, with just 8.000 records incomplete. The critical issue with this data point is that the metadata file does not contain a consistent format. The date of development is specified in different ways, sometimes in a textual type. Just state the original development date, often one month and the same day.

A programming script was generated to fix incompatible formats, utilising regular expressions in conjunction with an existing Arrow1 Python library to remove the years and place them in a uniform format. For the development year in a machine-readable system, detailed analyses of data are available, and paintings and illustrations may be published at various periods.

Dataset Composition

Standardising the standard for the production date allows the dissemination of works in paintings and drawings databases previously created to be used. Figure 1 shows in the first dataset the distribution of 6,500 pictures. Out of the 7 000 art data sets, there are about 500 pictures without any details about the date of production. In the 1900s, more than 700 of the samples were collected. Around 1800 and 1900, several pictures were also made. The third biggest group was pictures from the mid-17th century. Such unequal distribution does not require us to create classes that suit into any art age or design for our neural network model. Therefore, we establish three fairly even groups spanning the following spans (as seen in Figure 2):

I. 1400-1759, 2305 images

II. 1760-1870, 2115 images

III. 1870-present, 1959 images

The first batch, from 1400 to 1759, includes artwork from the time of Renaissance (1400-1600), Baroque (1600-1725), Rococo (1720-1760). For, e.g., Romantics (1800-1850) and Realism (1840-1870) were the second levels, 1760–187. The first grouping, 1870-1900, would encompass a broad spectrum of visual movements from Impressionism (1870-1900), post-Impressionism, Expressionism and Surrealism.

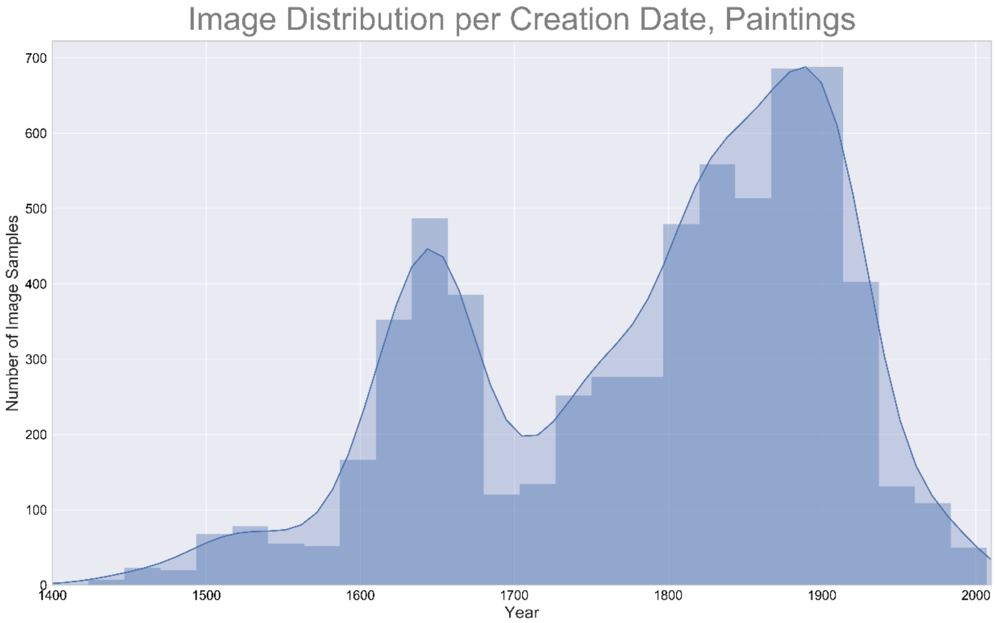

Figure 1 Image Distribution per Creation Date, Paintings

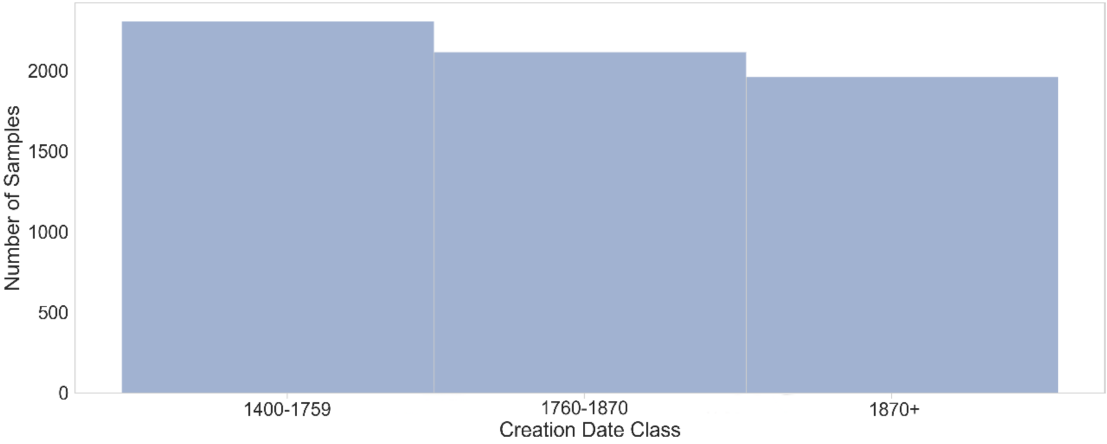

Figure 2 Paintings Class Distribution

We will division the dataset into two parts for the image classification model. One more significant part is used to train an object, generally referred to as a machine-learning training package. To test the pattern, a smaller portion of the dataset is used. Known as a test set or a reference package, it involves pictures that the model did not see at exercises and is used to verify whether the qualified model will accurately forecast courses. Data can be broken down into the training and validation set in different ways. The data were also split into a training set of 50% and a testing set of 50%. Nevertheless, 80% to 20% or even 90% to 10% of splits are often accessible where the data collection is low. In our situation, the dataset is incredibly low, so about 10%, which is 250 pictures per class, of the data, are placed in a validation package.

Besides drawings, the distribution of the dataset of white and green drawings can now be studied. Just 7,300 of them have details about their production date of 7.700 photos initially collected. Figure 3 shows the way the results are transmitted based on the data collection period. The largest picture cluster is about 1650, with the second-largest number of images being generated just before 1800. It is a somewhat distinct structure, from the paintings dataset. So the image class distribution is also different from that which we used in the data set of paintings. We shall divide the sketches into five sections with a relatively equal number of images:

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 package contains 250 photos per class, as well.

Generally, both repositories have a small number of photos available. To order to train a robust neural network model, we must preferably be evenly distributed at-art epoch or at least per cent. Nonetheless, the issue of image classification is typically limited volumes of data. Many images, text recognition and processing tasks have the issue of data scarcity. The next chapter describes the specific scientific framework behind the neural networks, deals with the Convolutional Neural Networks, pass education for the identification of images and is technically used to explain the experiments carried out to address the question of the classification of images of paintings and drawings in conjunction with their year of creation.

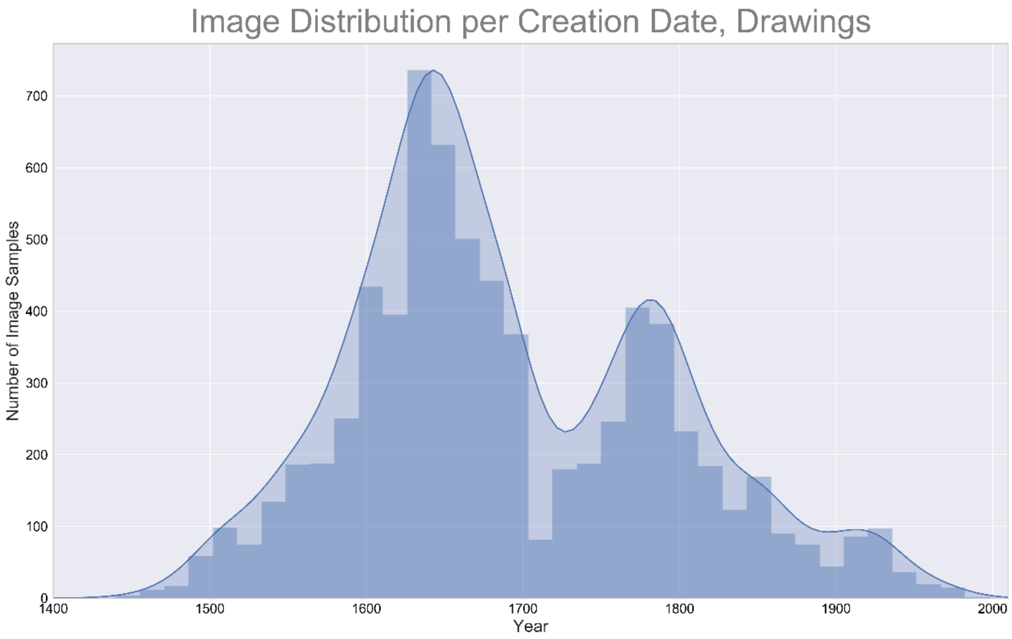

Figure 3 Image Distribution per Creation Date, Drawings

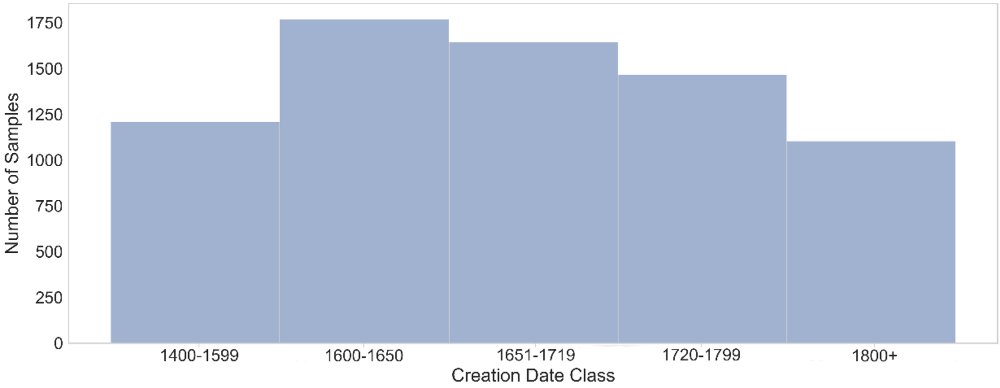

Figure 4 Drawings Class Distribution

Neural Networks for Image Classification

From Machine Learning to Deep Learning

Over the past couple of decades, most work relevant to machine learning has taken as the only way of achieving state of the art in classification and rectification problems features and algorithms, which can be used to view certain derived features, or, in other words, data trends. Aid for vector machines, decision trees, and several other algorithms have provided excellent results in text, voice, image recognition and never really an excellent precision to use in technical solutions in daily life.

Around the moment, with Rosenblatt's Perceptron model, a primitive graphical depiction of a human brain cell, first proposed in 1958, the scientific foundation for the neural network persisted for more than 50 years. Unfortunately, while many theorists have had promising hopes for neural networks in the past few decades, they have never really obtained fair outcomes. Thanks mainly to the Web, which helps researchers to collect vast data sets and to make significant advancement in technology, especially in the area of graphic processing units, that has changed dramatically over the past decade. Notably, after many neural network models in a high Google ImageNet competition won top positions, deeper learning was popularised in 2013 and machine learning-based models slowly outperformed in different occupations.

Computational neural networks on a considerable level seek to mimic our brain's neural network. The brain cell is composed of dendrites and has a strong influence. This is stored in a portion of the cell called soma and triggered by an axon. Neurons form neural networks in which they communicate and store these electric stimuli. Neural networks are mathematically described in deep learning.

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

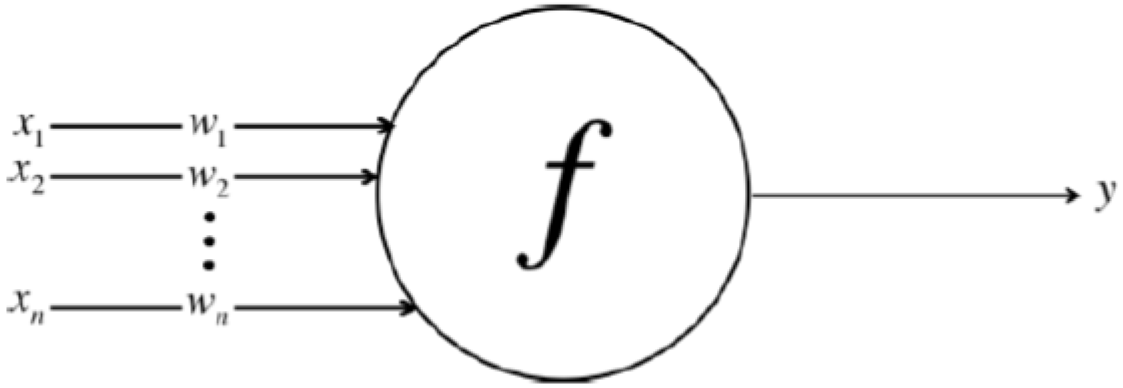

Figure 5 Neuron in a neural network

The study paper provides a more comprehensive statistical description, but the key conclusion is that most scientists in the area of deep learning are focused first on observations on the way the brain on primates functions.

Deep Learning with Convolutional Neural Networks

Various machine systems seek to mimic our mind’s neural network. For, e.g., recurrent neural networks, long short-term memory (LSTM), neural gated networks, neural adversarial nets and more. Most address specific problem sets. For sequential data such as the human language or a series of photographs in a video, for example, LSTMs are also used. CNN’s are used to address different problems with the representation of pictures. Convolutional Neural Networks (CNNs).

Throughout the past, the image recognition models focused on hand-coded processing features developed by the researchers utilising machine-learning algorithms. The algorithm will search at places where light intensity varies where, at an example, the region in which the eyes usually are significantly brighter than the upper part of the buttocks. Neural networks escape the need for practical engineering because their features can be removed by data, picture and output, entity classification.

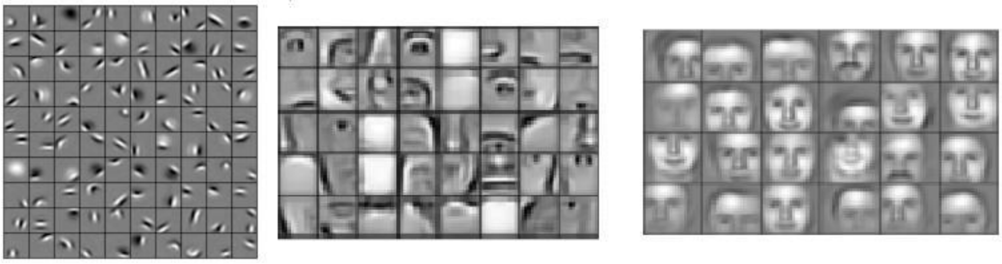

Figure 6 Face Recognition Features in a Neural Network

Figure 6, as seen in Lee’s (2011), illustrates how a neural network model will recognise a face automatically by detecting contours of different sections of the face and creating high-level photos. It also visually reveals how a Convolutional Neural Network removes information from the picture automatically.

CNN will view vector images utilising strategies like convolution and full pooling. This represents a collection of pixels in the picture as higher-order vector representations or more accurately. The height, the width and the number of channels the picture has (i.e. the RGB of three channels) are taken into account. All this serves as the first data, which is then translated into a convolution layer where the representation of the higher-order is constructed. Next, a top pooling layer is used to reduce input dimensions gradually to ensure that no excess is achieved. Figure 7 provides an intuitive visual account of how CNNs would classify the image by turning its pixels into numerical vectors (Patter son & Gibson, 2017) as the mathematical processes of the CNN architectural are beyond the scope of this research study.

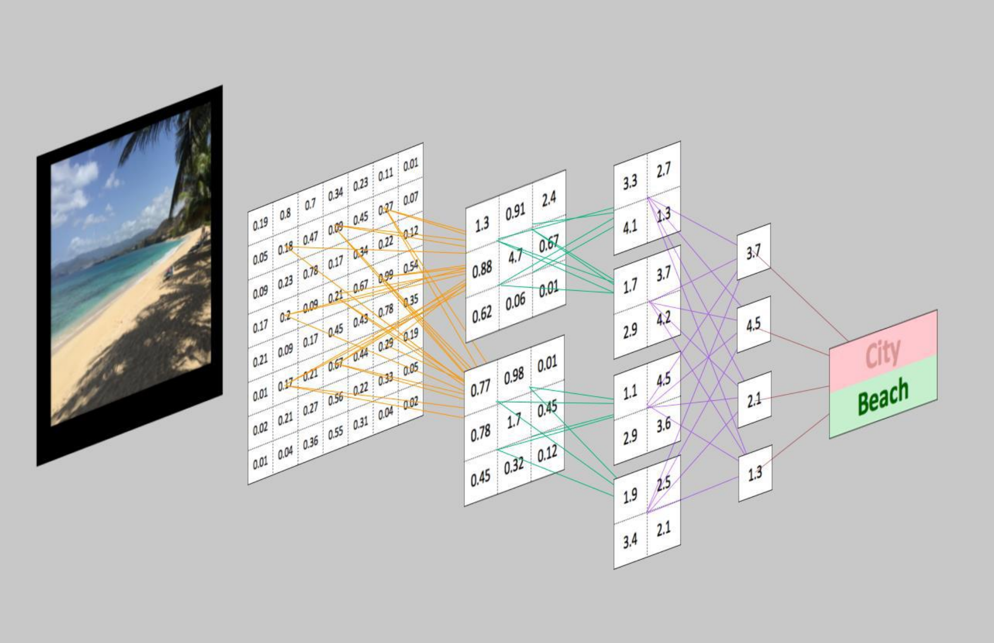

Figure 7 Visual Representation of how CNNs Classify Images

We would like to model how neural networks function in our brain, as described earlier. Much as a neuron in the brain, neuron layers in deep research models will disperse the stored inputs to the next neuron layer. This can be achieved by utilising a scalar to scalar method regarded as an activation function in mathematical terms. There are many solutions from a mathematical viewpoint: some of the definitions include cubic, sigmoid, tanh and softmax functions. Throughout our models, we use the activation function Rectified Linear (ReLu) to distribute the signal when the input is more significant than zero. ReLu is very good in most deep learning issues, but it has not been thoroughly shown that it performs better than other active functions, and its efficacy is still a mystery for scientists (Patterson & Gibson, 2017)

Neural network models frequently know very much from the training set and can not generalise on unsightly results at long last. The software is assumed to overfit as this happens. Different machine-learning regularisation techniques, such as L1 or L2 regularisation, may be used to avoid overfitting. Deep learning also has unique regularisation methods, such as the first discontinuance method proposed by Srivastava et al. 2014. We kill a certain proportion of neurons arbitrarily in training by utilising the drop-off. It helps the model to learn even though portions of it are damaged, and the active neurons will compensate by changing their weights correctly for the absent ones. Both nerves are reactivated during the study process. In brief, this method produces a variety of smaller neural networks in which network components are disabled and merged into one network at preparation. The neural network modifications as a consequence of abandonment (Srivastava et al., 2014) as seen in Figure 8.

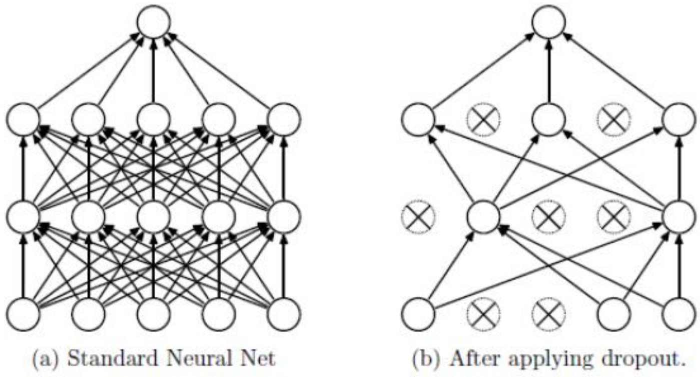

Figure 8 Neural Network Model with Dropout

The parameter defined as precision is used to test neural network models. This indicates how many validity collection cases have been correctly listed. In addition to accuracy, another parameter is used to calculate whether or not the neural network model is correctly trained, which is considered a failure. The researchers will benefit from the results by analysing and assessing the mistakes that they create during the preparation, utilising the loss function. This model's primary purpose is to determine the right conditions, weights and prejudices to reduce training loses. If losses are smaller, the similar the model is to the optimal input for mapping output (Patterson & Gibson, 2017).

Transfer Learning

How accurate a neural network model also depends on the volume of training data available and on the computing tools we have access to. The more computational power is necessary for, the more layers within a network. The fewer images we have in our data package, the less our model can spread to unseen examples. We may use transfer learning to solve all these issues.

Transfer education is a method that allows us to partially reuse a large dataset of an existing complicated neural network. Among the more in-depth learning science, the usefulness of transfer learning is commonly debated, with Yosinki et al. (2014) and Razavian et al. (2014) having some outstanding papers on this topic. Briefly, we may delete an established network and keep the weights learned from the optimisation of the gradient decrease on those layers, and then eliminate some of the components from the model, and substitute them with a specialised layer and a classifier to solve the classification problem.

Many detailed models of neural networks are accessible online. A VGG model published by Simonyan and Zisserman in 2014, consisting of more than 20 layers of convolution, is the most common translation architecture used. This model can not be conditioned on desktop hardware from the very beginning. It was developed and won first place at the 2014 Google Large Scale Visual Recognition Challenge. The model was prepared for 14 million pictures and identified 1,000 groups with a high precision of 90 per cent (mostly things, such as pets, vehicles, human beings, etc.).

We use the lower layers of the VGG model in our learning transition to recover the weight it has acquired through the analysis of the edges and contours of various items in the broad 14 million pictures.

In different deep learning systems, both implementation of Convolutional Neural Networks and transfer learning mechanisms are available.

Deep Learning Frameworks

There are a variety of architectures that enable programmers and researchers to interact with neural networks to a higher degree without grappling with any of the in-depth learning training and yet often complicated.

The bulk of the accessible implementations are actually in the vocabulary of Python programming. Theano4, Tensorflow5, PyTorch6 and Keras7 are some of the more popular frameworks. Whereas Theano and Tensorflow are more systems of lower rank, which need a detailed understanding of programming and neural networks, Keras is a higher-grade system that can allow the construction of less complex neural architectures. This is a product of three main design frameworks: modularity, with a series of graphic templates for each element in Keras; minimalism; both operations and the GUI are short-term and self-described; extensibility and new features will quickly be expanded for Keras (Pal&Gulli, 2017). To expand further on modularity, first of all, we need to realise that lower-level frames need much technological comprehension to construct data flow graphs, which are conceptual sequences in profound learning models. The sequences are often displayed in a few short lines of code in Keras. The view of data flow charts in Figure 95 is shown in Tensorflow.

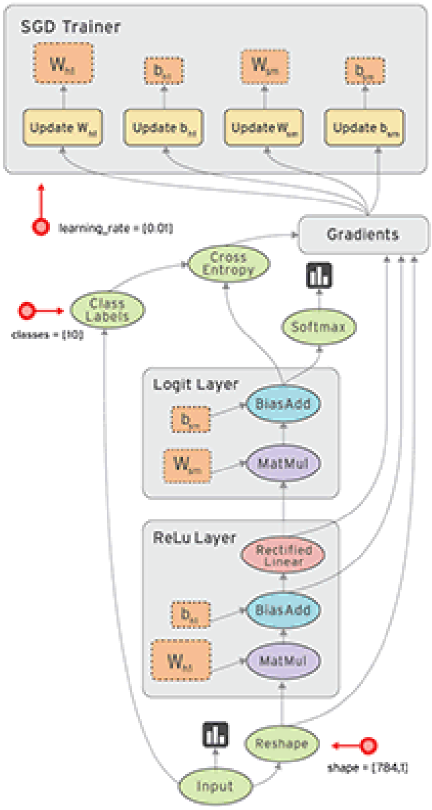

Figure 9 Data Flow Graph Example

The following chapter discusses the neural models created to solve the problem of image classification. Such frameworks are applied in the sense of Keras. In comparison, the experiments' source code is accessible in this study paper as an appendix.

Implementation and Evaluation

We're using Keras as the critical deep learning system to classify paintings and drawings into the groups described above, and are practising on a desktop computer using a 2 Gbps graphical processing device, namely an NVIDIA GPU. Before recently, a robust neural network design could not be built on a desktop machine, and this was only feasible with recent hardware development. Wherever neural network architectures become highly complicated, the optimal approach is either to operate the machine on a cluster of GPUs in the cloud or a supercomputer. Secure neural networks will last weeks before their convergence. This chapter will demonstrate how even to figure out the question of a grouping of plays by date using a personal machine to obtain fair accuracy.

Python Implementation Source Code run on Jupyter Notebook

```
from keras import applications
from keras.preprocessing.image import ImageDataGenerator
from keras import optimizers
from keras.models import Sequential, Model
from keras.layers import Dropout, Flatten, Dense, GlobalAveragePooling2D
from keras import backend as k
from keras.callbacks import ModelCheckpoint, LearningRateScheduler, TensorBoard,
EarlyStopping
# define various hyperparameters
nb_train_samples = 5622
nb_validation_samples = 750
batch_size = 16
epochs = 50
# load the pre-trained model for transfer learning
img_width, img_height = 256, 256
model = applications.VGG19(weights="imagenet", include_top=False, input_shape=(
    img_width, img_height, 3))

# to load other models follow use the following parameters:
"""
{
    "xception":    { "width": 299, "height": 299 },
    "vgg16":       { "width": 224, "height": 224 },
    "vgg19":       { "width": 224, "height": 224 },
    "resnet50":    { "width": 224, "height": 224 },
    "inceptionv3": { "width": 299, "height": 299 },
    "mobilenet":   { "width": 224, "height": 224 }
}
"""
# Freeze the layers which you don't want to train. Here I am freezing the first 5 layers.
for layer in model.layers[:5]:
    layer.trainable = False
# adding custom Layers
x = model.output
x = Flatten()(x)
```

```
x = Dense(1024, activation="relu")(x)
x = Dropout(0.5)(x)
x = Dense(1024, activation="relu")(x)
predictions = Dense(3, activation="softmax")(x)
# creating the final model
model_final = model(input = model.input, output=predictions, training = True),

# compile the model
model_final.compile(loss="categorical_crossentropy",
                optimizer = optimizers.SGD(lr=0.0001, momentum=0.9),metrics = ["accuracy"]
                )
return Add()([inputs, h])
# Initiate the train and test generators with data Augumentation
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    horizontal_flip=False)

# this is the augmentation configuration we will use for testing:
# only rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    'D:\Vellore Institue of Technology\Semester wise Material\Semester 3(Fall Semester)(2020-2021)\MAT-1003 Discrete Mathematical Structures Slot F\Classical_Paintings_Forensics-master\Dataset',
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode="categorical"
)

validation_generator = test_datagen.flow_from_directory(
    'D:\Vellore Institue of Technology\Semester wise Material\Semester 3(Fall Semester)(2020-2021)\MAT-1003 Discrete Mathematical Structures Slot F\Classical_Paintings_Forensics-master\Validation',
    target_size=(img_height, img_width),
    class_mode="categorical"
)

# Save the model according to the conditions
checkpoint = ModelCheckpoint("vgg19_1.h5", monitor='val_acc',
                      verbose=1, save_best_only=True,
                      save_weights_only=False,
                      mode='auto', period=1)
# monitor the loss
early = EarlyStopping(monitor='val_acc', min_delta=0, patience=10, verbose=1, mode='auto')
# Train the model
```

```
hist = model_final.fit_generator( train_generator, steps_per_epoch=1800 // batch_size,
epochs=100,
   validation_data=validation_generator,
   validation_steps=250 // batch_size,
   callbacks=[checkpoint, early],
   workers=8  # cpu generation is run in parallel to the gpu training
)

print("Maximum train accuracy:", max(hist.history["acc"]))
print("Maximum train accuracy on epoch:", hist.history["acc"].index(max(hist.history["acc"]))
+ 1)

print("Maximum validation accuracy:", max(hist.history["val_acc"]))
print("Maximum validation accuracy on epoch:",
hist.history["val_acc"].index(max(hist.history["val_acc"])) + 1)

"""
Epoch 00051: early stopping
Maximum train accuracy: 0.955357142857
Maximum train accuracy on epoch: 52
Maximum validation accuracy: 0.75757575835
Maximum validation accuracy on epoch: 41
"""
# visualize the results
import matplotlib.pyplot as plt


def plot_history(hist):
   plt.figure()
   plt.xlabel("Epochs")
   plt.ylabel("Loss")
   plt.plot(hist.history['loss'])
   plt.plot(hist.history['val_loss'])
   plt.legend(['Training', 'Validation'])
   plt.savefig('paintings_loss_vgg19.png', dpi=400)

   plt.figure()
   plt.xlabel("Epochs")
   plt.ylabel("Accuracy")
   plt.plot(hist.history['acc'])
   plt.plot(hist.history['val_acc'])
   plt.legend(['Training', 'Validation'])
   plt.savefig('paintings_accuracy_vgg19.png', dpi=400)


plot_history(hist)
```

Custom CNN Architecture

First of all, we seek with a customised CNN architecture to overcome the classification question which we have established over several iterations. Once many versions have been tested, 70 per cent of the right design software architecture can be obtained. It consists of three convolutionary layers, each of 16, 64 and 64 neurons, and a fixed pollinating layer with 2x2 filters between them. The last dense layer is composed of 64 neurons with a regularisation rate of 0.01 L2 and a dropout rate of 50 per cent. With the control feature RELU, all layers are disabled, and the pictures are resized to 150x150.

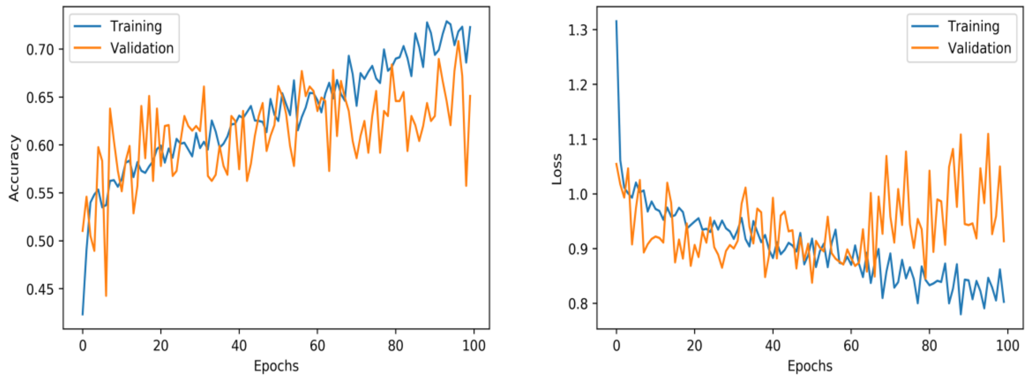

Figure 10 Classifying Paintings, Accuracy and Loss Graph

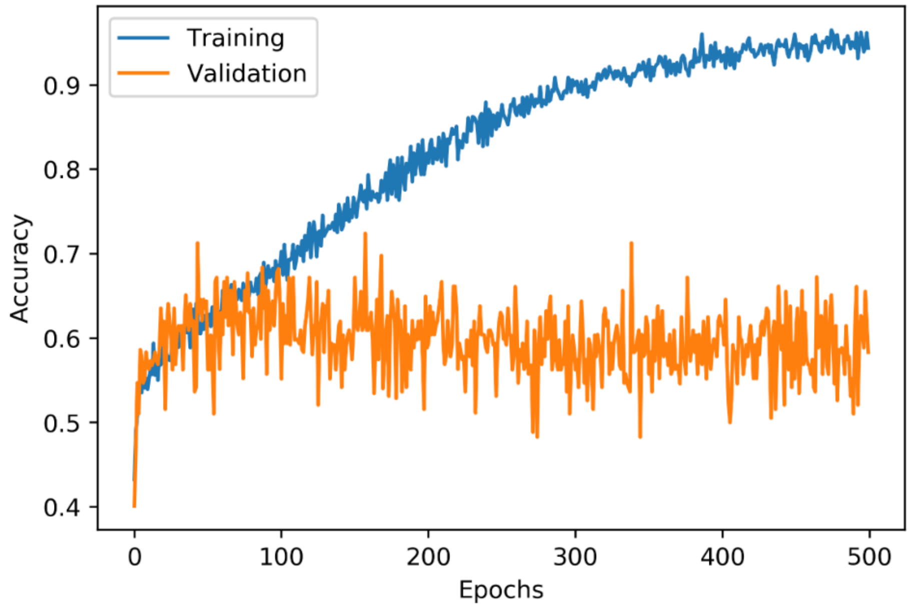

***Figure 11** Classifying Paintings with 500 Training Epochs*

Figure 10 shows how precision and loss during 100 training periods have been updated. Altogether different models were overfitted, and with much higher precision on the training system, this final iteration retains consistency than the validation package justly. In parallel, the loss of validation does not too much different from the loss of training.

If the model trains over more than 100 epochs, the precision of the training set rises to 100%, but stages at about 70%, as shown in Figure 11. A more aggressive regularisation prevents the model from learning and results from being worse.

As the dataset is low, we have transferred another 100 images from the validation set to the training set per class to see if accuracy increases with an increasing volume of data. Generally, the precision stays at 70%, but at some stage, as seen in Figure 12, it eventually reached 100 % accuracy. It may, though, be a random outcome, since the model will not converge later.

The efficiency of the drawings data collection is considerably better, with a similar but slightly modified CNN design. We only achieve a precision of 55 per cent here—the improvement in precision and failure during preparation, as seen in Figure 13.

They would also discuss whether performance can be enhanced by transfer learning.

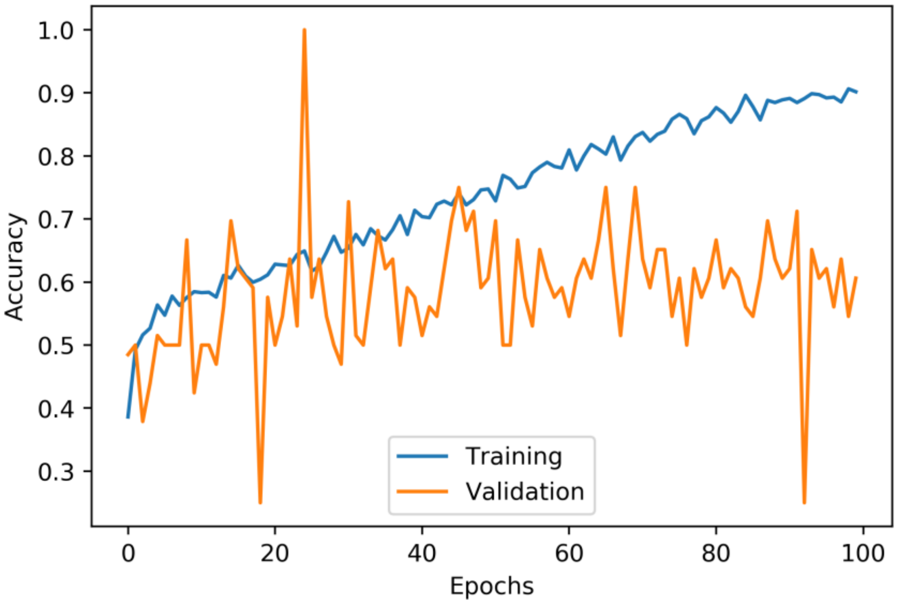

Figure 12 Smaller Validation Set Reaches 100% Accuracy

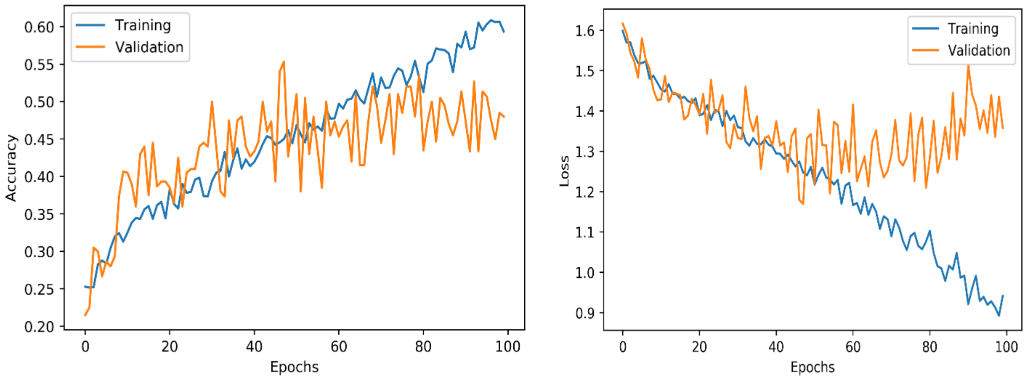

Figure 13 Classifying Drawings, Accuracy and Loss Graph

Augmenting Transfer Learning Models

We use the current VGG19 model (Simonyan & Zisserman, 2014) to improve the precision and apply it to our classification issue instead of pursuing studies on unique CNN architectures. Regrettably, because of the shortage of computing resources on our desktop Computer, the VGG model itself can not be run so it will know all the hyperparameters in this dataset. Nonetheless, we will reuse much of the weight acquired from the 14 million-image ImageNet dataset and just raise some of the weight layers in our dataset.

For the 1st third of the architecture, we abandon the VGG19 and let the remainder of the architecture learn for our dataset. We also add two large neurons, one in the lower section of the system, to the VGG design and train the classifier with a Stochastic Gradient Descent separately with data sets of paintings and drawings.

Transfer learning allows us to improve the accuracy of 70% to 75% of paintings (figure 14) and of drawing datasets (figure 15 for detailed losses and accuracy visualisations), from 55% to 62%. (see figure 15).

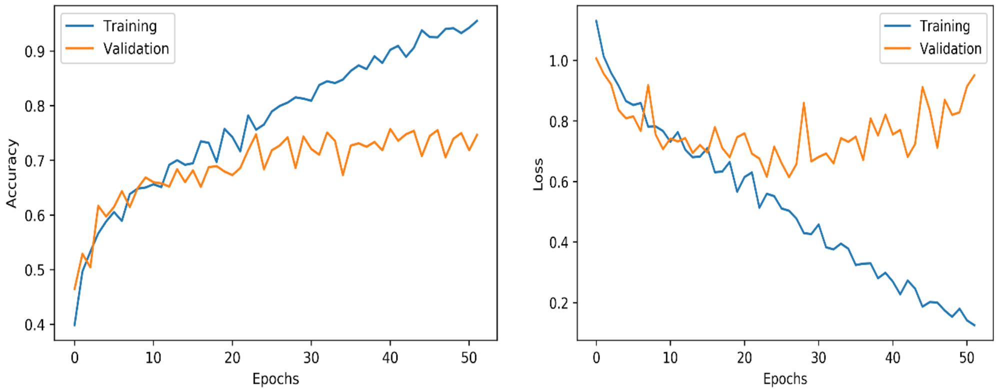

Figure 14 Classifying Paintings with VGG19, Accuracy and Loss Graph

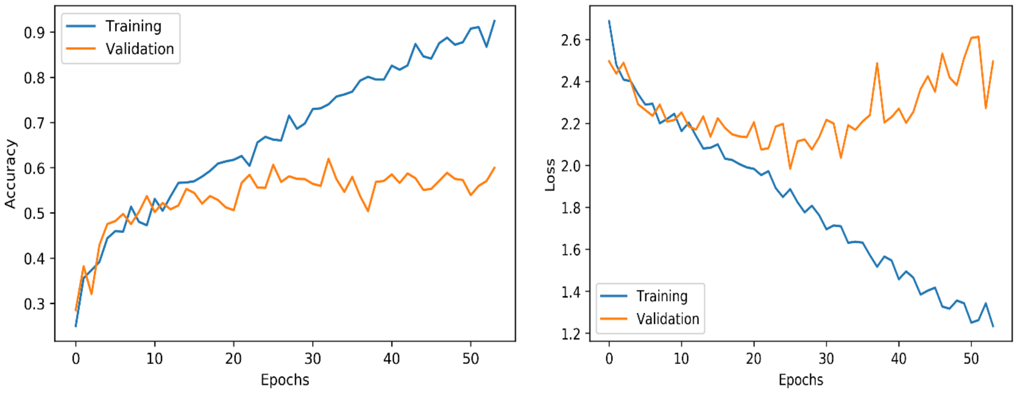

Figure 15 Classifying Drawings with VGG19, Accuracy and Loss Graph

Conclusion and Future Work

The question of automatically classifying artworks depending on their date of origin was addressed in this Research Paper. We also collected a list of nearly 7,000 colour paintings from different historical times, along with 7,500 grayscale paints and a related dataset.

Through using CNC, we could achieve an accuracy of 76 per cent in grading pictures into three historical phases (yourself in 1400-1759, 1760-1870 and 1870 to the present day), whereas a precision of 62 per cent could be achieved in classifying pictures in five historical phases (1400-1599, 1600-1650, 1651-1719, 1720-1799, 1800).

This distinction is a complicated issue because there are few picture samples in the dataset. To order to avoid the creation of more complicated CNN architectures, we have also equipped neural network models on a computer. Research on this deeper classification issue might theoretically increase its reliability, but a more extensive dataset might hopefully lead to more reliable performance.

Further research may be performed in the area of the transference of the neural types, which Gatys et al. first introduced in 2015, where a neural network is capable of catching the visual style of an image and converting it into a specific picture because this method involves substantial study and testing in philosophy and theoretical application.

Methodology

We present the preliminary findings for both unattended in this project Training and controlled methods to recognising “how art is perceived by the computer” and classifying visual art. Learning supervised is a machine learning activity, which provides data, feedback, and the mapping method to map output information. Two common problems in supervised learning are classification and regression. The computer, on the other hand, learning a feature that defines the unlabelled data structure in unattended computing. Unchecked research was explicitly used to address the tasks of clustering and representation. For picture comprehension, both directed and unregulated forms, Convolutional Neural Networks (CNN) is used significantly. The neural networks of convolution involve several layers for different functions. Conveyor layers can sense rims and shapes of the figures and packages to avoid overfitting of the layout. We use ResNet-50, which overcomes both existing cutting-edge technologies and even human performance in the ImageNet dataset image recognition, to perform the study of this project.

The approach is influenced by the simulation of data incorporation and time analysis. Every visual research has been inserted into a space vector of 2048 dimensions using a previously trained ResNet50 trained on the corpus ImageNet (a corpus of 1.2 M of original web images). Since the model is pre-trained by natural photographs, the depiction is meant to be generic, with the boundaries and shapes of the characters being recorded. To reduce the dimension and track the first and second compounds as heat maps to relay information, we conduct primary component analyses. We use the same ResNet50 pre-trained model to train the classification.

Expected Result

In this project, we would be able to present preliminary results from WikiArt, the Metropolitan Museum of Art and Arts focused on incorporation into the field of modern artworking. We also reached high accuracy through the usage of a CNN system for classification of images by form and low accuracy classifying images by generation. The coarse granularity of data and the diversity of art styles over a single era are potential reasons for this precision concerning time frame.

The performance of the form classifier offers the means to expand automated metadata for digital art collections, which may be used to replace the missing metadata in their data set by correctly classifying the category and to increase the pace at which items may be applied to the collection by machines being able to tag photos instead of needing an individual to tag them.

References

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

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

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

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

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

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

[7] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, p. 1627, 2010.

[8] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39–51, 2002.

[9] C. Wojek, P. Dollar, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, p. 743, 2012.

[10] H. Kobatake and Y. Yoshinaga, "Detection of spicules on mammogram based on skeleton analysis." IEEE Trans. Med. Imag., vol. 15, no. 3, pp. 235–245, 1996.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for

[12] fast feature embedding," in ACM MM, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012.

[14] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in CVPR, 2017.