Deep Learning-Based Approach for Illustration and Diagram Detection in Large-Scale Datasets of Historical Document Images

by

Zohreh HAJABEDI

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN SOFTWARE ENGINEERING
M.A.Sc.

MONTREAL, AUGUST 18, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Mohamed Cheriet, Thesis Supervisor
Department of Systems Engineering, École de technologie supérieure

Mr. Kim Khoa Nguyen, President of the Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mrs. Sylvie Ratte , Member of the jury
Department of Software and Information Technology Engineering,
École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

JULY 29, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my research supervisor, Professor Mohamed Cheriet, who has helped, motivated and supported me to commence and complete this research work in the last two years. It was a great opportunity and honour to work under his supervision.

I am also extremely grateful to my parents, my husband, my sisters, my brother and my little daughter, Rosha, for their love, understanding, patience, prayers and continuing support to complete this research work.

My special thanks to my sister Sama and also my friend Sara for introducing me to Professor Cheriet and Synchromedia Laboratory.

Approche fondée sur l'apprentissage profond pour la détection des illustrations et des diagrammes dans des bases de données à large échelle d'images de documents historiques

Zohreh HAJABEDI

RÉSUMÉ

Les manuscrits historiques contiennent des informations précieuses sur les cultures et les connaissances de l'être humain dans de nombreux domaines. Ces précieuses ressources doivent être préservées, entretenues et partagées. À cette fin, de nos jours, des référentiels de documents numérisés ont été créés à partir de ces ressources et, par conséquent, de larges volumes d'images de documents numérisés sont maintenant disponibles. Le défi est alors de récupérer des informations à partir des ressources numérisées extrêmement volumineuses. D'autre part, la diversité de la structure et de la mise en page de ces manuscrits anciens, ainsi que la détérioration habituelle des documents historiques, font de l'extraction des informations et de leur analyse une tâche difficile qui est peu susceptible d'être effectuée par des êtres humains.

Outre le contenu du texte, les documents historiques contiennent également des objets typographiques tels que des illustrations et des diagrammes qui portent des connaissances visuelles et soutiennent le contenu du document en fournissant une vue abstraite des concepts. Ces objets aident à comprendre le contenu du texte de manière plus productive. L'identification de ces objets typographiques nous renseigne sur la structure des documents. De plus, des informations sur les objets typographiques seraient utiles pour créer un index et des métadonnées pour les grands référentiels de documents numérisés.

En raison du récent succès des approches d'apprentissage profond dans les applications de vision par ordinateur, dans cette thèse, une approche basée sur CNN a été utilisée pour détecter les illustrations et les diagrammes et classer les images du document en fonction de la présence de ces objets typographiques. Le modèle proposé a été appliqué à de grandes bases de données d'images de documents historiques d'ECCO et de NAS. Ces deux bases de données contiennent respectivement plus de 32 millions et 500,000 images de documents anciens.

À l'instar des autres applications du monde réel, dans nos bases de données cibles, nous avions accès à un nombre limité de données étiquetées pour l'ensemble d'entraînement et de test. De plus, notre base de données d'entraînement est déséquilibrée et il y a une distribution inégale des classes. Pour faire face à ces problèmes et aussi pour atténuer le sur-apprentissage qui en résulte, nous avons rai forcé notre approche avec des techniques de régularisation et d'augmentation pour améliorer les performances. Le modèle final a obtenu des résultats prometteurs sur les grands ensembles de données ECCO et NAS.

Mots-clés: analyse d'image de document, images de documents historiques, détection d'illustration, détection de diagramme, classification d'image de document, deep learning, Jeu de données déséquilibré, augmentation

Deep learning-based approach for Illustration and Diagram detection in large-scale datasets of historical document images

Zohreh HAJABEDI

ABSTRACT

Historical manuscripts contain precious information regarding human being's cultures and knowledge in many different domains. These valuable resources need to be preserved, maintained and shared. To this end, nowadays, repositories of digitized documents have been created from these manuscripts and as a result, huge volumes of scanned document images are available. Retrieving information from extremely large digitized resources is the next concern. On the other hand, the diversity in structure and layout of these ancient manuscripts, as well as the deterioration that is usual in historical documents, make extracting information and analyzing them a challenging task that is unlikely to be done by human beings.

Besides text contents, historical documents also contain some typographical objects such as illustrations and diagrams which carry visual knowledge and support the document content by providing an abstract view of the concepts. These objects help to understand the text content more productively. Identifying these typographical objects gives us information regarding the structure of documents. Moreover, information about typographical objects would be beneficial in creating indexes and metadata for large repositories of digitized documents.

Due to the recent promising success of deep learning approaches in computer vision applications, in this thesis, a CNN-based approach has been used to detect illustrations and diagrams and classify the document images based on the presence of these typographical objects. The proposed model has been applied on large datasets of historical document images of ECCO and NAS. These two datasets contain over 32 Million and 500,000 ancient document images respectively.

Similarly to the other real-world applications, in our target datasets, we had access to only a restricted number of labelled data as training and test set. Furthermore, our training dataset is imbalanced and there is an unequal distribution of classes. To deal with these issues and also to alleviate the resulting overfitting, we have empowered our approach with regularization and augmentation techniques to improve the performance. The final model achieved promising results on the large datasets of ECCO and NAS.

Keywords: document image analysis, historical documents images, illustration detection, diagram detection, document image classification, deep learning, imbalanced dataset, augmentation

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

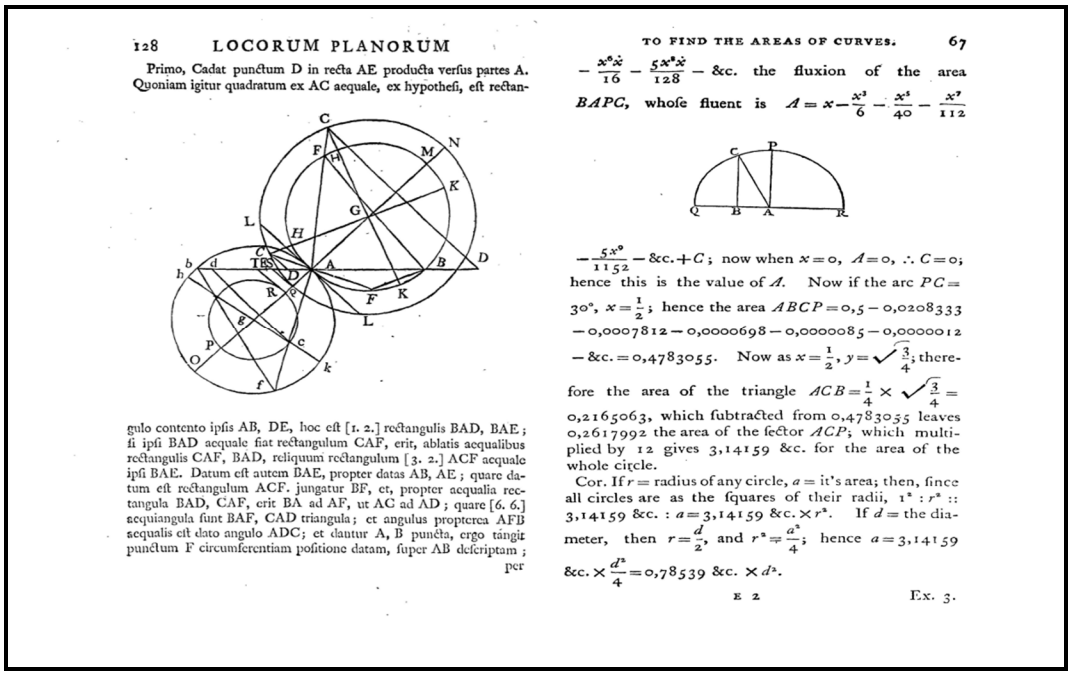
INTRODUCTION

0.1 Relevance and motivation

Historical and ancient documents are precious sources of knowledge regarding the human being's cultural heritage. These valuable resources need to be preserved, maintained and shared with interested researchers all over the world would (Cheriet, Farrahi Moghaddam, & Hedjam, 2013).

Historical documents are located physically in a variety of libraries around the world. To make them accessible for more people and also protect them from more degradation, digitizing these resources has become a top priority for their holders. As a result, nowadays tremendous volumes of scanned document images are available (Zhalepour, 2018). Due to the large volume of these digital libraries, human beings are unable to extract information from them. As a result, the precious information contained in these resources is prone to get lost or never seen (Kavasidis, et al., 2018).

On the other hand, retrieving information and also analyzing the contents of these scanned documents are challenging tasks regarding the various structures and formatting, besides the degradation, which is common in old documents (Mehri, 2015). Considering the mentioned challenges, nowadays, large numbers of researchers around the world have concentrated on automated document analysis approaches. Considerable numbers of these researches are focused on Natural Language Processing (NLP) which is engaged with syntax and semantics of the document texts to extract and summarize information from scanned documents. While the text is one of the most important ways to transmit information, sometimes graphical elements are far more effective (Kavasidis, et al., 2018). For instance, in scientific manuscripts, the experimental results can be expressed more specifically through tables and illustrations in comparison with pure text. Examples from our target datasets are shown in Figure 0.1 in which demonstrating the theory and concepts would not be possible without including diagrams. As a result, detecting these graphical elements as part of the document layout analysis process would be a vital step in information extraction.



128 LOCORUM PLANORUM

Primo, Cadat punctum D in recta AE producta versus partes A. Quoniam igitur quadratum ex AC aequale, ex hypothesi, est rectangulo contento ipsis AB, DE, hoc est [1. 2.] rectangulis BAD, BAE; si ipsi BAD aequale fiat rectangulum CAF, erit, ablatis aequalibus rectangulis CAF, BAD, reliquum rectangulum [3. 2.] ACF aequale ipsi BAE. Datum est autem BAE, propter datas AB, AE; quare datum est rectangulum ACF. jungatur BF, et, propter aequalia rectangula BAD, CAF, erit BA ad AF, ut AC ad AD; quare [6. 6.] aequiangula sunt BAF, CAD triangula; et angulus propterea AFB aequalis est dato angulo ADC; et dantur A, B puncta, ergo tangit punctum F circumferentiam positione datam, super AB descriptam; per

TO FIND THE AREAS OF CURVES. 67

$-\frac{x^6\dot{x}}{16}-\frac{5x^8\dot{x}}{128}$ - &c. the fluxion of the area $BAPC$, whose fluent is $A=x-\frac{x^3}{6}-\frac{x^5}{40}-\frac{x^7}{112}$

$-\frac{5x^9}{1152}$ - &c. $+C$; now when $x=0$, $A=0$, $\therefore C=0$; hence this is the value of A. Now if the arc $PC=30°$, $x=\frac{1}{2}$; hence the area $ABCP=0,5-0,0208333-0,0007812-0,0000698-0,0000085-0,0000012$ - &c. $=0,4783055$. Now as $x=\frac{1}{2}$, $y=\sqrt{\frac{3}{4}}$; therefore the area of the triangle $ACB=\frac{1}{4}\times\sqrt{\frac{3}{4}}=0,2165063$, which subtracted from $0,4783055$ leaves $0,2617992$ the area of the sector ACP; which multiplied by 12 gives 3,14159 &c. for the area of the whole circle.

Cor. If $r=$ radius of any circle, $a=$ it's area; then, since all circles are as the squares of their radii, $1^2:r^2::3,14159$ &c. $:a=3,14159$ &c. $\times r^2$. If $d=$ the diameter, then $r=\frac{d}{2}$, and $r^2=\frac{d^2}{4}$; hence $a=3,14159$ &c. $\times\frac{d^2}{4}=0,78539$ &c. $\times d^2$.

E 2 Ex. 3.

Figure 0.1 Examples of the necessity of diagrams along with text

Categorizing document images into specific categories is the first and necessary step for document analysis and information extraction (Kang, Kumar, Ye, Li, & Doermann, 2014). As mentioned earlier, due to the importance of layout in document image analysis, we can consider the layout and graphical structure of document images as a discrimination factor to classify them. This also provides us information to create meta-data for the document image repositories. Considering the large volume of historical manuscripts images, creating informative meta-data to describe and introduce these resources more efficiently is a must. This meta-data information could be descriptions of the content or layout of the images. It creates the possibility of categorization and indexing of the huge historical document repositories (Zhalepour, 2018).

Furthermore, indexing and categorizing document images based on layout analysis can be a step before OCR[1], providing OCR modules helpful information regarding the visual structure of the documents, leading them to more precise content analysis and information extraction (Harley, Ufkes, & Derpanis, 2015).

[1] Optical Character Recognition

0.2 Thesis focus, problem statement and thesis objectives

The four major typographical objects which are more informative and important in document image analysis are ‘footnotes’, ‘tables’, ‘illustrations’ and ‘diagrams’ (Zhalepour, 2018). These graphical elements provide a visual summary of the manuscript’s most important content (Saha, Mondal, & Jawahar, 2019). In 2018, (Zhalepour, 2018) has addressed the detection of ‘footnotes’ and ‘tables’ in large-scale historical documents.

In this thesis, our focus is on detection of ‘Illustration’ and ‘Diagram’ in large-scale historical document image repositories. The common definitions of these two informative typographical objects are as follows:

Illustrations: “An illustration is a decoration, interpretation or visual explanation of a text, concept or process” (Illustration Definition, 2021). Illustrations, as one of the most important visual objects in manuscripts, play a significant role to make humans more engaged with the document’s content. Sometimes they present an abstract view on the complicated and long content and provide the reader with a better perspective to understand the underlying text.

Some examples of illustrations in our datasets of historical document images are shown in Figure 0.2.

Diagrams**:** diagrams can be specified as a subcategory of illustrations. “The essence of a diagram can be seen as:

- A display that does not show quantitative data (numerical data), but rather relationships and abstract information,
- With building blocks such as geometrical shapes connected by lines, arrows, or other visual links” (Diagram Definition, 2021).

With visualizing systems patterns and structures, illustrations and diagrams, create the possibility of deeper analyzing and understanding the scientific concepts.

Four examples of diagram pages in our dataset are displayed in Figure 0.3.

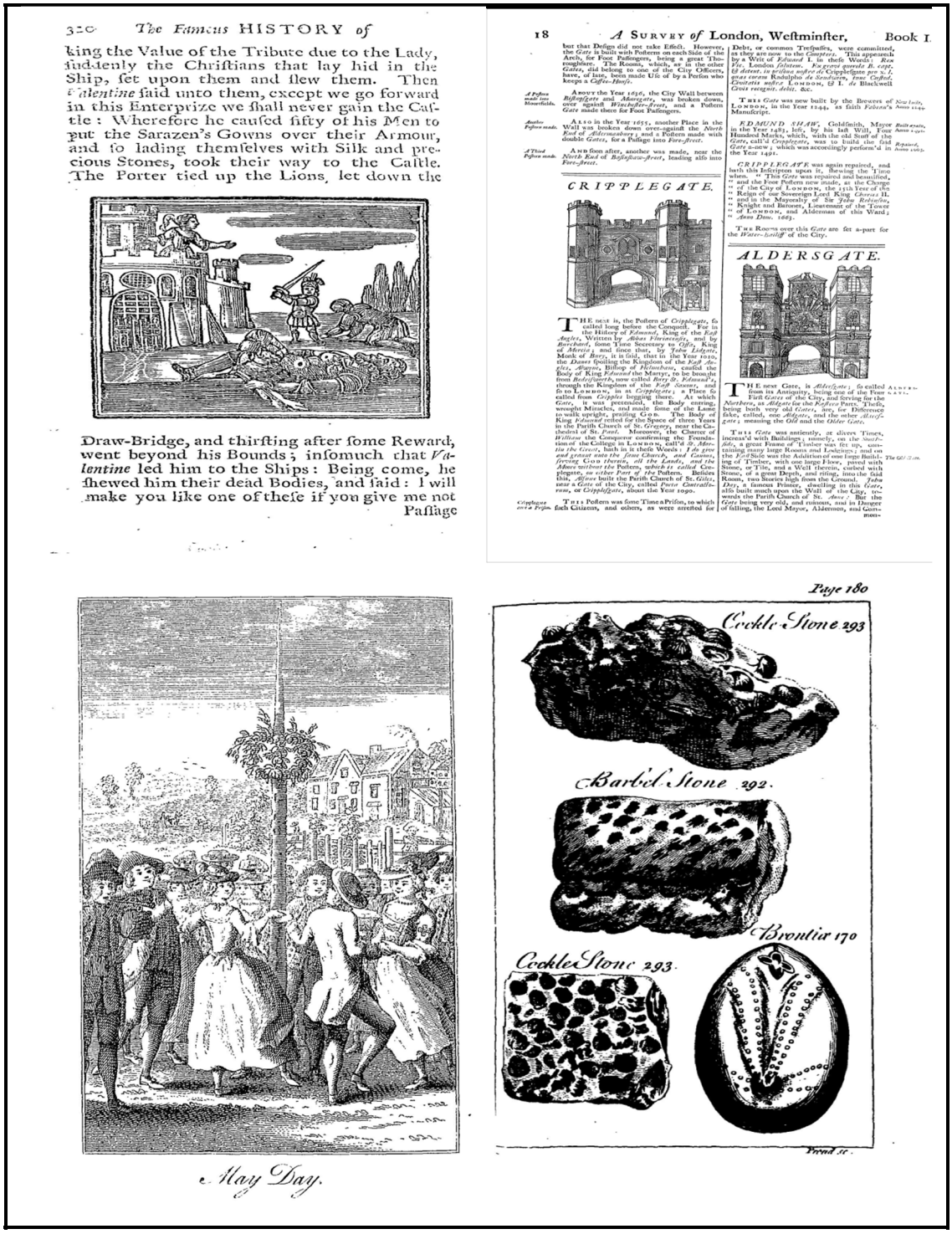


Figure 0.2 Examples of illustrations in our dataset of historical document images

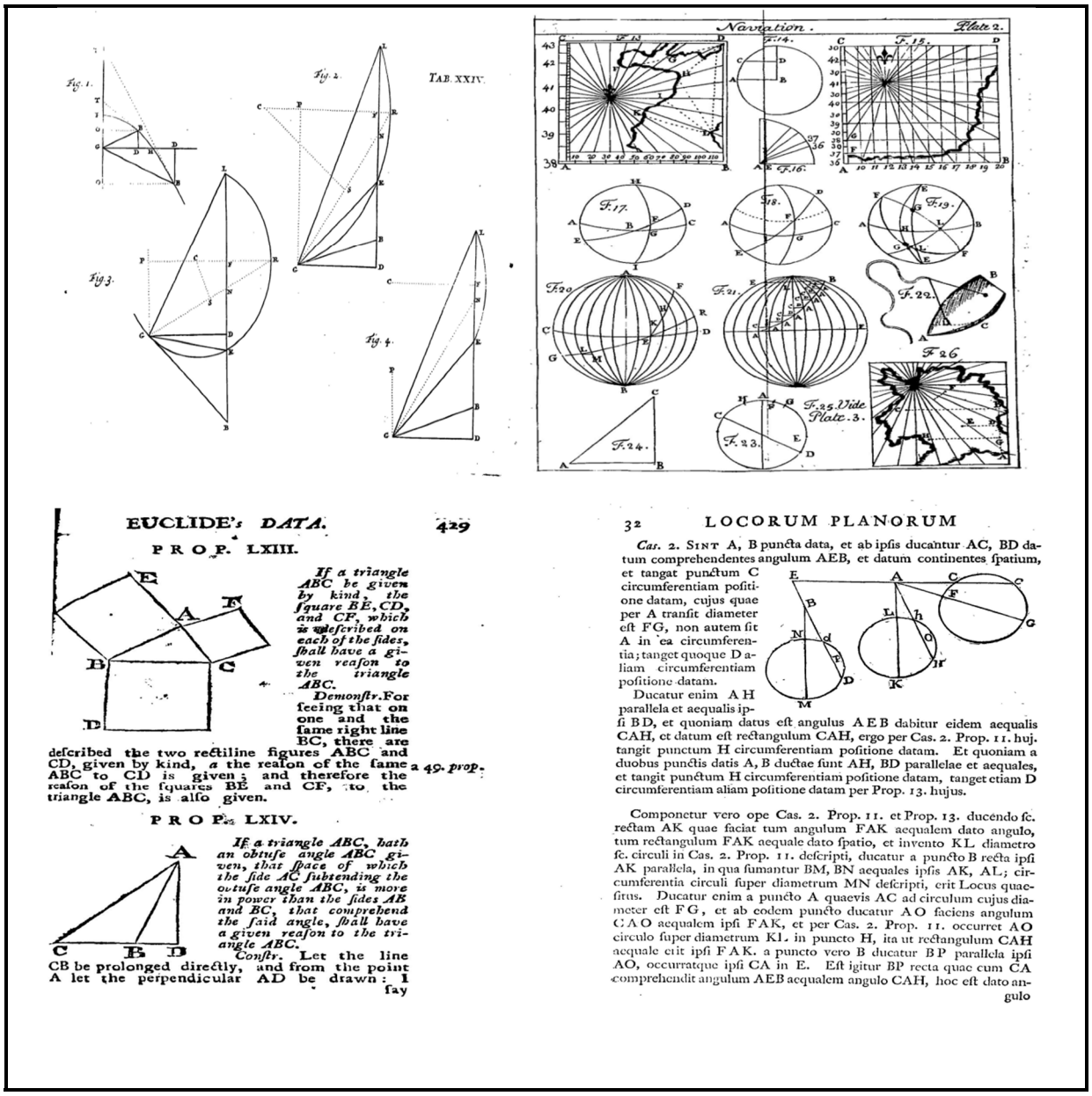


Figure 0.3 Examples of diagrams in our dataset of historical document images

In this thesis, we aim to retrieve information from historical document images with a focus on detecting illustrations and diagrams which are among the most important graphical information containers in documents. In order to achieve this objective, we are going to categorize document images based on the presence of illustrations and diagrams in them. The main issues should be addressed to achieve this objective are as following:

- Intra-class variability and various structures,

  Illustrations and diagrams might have various structures in different manuscripts. This issue will result in an intra-class variability problem which makes the classification task more challenging and prone to errors. As it can be seen in Figure 0.4, the three images are labeled as illustration pages in ground truth but they have totally different structures and layouts.

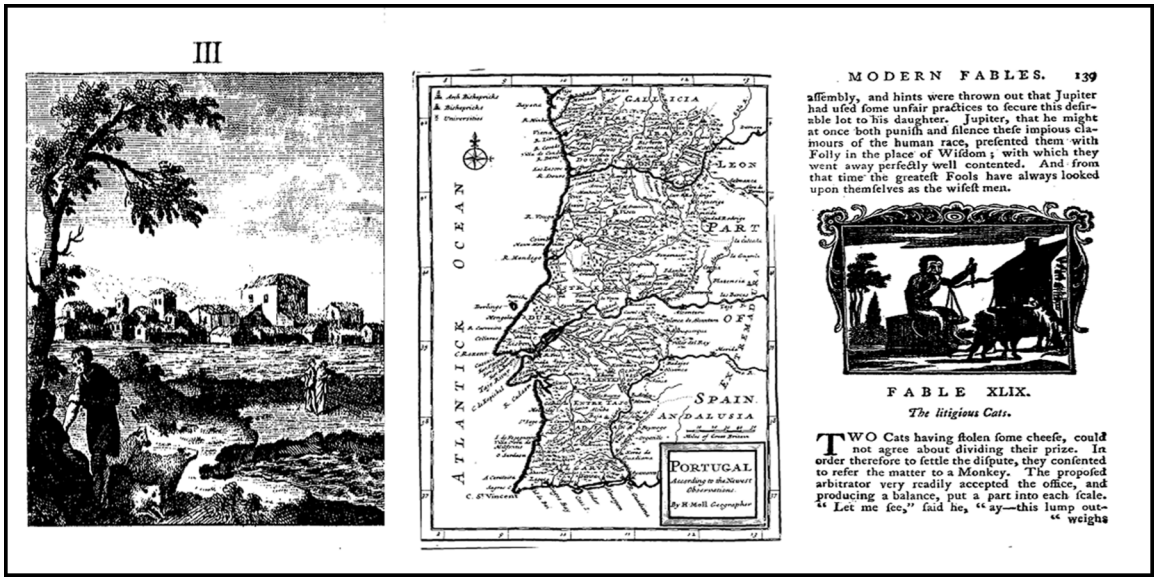
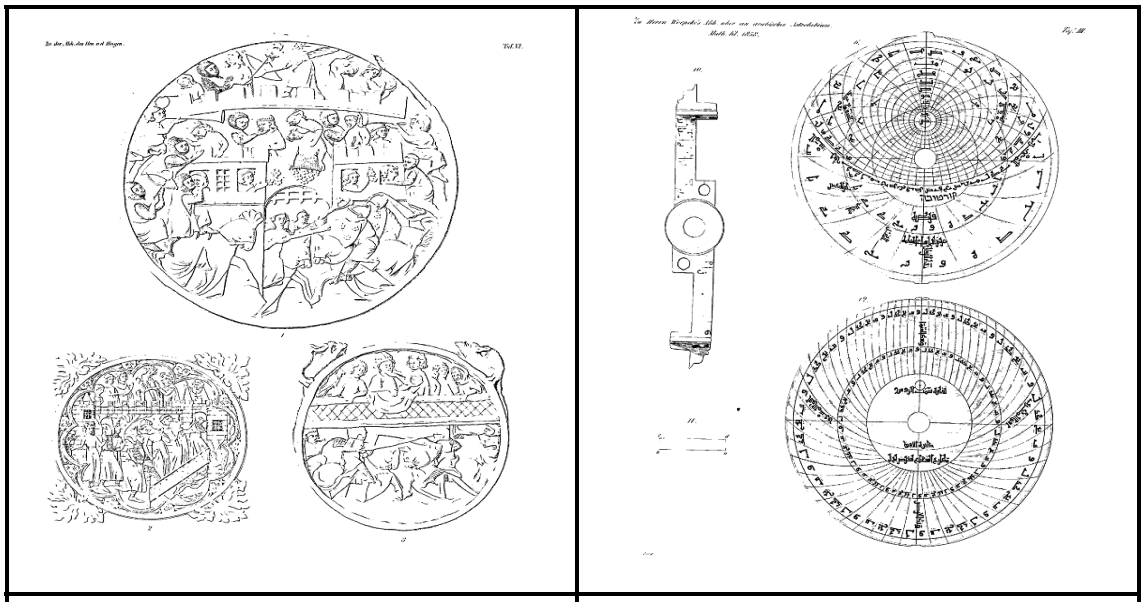


Figure 0.4 Intra-class variability issue

- Inter-class similarity,

  There are similarities between some visual objects in different classes. Some diagrams, for instance, have a comparative structure with tables and prone the classification problem to inter-class similarity which inevitably would affect the performance of the model. An example of structure similarity between illustration and diagram is showed in Figure 0.5.



a. Illustration | b. Diagram

Figure 0.5 Inter-class similarity

- Scarcity of labelled data,

  The same as most of the real-world problems, in our target domain we have access to restricted labelled samples to train our model. This could result in deficiency in learning capacity and generalization power of the model.

- Imbalanced dataset.

  Another issue with our labelled dataset is imbalance classes. In imbalanced datasets, the distribution of classes is not uniform and we have over sampled and under sampled classes that leads model to have difficulties in detection of under sampled class specifications.

Thesis Objective: The main objective in this thesis is to propose a reliable framework to detect visual objects of illustrations and diagrams in large-scale unlabeled datasets of historical document images. We formulate our sub objectives to empower our model with techniques as presented in the following:

1. To extract a reliable representation scheme for document images to conquer intra-class variability and inter-class similarity,

2. To tackle restricted amount of labelled data to train the model,
3. To alleviate the effect of imbalanced training dataset,
4. To validate the proposed approach on large-scale datasets.

Document image analysis approaches can be divided into two categories; 'region-based analysis' in which the document image will be segmented into some regions such as footnote and body and the discriminative features would be extracted independently in each region. The second category would be 'whole image analysis' in which the whole document image would go under the feature extraction process at once. Due to our goal in this thesis, we are not concerned about the exact localization of objects in each document image and so our proposed approach is not region-based and resides in the second category of 'whole image analysis'.

From a different point of view, document image analysis methods can be categorized into; methods relying on 'handcrafted features' and methods based on 'automatic machine-learned features' (Harley, Ufkes, & Derpanis, 2015). Our target datasets, in this thesis, are two large-scale repositories of historical manuscripts. There are various layout structures in these historical aged documents and due to variable layouts, degradation, noises and marginalia[2], detecting objects in them cannot rely on any specific assumption regarding the objects structure and no prior knowledge is applicable on the specification of the objects which reside in these documents. Considering these characteristics, the handcrafted features-based methods would not fit our task of classification and in this work, we address the classification of document images with an 'automatic machine-learned features' approach. To implement automatic feature extraction and overcome the intra-class variability and inter-class similarity issues, discussed earlier, we have used a CNN-based network as a data-driven and heuristic-independent approach.

[2] Marginalia (or apostils) are marks made in the margins of a book or other document.

0.3 Overview of the thesis structure

This thesis is organized into four chapters. In the Introduction chapter, the motivation and the thesis focus are presented. It also introduces the issues and the objectives. The First chapter focuses on reviewing the state-of-the-arts in document image classification. Chapter 2 is dedicated to explaining the details of the proposed deep learning-based approach to detect illustrations and diagrams in large-scale document images databases. Chapter 3 covers the experiments and results of the proposed approach on the datasets and also includes discussion and analysis of the results. Finally, we conclude this work with conclusions, recommendations and future works sections.

CHAPTER 1

STATE-OF-THE-ART

In this chapter, we will overview some of the state-of-the-art works that have been done in the domain of visual object detection in document images. We organized these works into two categories of ‘Handcrafted feature extraction-based approaches’ and ‘Deep learning-based feature extraction-based approaches’.

1.1 Handcrafted feature extraction-based image classification approaches

Baluja & Covell (2009) proposed an approach in which SIFT-based local features have been used to detect images and line drawings in scanned documents. They applied multiple classifiers trained via AdaBoost to classify document images based on the presence of images and line drawings. They have used local image features because these features are more informative in terms of local information content, and besides, they are reliable even when local and global perturbations such as rotation, skew, and noise is made to images. However, SIFT algorithm is mathematically complicated and it’s dramatically slow and is not applicable on large-scale datasets.

Kamola et al. (2014) presented a comprehensive solution for document structure recognition. Their whole process is categorized into two main steps of ‘localizing the elements’ and ‘labelling the detected elements’ into ten classes of ‘Abstract’, ‘Author’, ‘Caption’, ‘Header’, ‘Page Footer’, ‘Page Header’, ‘Paragraph’, ‘Table’, ‘Title’ and ‘Graphic elements: line, picture, diagram, scheme, chart’. The input to their method is the image of a document page and the output is an XML file listing the elements bounding and labelling. This method is a rule-based approach that had to set some assumptions regarding the textual and graphical specification of the page elements both in the segmentation step and in labelling. Meanwhile, due to the variability of page object structures in different manuscripts, these assumptions wouldn’t work well in real-world problems. On the other hand, the dataset used in this work are pages that are acquired from the Internet. They assumed that the pages are preprocessed

and noise-free and they also considered some requirements such as high resolution, uniform background colour and distinct contrast between background and foreground in document images. This is what we cannot always rely on in the real world and specifically historical aged document images.

1.2 Deep Learning feature extraction-based image classification approaches

Kang et al. (2014) proposed a CNN-based approach to classify document images into ten classes of ‘ad’, ‘email’, ‘form’, ‘letter’, ‘memo’, ‘news’, ‘note’, ‘report’, ‘resume’ and ‘scientific’ based on the document layout. They have mentioned that they’ve employed CNNs due to their capability to learn hierarchical layout features and identifying complex document layouts. They also have equipped their model with Rectified Linear Units (ReLU) to speed up training. Moreover, since dropout reduces the possibility of overfitting by imposing some noise on the training samples, they employed dropout to alleviate overfitting.
Harley (2015) used CNN-based feature extraction to classify the document images. They’ve employed CNNs for both ‘region-based feature extraction’ and ‘whole image feature extraction’. To be more precise, in their model, five CNNs have been used, from which, four CNNs are targeted to extract region-based features in four regions of interest of ‘header’, ‘right body’, ‘left body’ and ‘footer’. The fifth CNN works on the whole document image and the final feature vector is created by concatenating the outputs of all five CNNs. To conquer the scarcity of labelled data in document image scope, they facilitated transfer learning and pre-trained their model on the ImageNet 2012 (Stanford Vision Lab, 2020) which contains more than a million labelled images from nature.
Kavasidis et al. (2018) proposed a fully convolutional neural network for detecting tables and charts in document images. Their method does semantic image segmentation and as a predictor, they used a fully connected Conditional Random Field (CRF) (Krahenbuhl & Koltun, 2011). Since tables and charts are usually the most salient regions in the image, to make their model able to learn basic visual cues, they first trained the model on saliency detection datasets and then adjusted it on the target document images dataset.

Hu et al. (2019) employed the LeNet-5 network (Lecun, Bottou, Bengio, & Haffner, 1998) as a deep learning-based model for document image classification. LeNet-5 is a simple CNN consisting of three convolutional layers, two subsampling layers and two fully connected layers. They have applied some changes to the structure of the LeNet-5 to improve its performance on their document image classification problem. For instance, their model has been equipped with L2 regularization to conquer the over-fitting phenomenon. They also implemented the idea of cross-connected learning, in which the features detected by the first layer are connected to the last convolutional layer and also fully connected layer and this way these layers can use more diverse low-level features in their calculations and predictions. The datasets that have been utilized to evaluate this model are CIFAR-10 (Krizhevsky, 2009) and Fashion Mnist (Fashion-MNIST Dataset); both include images from nature, cars and clothes.
In 2019, a deep learning-based framework is presented by Saha et al. (2019) to localize and detect graphical objects in document images. They use Faster R-CNN for the two steps of proposing graphical regions and prediction of class labels. Their model faces difficulties when the objects' structures are ambiguous. For instance, sometimes it misclassifies tables as figures and vice versa, or some paragraphs are predicted as tables by the model.

In the previous works, datasets used for training and evaluating the models are usually common preprocessed image datasets and not heterogeneous aged documents images. Furthermore, the size of our target dataset is extremely larger that the common used image datasets. Based on our problem's specification, hand-crafted feature extraction approaches wouldn't fit on our requirements. We need to design a deep learning-based approach and empower it with techniques to tackle the discussed issues and complexities mentioned earlier in the Introduction chapter.

CHAPTER 2

DEEP-LEARNING BASED APPROACH TO DETECT ILLUSTRATIONS AND DIAGRAMS IN LARGE-SCALE HISTORICAL DOCUMENT IMAGES

In all classification problems, feature extraction plays an important role in the performance of the final task. Traditional image classification problems were based on hand-crafted features. Meanwhile, variability in the structure of historical ancient documents does not allow us to design specific features which are applicable for all manuscripts. So, considering the most significant benefit of Convolutional Neural Networks as being independent of hand-crafted features and its ability to do unsupervised or semi-supervised and hierarchical feature learning (Wang Z. , 2015), in this thesis, we've adopted a CNN-based model to detect illustrations and diagrams in documents images.

The whole process of our work is done with a deep learning-based framework and in two major tasks:

- Task 1 : Illustration detection:

  It is a two class classification problem: 'Illustration' and 'NON'.

- Task 2 : Diagram detection

  It is a three class classification problem: 'Illustration', 'Diagram' and 'NON', which aims to detect diagrams as a subcategory of detected illustrations in task1

In the following sections, we present the proposed approach and the techniques that were employed to address our objectives in this thesis.

2.1 Convolutional Neural Network

First CNN was presented in 1998 by Yann LeCun (Lecun, Bottou, Bengio, & Haffner, 1998). Machine learning techniques, especially Neural Networks, have received much interest in designing pattern recognition systems in recent years. Traditional pattern recognition systems consisted of two main modules of 'fixed feature extraction' and 'trainable classifier'. The overall structure of traditional pattern recognition systems is illustrated in Figure 2.1. The feature extractor mainly relies on prior knowledge and is usually very specific to the task.

Moreover, since it is handcrafted, it takes most of the effort in the designing of the overall system (Lecun, Bottou, Bengio, & Haffner, 1998).

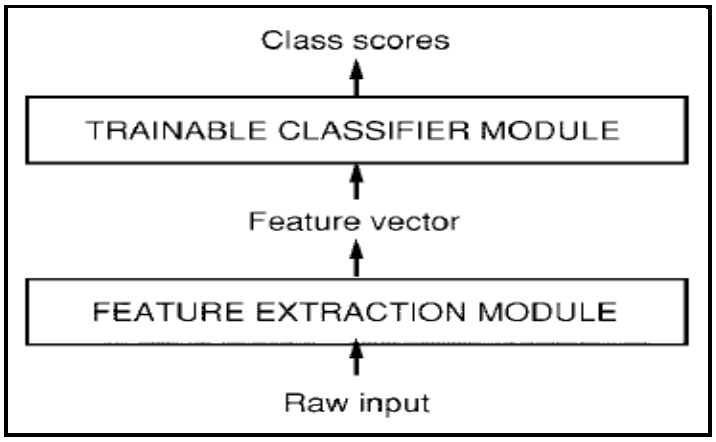


Figure 2.1 Traditional Pattern Recognition Systems
Taken from Lecun et al. (1998)

The main idea behind introducing Convolutional Neural Networks is designing pattern recognition systems in which the hand-crafted feature extractor module is substituted with an automatic feature extractor.

The CNNs have the common neural networks basis with two prominent innovations of ‘convolutional blocks’ and ‘pooling blocks’ which are aimed mainly to do automated feature extraction and dimensionality reduction, respectively (Fabio, 2018). A common CNN has the structure shown in Figure 2.2.

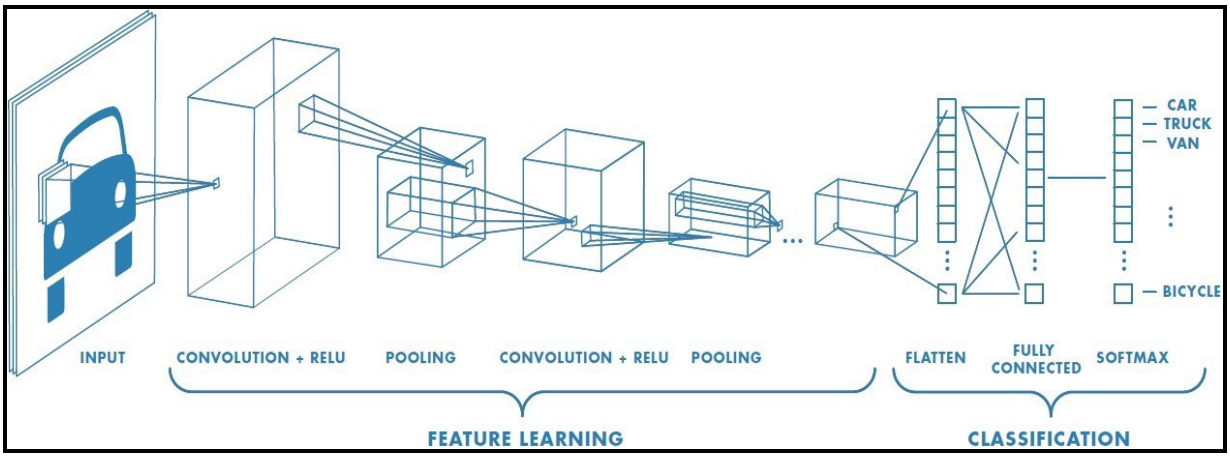


Figure 2.2 Convolutional Neural Network architecture
Taken from Saha S. (2018)

2.1.1 Convolution

Convolution is a mathematical function with two inputs that in computer vision problems these two inputs are ‘an image matrix’ and ‘kernel’. The kernel is a small matrix that processes the image matrix and allows us to process small portions of the image. (Goodfellow, Bengio, & Courville, 2016) Figure 2.3, illustrates an example of a convolution operator.

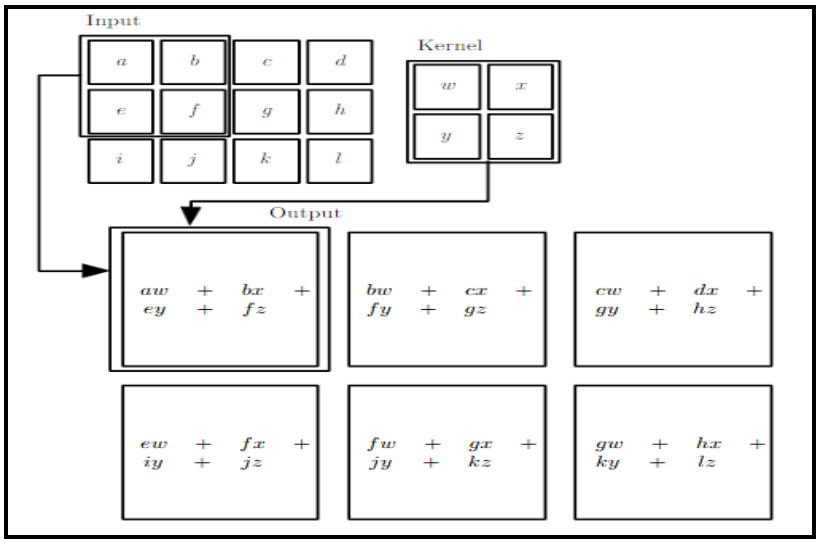


Figure 2.3 Convolution operator

Taken from Goodfellow et al. (2016)

In some circumstances, for instance, for a character recognition problem with small-sized images, ordinary fully-connected neural networks would be able to gain some success. However, there are still considerable drawbacks. If the image size is large, the number of neurons in hidden layers would be large, leading to an increase in the capacity of the network and accordingly requiring more training data which in most cases is not easily accessible. On the other hand, the main inefficiency of the common neural networks in image recognition is their lack of an invariance mechanism regarding the translations and local distortions of the input image. On the contrary, in CNN, the weight matrix strides over the whole parts of the image and thereby captures the relevant feature in every part of it. This makes the CNN architecture resistant to translations and local distortions of the input image (Lecun, Bottou, Bengio, & Haffner, 1998).

Convolution operation makes use of three major key concepts which could enhance a machine learning system: 'sparse interactions', 'parameter sharing' and 'equivariant representations' (Goodfellow, Bengio, & Courville, 2016).

- Sparse interaction: in the traditional neural networks, to calculate the output of a layer there is a separate weight parameter that corresponds to each input unit therefore there are interactions between each input unit and output unit. However, in

convolutional neural networks, kernels or parameter matrixes are chosen smaller than input which can detect simple and meaningful features out of the input. It results in reducing computation complexity and memory requirements. Sparse connectivity is visualized in Figure 2.4.

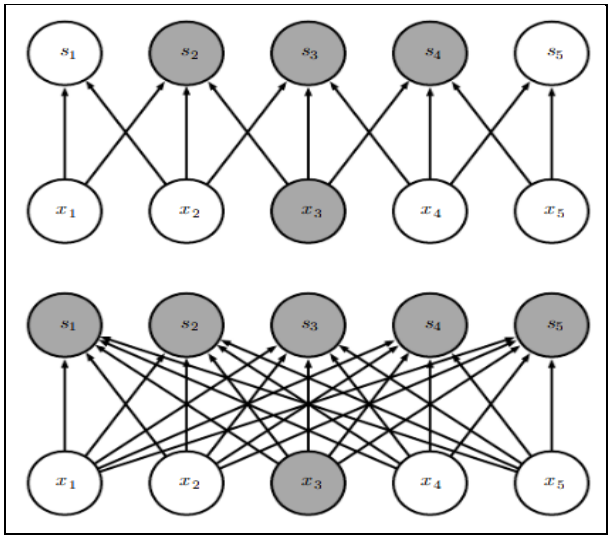


Figure 2.4 Sparse connectivity on the top versus tight connectivity on the bottom
Taken from Goodfellow et al. (2016)

- Parameter sharing:
  In traditional neural networks, to calculate the output of a layer, each parameter of the weight matrix is visited just once. On the other hand, in CNNs, the filter (weight matrix), slides through the input and each parameter of the kernel is used at every position of the input and this is called parameter sharing.
- Equivariant representation: the parameter sharing which is done in CNNs equipped these networks with the 'equivariance to translation' property which indicates changes in the inputs cause the output change in the same way.

Deep convolutional neural networks facilitate multiple levels of abstraction in the learned features. Features extracted in the early layers contain low-level information, mostly related

to spatial properties of the objects such as edges. On the contrary, the latter layers abstract semantic information (Ma, Huang, & Yang, 2017). In Figure 2.5, an example regarding the feature learned through the layers of a CNN is illustrated.

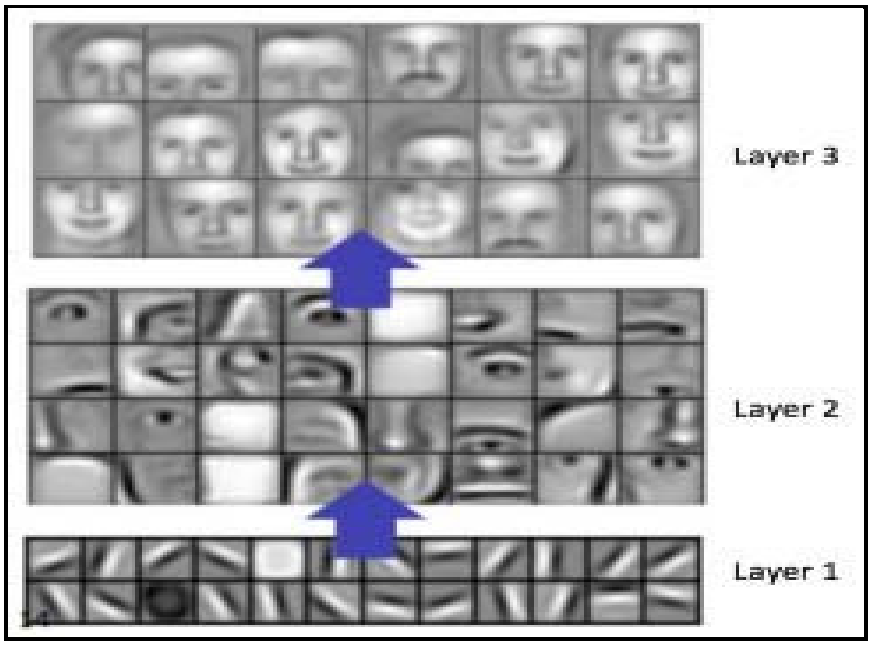


Figure 2.5 Learned features from a Convolutional Neural Network
Taken from Saad et al. (2017)

As it can be seen in the above figure, edges are determined in the first layer, simple shapes are detected in the second layer and finally, the higher-level features such as shapes and faces are detected in the last layer (Saad, Tareq, & Saad, 2017).

2.1.2 Pooling

As mentioned in the section 3.1, pooling is the second important concept that discriminates CNNs over traditional neural networks. The main goal of pooling is to reduce the complexity of subsequent layers through down-sampling. In the CNN, the pooling algorithm will apply to the output of convolutional layers to reduce the size of the feature maps. One of the most common pooling methods is max pooling in which the input is divided into rectangular sub-

regions and the algorithm just returns the maximum value in each rectangular. An example is shown in Figure 2.6.

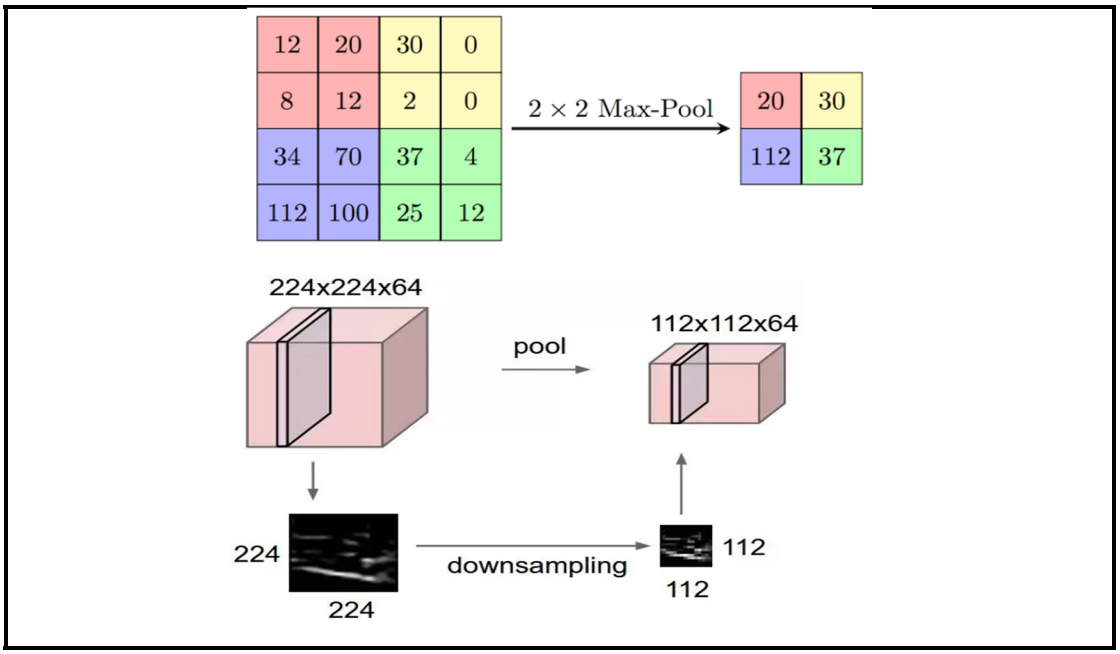


Figure 2.6 Max pooling

Taken from Max-pooling (2018)

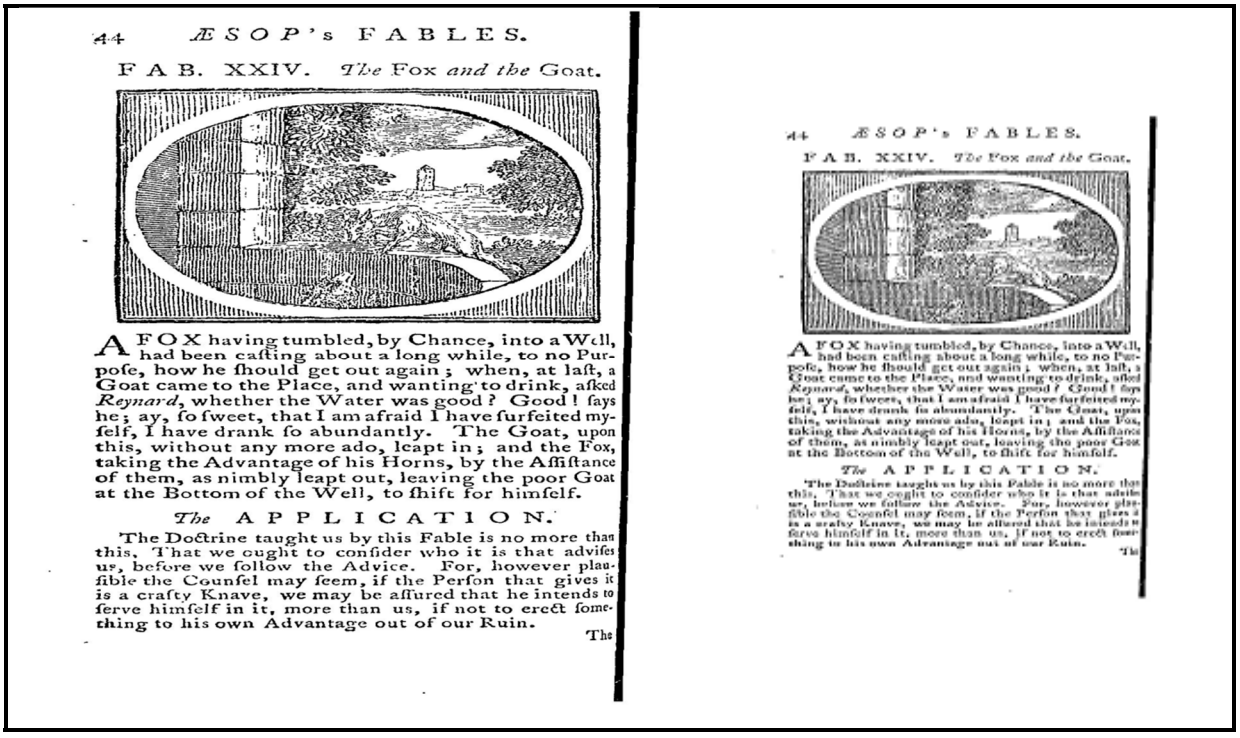
By applying pooling on the feature maps, the model would be invariant to small translations in input. It indicates that if small changes apply to the input, the pooled output would not change. This property would be beneficial whenever we are only interested to detect whether an object is present in an image and it's not important to detect the exact position of the object in the image. (Goodfellow, Bengio, & Courville, 2016)

In 2012, Krizhevsky (2012) trained a deep CNN that outperformed the state-of-the-art and won the contest of classification 1.2 million images in the ImageNet LSVRC-2010. Their CNN had five convolutional layers and 60 million parameters. They also applied the dropout method to alleviate the overfitting. This achievement triggered much interest in the application of CNN in computer vision problems.

2.2 Preprocessing

In machine learning-based solutions, the prominent prerequisite step before applying the algorithm on the target dataset is preprocessing data. It is an essential step to make the data fit into the algorithm and also contribute to the performance of the learning algorithm.

In this thesis, the document images of our datasets of ECCO and NAS have different sizes in ranges [1:3000] × [1:3000] which are too large to be fed to the CNN model. Furthermore, the images should be in the same size to be compatible with the CNN model architecture. So we down sampled the images into the size of 224 × 224. In this new size, although the details such as text character cannot be distinguished clearly, the overall structure of the document images is preserved and this is what we need to categorize the images in this work. Figure 2.7, shows a document image before and after down sampling.


44 ÆSOP's FABLES.

FAB. XXIV. *The* Fox *and the* Goat.

A FOX having tumbled, by Chance, into a Well, had been caſting about a long while, to no Purpoſe, how he ſhould get out again; when, at laſt, a Goat came to the Place, and wanting to drink, aſked *Reynard*, whether the Water was good? Good! ſays he; ay, ſo ſweet, that I am afraid I have ſurfeited myſelf, I have drank ſo abundantly. The Goat, upon this, without any more ado, leapt in; and the Fox, taking the Advantage of his Horns, by the Aſſiſtance of them, as nimbly leapt out, leaving the poor Goat at the Bottom of the Well, to ſhift for himſelf.

The APPLICATION.

The Doctrine taught us by this Fable is no more than this, That we ought to conſider who it is that adviſes us, before we follow the Advice. For, however plauſible the Counſel may ſeem, if the Perſon that gives it is a crafty Knave, we may be aſſured that he intends to ſerve himſelf in it, more than us, if not to erect ſomething to his own Advantage out of our Ruin.

The

44 ÆSOP's FABLES.

FAB. XXIV. *The* Fox *and the* Goat.

A FOX having tumbled, by Chance, into a Well, had been caſting about a long while, to no Purpoſe, how he ſhould get out again; when, at laſt, a Goat came to the Place, and wanting to drink, aſked *Reynard*, whether the Water was good? Good! ſays he; ay, ſo ſweet, that I am afraid I have ſurfeited myſelf, I have drank ſo abundantly. The Goat, upon this, without any more ado, leapt in; and the Fox, taking the Advantage of his Horns, by the Aſſiſtance of them, as nimbly leapt out, leaving the poor Goat at the Bottom of the Well, to ſhift for himſelf.

The APPLICATION.

Figure 2.7 A document image before and after down sampling

As another preprocessing step, the images are normalized. Normalization is an important step through which the range of input parameters (pixel intensity values) will be changed and

pixel values would have a similar data distribution. This will facilitate the network to converge faster and more smoothly.

2.3 MobileNet network architecture

In this thesis, we have adopted MobileNet CNN-based network as the base model of our approach to classify the large historical document image repository based on the presence of illustrations and diagrams. We adopted this network since it is a light and fast network and due to its different implementation of convolution operator, it has shown promising accuracy while being mindful of computation costs.

In 2017, Howard (2017), presented MobileNet architecture which was optimized for mobile and embedded vision applications. MobileNet is designed based on a smoothed architecture that builds lightweight deep neural networks leveraging depth-wise separable convolutions. The general trend in recent years has been to construct deeper and more complex networks to achieve better accuracy. However, these improvements in accuracy might not always imply that networks are more effective in terms of size and speed, especially in mobile applications. To address this issue, MobileNet has been designed with two hyper-parameters which enable us to design small and fast networks (Howard, et al., 2017). Figure 2.8 illustrates some mobile applications of MobileNet.

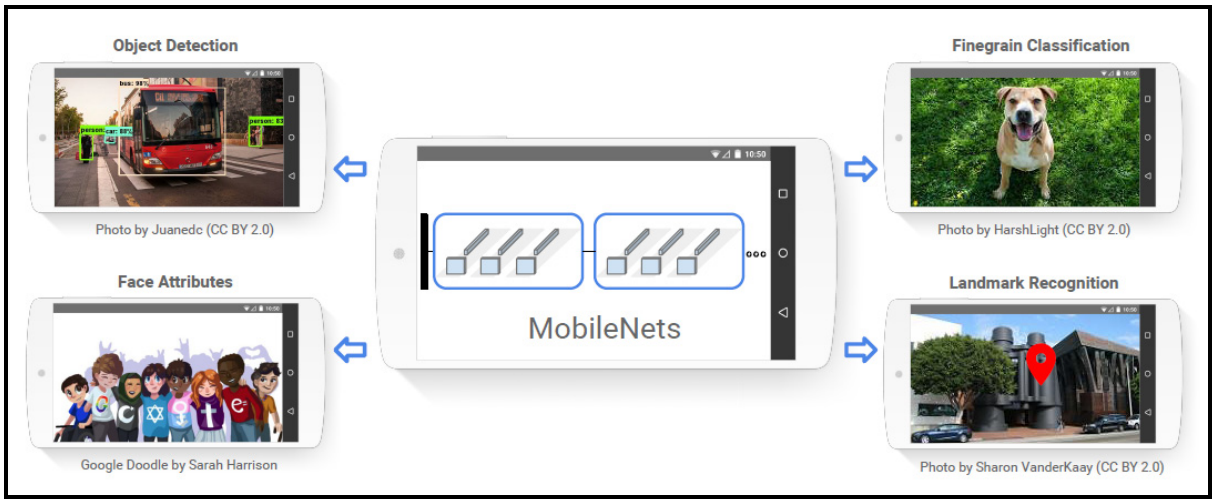


Figure 2.8 MobileNet models can be applied to various recognition tasks for efficient on-device intelligence

Taken from Howard, et al. (2017)

The prominent difference of MobileNet with other CNNs networks is its depth-wise separable convolutions in which the standard convolution has been substituted by two steps of a ‘depth-wise convolution’ and a ‘1*1 convolution’ called a point-wise convolution. These steps are presented in Figure 2.9.

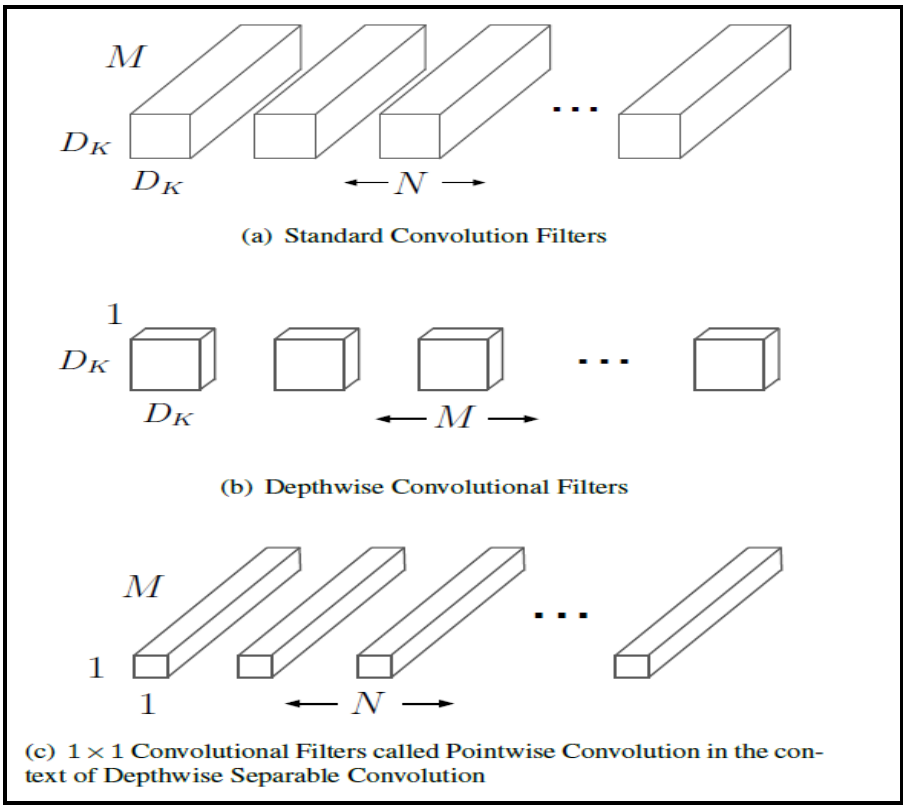


Figure 2.9 The standard convolutional filters in (a) are replaced by two layers: depth-wise convolution in (b) and point-wise convolution in (c) to build a depth-wise separable filter.
Taken from Howard et al. (2017)

In the first step, a filter is applied to each input channel and in the second step a 1*1 convolution would occur to combine the outputs of the previous step. By factorizing the convolution operation in this way, the computation complexity reduces considerably (Howard, et al., 2017). Considering M as the number of input channels (input depth), N as the number of output channels (output depth), D_K as the spatial width and height of the convolution filter and D_F as the spatial width and height of a square input feature map, standard convolutions have the computational cost which is illustrated in (2.1).

$$D_K \times D_K \times M \times N \times D_F \times D_F \quad (2.1)$$

Taken from Howard, et al. (2017)

On the other hand, with depth-wise separable convolutions in the MobileNet, the convolution operation is divided into two steps of 'depth-wise convolution' (Figure 2.9.b) and 'point-wise convolution' (Figure 2.9.c) and as a result the computation cost of convolution is as presented in (2.2) which is computationally less complex than standard convolution operation.

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \qquad (2.2)$$

Taken from Howard, et al. (2017)

The MobileNet architecture is illustrated in Figure 2.10.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 2.10 MobileNet architecture

Taken from Howard, et al. (2017)

2.4 Regularization

A critical challenge in machine learning is developing a model that works well on new unseen data as well as the train data and this indicates the generalization power of the model (Goodfellow, Bengio, & Courville, 2016). In other words, we are interested in the generalization capability of the model that could be evaluated by generalization error, also called test error.

In practice, the effort to fit the model precisely on the train data usually results in a complex model with a very low error rate on train data, which sometimes will result in a higher test error rate and this is referred to as over-fitting. It happens when there is a significant difference between train and test errors.

The performance of a machine learning algorithm can be evaluated by these two metrics:

- Ability to reach low training error,
- The small gap between training and test error.

These two aspects refer to the challenges of under-fitting and over-fitting in machine learning. If the model can't get an enough low error value on the training collection, it's called under-fitting. On the other hand, over-fitting occurs when there is a large gap between training and test error. (Goodfellow, Bengio, & Courville, 2016) Under-fitting and over-fitting can be controlled by the model complexity. The more the complexity, the higher risk of over-fitting. The complexity of a model refers to its learning capacity which is determined by the number of model's parameters. For the training set in Figure 2.11, as it can be seen on the left, a linear function cannot learn the data. In the center, it's been shown that a quadratic function is sufficient to present this data. On the right, a polynomial of degree 9 is fitted on the data which results in over-fitting.

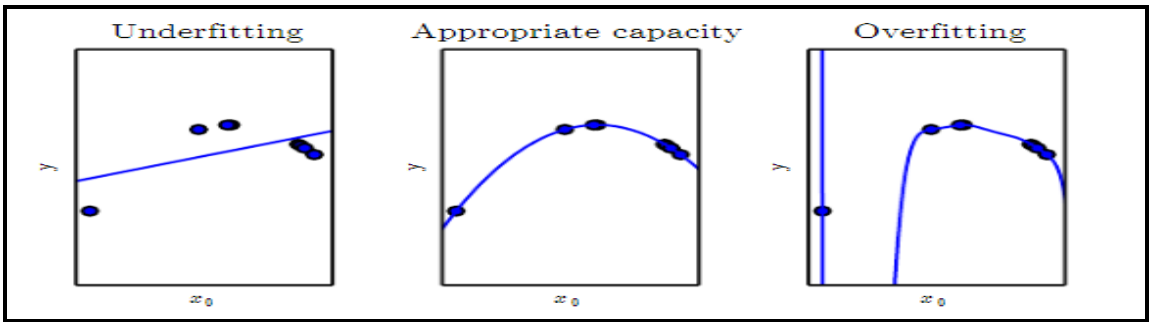


Figure 2.11 Under-fitting and over-fitting
Taken from Goodfellow et al. (2016)

In Figure 2.12, the typical relationship between model capacity and error rate is illustrated.

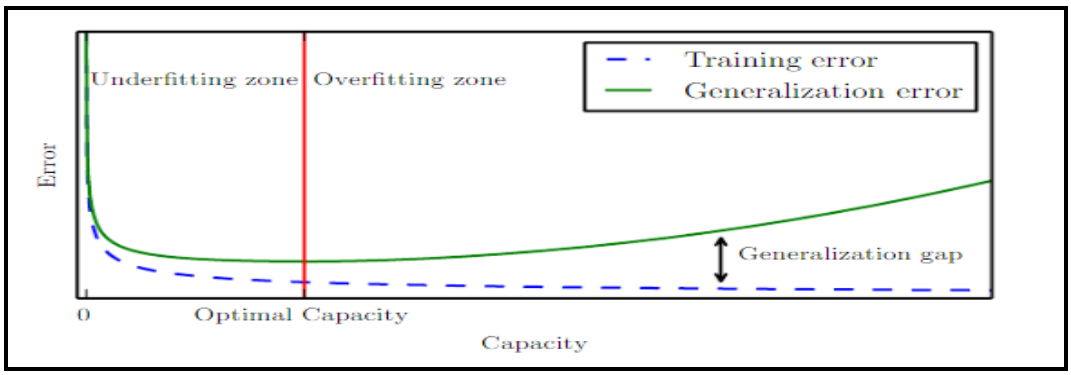


Figure 2.12 The relationship between model capacity and error rate
Taken from Goodfellow et al. (2016)

There are many strategies in machine learning to mitigate over-fitting. One of the efficient strategies is 'Regularization'. Regularization is a strategy for reducing the model's complexity.

When over-fitting occurs, the model is complex and usually, parameters of the model have large values. Regularization provides a way to control model complexity and thereby alleviate the over-fitting (Bishop, 2006). This is accomplished by penalizing the loss function. Two common regularization algorithms are 'L1 regularization' that is also called 'lasso' and 'L2 regularization'. L1 regularisation imposes an L1 penalty equal to the absolute

magnitude of the parameters' magnitude. In other words, it sets a restriction on the parameters' size. Models with L1 can be sparse. (i.e. models with few parameters); some less informative parameters can become zero and eliminated from the model, resulting sparse models.

L2 regularisation imposes a penalty equal to the square of the parameter magnitude. Models based on L2 would not be sparse and all parameters are shrunk by the same ratio (none are eliminated) (Hastie, Tibshirani , & Friedman , 2009). In L2, a penalty factor is added to the objective function to penalize the model's parameters to be large and set a restriction on their values. (Murphy, 2012)

The key difference between L1 and L2 regularisation is that L1 regularisation attempts to estimate the data's median, while L2 regularisation attempts to estimate the data's mean to prevent overfitting. In this work, we have applied the combination of L1 and L2 which is also called Elastic Regularization:

$$Regularized\ Loss\ Function = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + L1\ penalty + L2\ penalty \quad (2.3)$$

$$L1\ penalty = \lambda 1 \sum_{j=0}^{M} |W_j| \quad (2.4)$$

$$L2\ penalty = \lambda 2 \sum_{j=0}^{M} W_j^2 \quad (2.5)$$

The results indicate that training our model with this regularised loss function prevents our model from having a wide generalisation gap between training and test error, making it less prone to over-fitting.

2.5 Imbalanced dataset

An imbalanced classification problem arises when the distribution of classes is not equal in the training dataset and there is a small amount of samples representative for minority class and against there are a large number of samples from the majority class. A comparative example of balanced and imbalanced datasets is illustrated in Figure 2.13. Imbalanced datasets are common in real-world classification problems. This issue would affect the prediction performance of the trained model negatively.

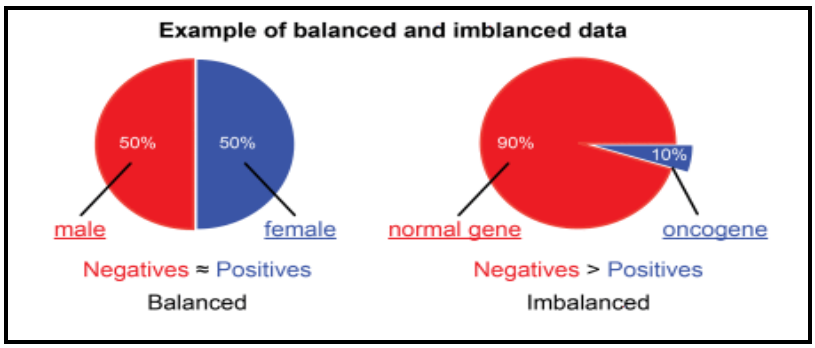


Figure 2.13 A comparative example of balanced and imbalanced datasets

Taken from Tripathi (2019)

Due to insufficient samples from minority classes, the model would have difficulties learning the minority class's specifications. If we consider minor class as the positive class and major class as the negative one, imbalanced training datasets lead to models with good specificity[3] but poor sensitivity[4]. (Kuhn & Johnson, 2013)

To tackle difficulties of the models in learning from imbalanced data, three main categories of approaches could be considered: (Krawczyk, 2016)

- Data-level techniques, which try to create balance in the training data samples, with over-sampling or under-sampling;
- Algorithm-level techniques, which make changes in the learning algorithms to lead the model to be more focused on the minority class;
- Hybrid techniques are equipped with the advantages of the first two approaches.

In this thesis, the training datasets are imbalanced. They are depicted in Figure 2.14. (More detailed information regarding our datasets specification is presented in chapter 3).

[3] Specificity is the rate of actual negatives, which predicted as negative (true negative)
[4]Sensitivity is the rate of actual positive cases which predicted as positive (true positive).

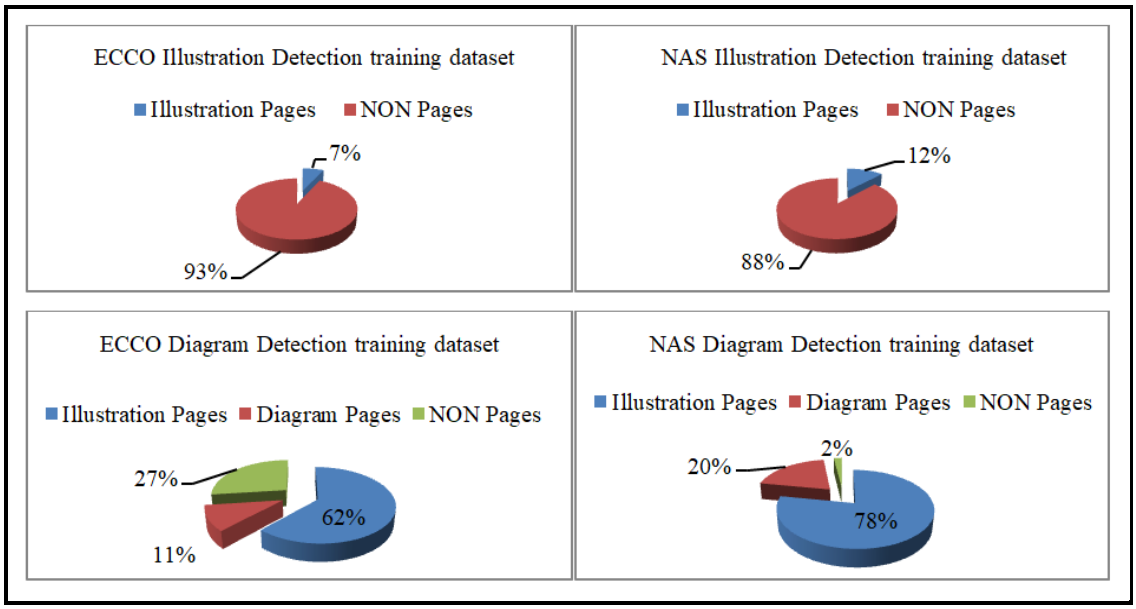


Figure 2.14 Imbalanced datasets of ECCO and NAS

We have applied an algorithm-level approach to alleviate the effect of imbalanced data. In this approach, class weights are calculated with the formula given in (2.6). As it can be seen, a class's weight is calculated in the inverse proportion of the number of samples representing that class.

$$class_j\ weight = \frac{total\ number\ of\ samples}{number\ of\ classes \times number\ of\ samples\ in\ class\ j} \tag{2.6}$$

Figure 2.15 shows an example of calculating class weights.

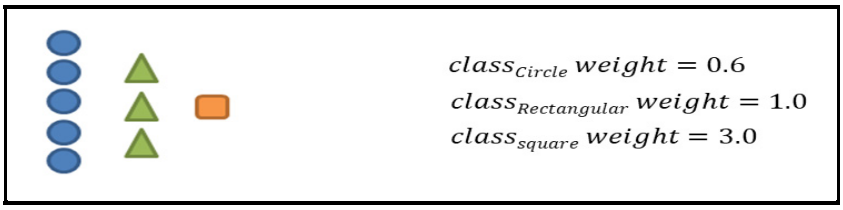


Figure 2.15 Calculating class weights in an unbalanced dataset

As it is depicted in Figure 2.15, the class that has the fewest number of samples (square class) has the highest class weight value. These class weights will be applied as coefficients for data samples in the loss function. This way, we make the model pay more attention to the classes that are under-sampled. The aim is to penalize the minority class more for misclassification by giving them a higher class weight. In other words, the model would be penalized more with the minor class's misclassified samples.

The majority class's cost function has a small weight, which leads to a lower error value and, as a result, fewer updates to the model parameters. For the minority class, a higher weight value is added to the cost function, resulting in an increased error calculation and, as a consequence, greater changes would be made in model parameters. This way, the model is biased to pay more attention to the minority class.

To evaluate a model which is applied to imbalanced data, accuracy would not be a reliable metric. When considering user interest in minority (positive) class samples, accuracy is not reasonable since the effect of the least represented, but more important, class is decreased when compared to the majority class (Branco, Torgo, & Ribeiro, 2015).

2.6 Transfer learning

Transfer learning creates the possibility of reusing the knowledge learned on a task in a different but related task to improve the learning process in the second task. The concept of transfer learning is depicted in Figure 2.16.

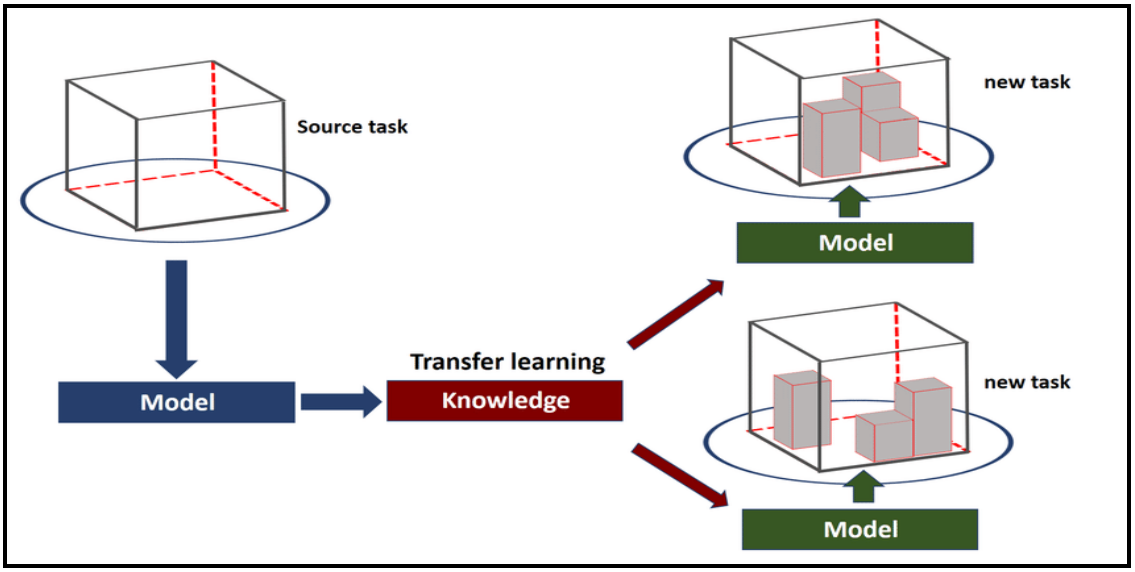


Figure 2.16 Transfer Learning
Taken from Bouhamed et al. (2020)

"Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones." (Sarkar, 2018). In recent years, deep learning methods have achieved considerable success in many complicated domains such as computer vision. On the other hand, compared with other machine learning algorithms, they require more amounts of data and time to train.

Nowadays, many deep learning-based pre-trained networks are already trained in computer vision domains. Most of these pre-trained networks have been shared by their developer to be used by others. We can use these networks to implement transfer learning in our domain of interest. (Sarkar, 2018)

In Figure 2.17, training error and loss are compared when transfer learning is implemented versus training the model from scratch. (Baykal, Dogan, Ercin, Ersoz, & Ekinci, 2019)

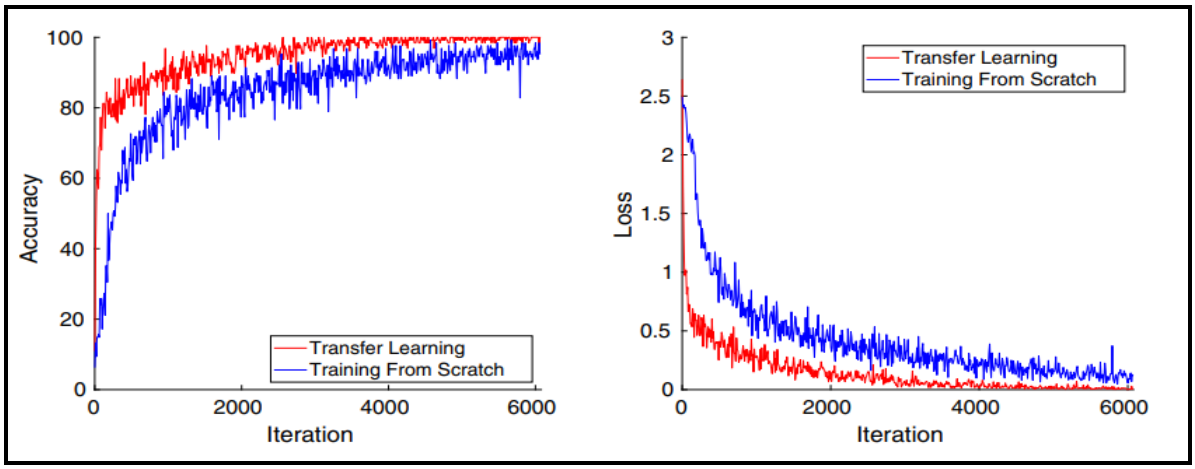


Figure 2.17 The effect of transfer learning in training accuracy and loss
Taken from Baykal et al. (2019)

In a classification task, whenever sufficient labelled data is not available to train the model from scratch, transfer learning would be significantly effective to facilitate the learning algorithm with rapid learning and higher performance.

Since deep learning algorithms require an enormous amount of data to train and on the other hand in most real-world problems, accessing large labelled data is not possible except with high cost, transfer learning has attracted more attention in CNN-based networks.

There are two common strategies to facilitate deep learning algorithms with transfer learning (Sarkar, 2018) :

- Off-the-shelf Pre-trained Models as Feature Extractors:

  Deep learning models leverage layered architecture to learn different features through different layers in a hierarchical way. There is another final layer that yields to the final output that in the classification problems is the label of the target classes. In off-the-shelf pre-trained models (Figure 2.18), the pre-trained model without the final layer is used as a fixed feature extractor for other tasks.

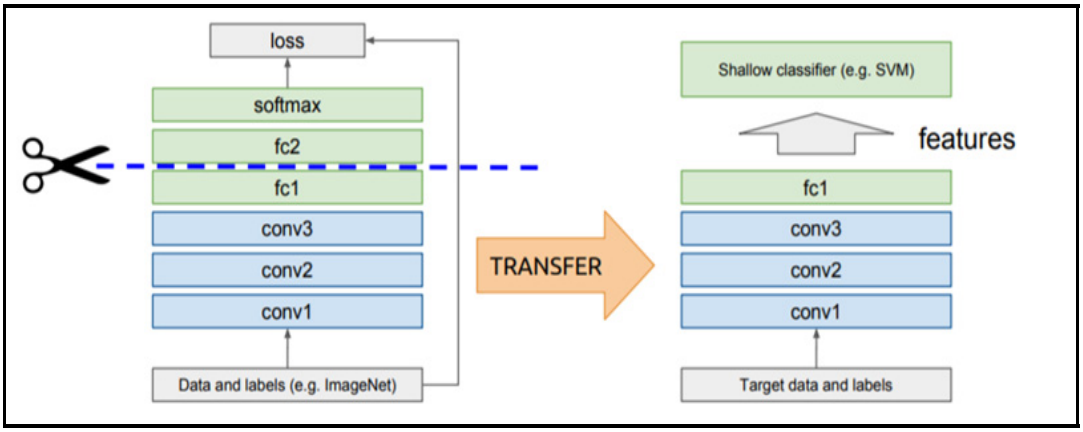


Figure 2.18 Off-the-shelf Pre-trained Models as Feature Extractors,
Taken from Sarkar (2018)

Razavian et al. (2014) investigated the results of using different feature extraction methods in the performance of 'object image classification', 'scene recognition, 'fine-grained recognition', 'attribute detection' and 'image retrieval' tasks. Figure 2.19 shows the results.

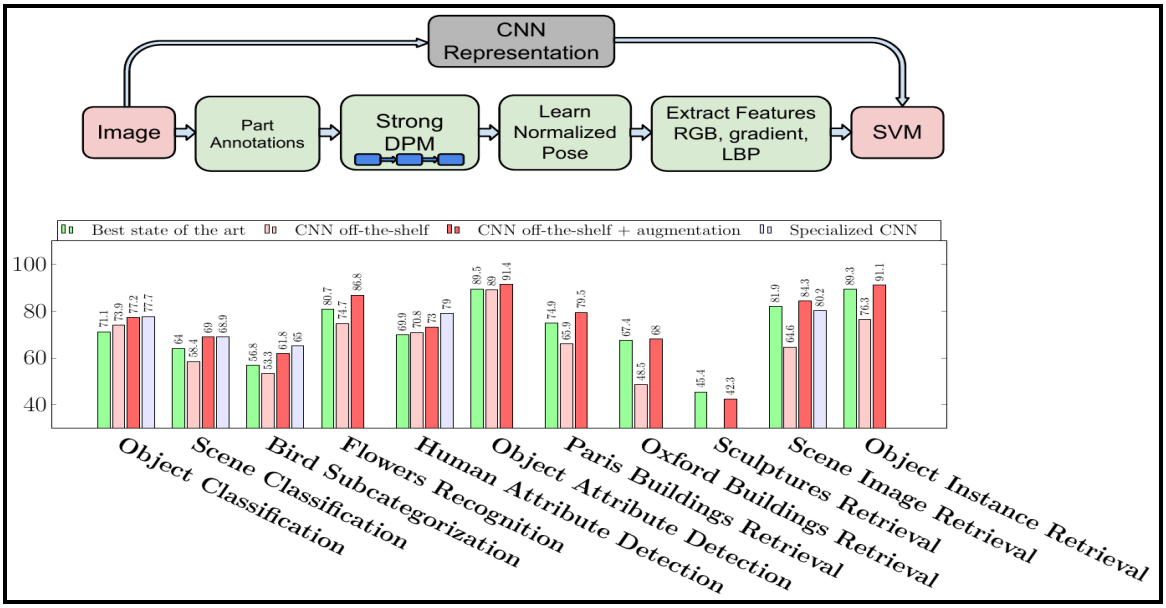


Figure 2.19 Off-the-shelf Pre-trained Models performance
Taken from Razavian et al. (2014)

According to the above results, the features from the pre-trained models significantly outperformed specialized task-focused deep learning models, as shown by the red and pink bars in the above figure.

- Fine Tuning Off-the-shelf Pre-trained Models: This strategy is more common in computer vision problems. In this type of transfer learning, in addition to replace the last layer of classification, some previous layers are also retrained on the target task.

Li and Plataniotis (2020) have used the concept of transfer learning in deep learning-base pathology image classification. They aimed to investigate to what extent transferring knowledge from a nature image classification problem to a pathology classification problem would be beneficial. Their experiment suggested the parameters learned in the nature images are also reusable in the pathology images.

In this thesis, we leverage the knowledge learned on the image classification problem on ImageNet dataset to empower our model with transfer learning. The strategy employed in our work to implement transfer learning is 'Fine Tuning Off-the-shelf Pre-trained Models'. Following this strategy to implement transfer learning, allows us to include knowledge and parameters learned on the source task into our model, as well as adjust some model parameters on our target dataset to focus on extracting more relevant features.

As mentioned earlier, we have used the MobileNet network as the base model to classify historical document images. We leveraged the parameters learned in a pre-trained MobileNet network which was trained on ImageNet, a large dataset containing over 14 million images from over 10,000 categories, to initialize our model's parameters.

The first layers in the CNN capture generic features, while the last ones focus more on sophisticated ones, specific to the task. Since our target datasets of historical document images have different structures in comparison with natural images datasets on which the base model had been pre-trained, we configured the last 18 layers of the MobileNet to be tuned on our datasets. These layers will concentrate on extracting features relevant to our target dataset.

2.7 Data Augmentation

Deep learning models significantly rely on the availability of enough data to present the best performance. Small datasets have the disadvantage that models trained on them do not generalize well to data from the validation and test sets. As a result, over-fitting is an issue with these models. (Wang & Perez, 2017).

Data Augmentation is a data-space approach to the issue of data scarcity. It enhances the size and diversity of the datasets by creating new copies of the existing data with random transformations. In other words, augmentation is a technique applied to training datasets to expose the model to more and more data during the training step to alleviate over-fitting.

Similar to most real-world problems, in the domain of this project we did not have access to a large amount of labelled data to fulfill the requirement of the model to yield promising results. We leveraged augmentation techniques to conquer the scarcity of the training data. In our implementation of augmentation, during the training process of the model, in each epoch, the model is applied on new randomly augmented images from the original dataset. Therefore, at the end of the training process, the model has been exposed to the total number of samples depicted in equation 2.7.

$$\textit{Number of training epochs} \times \textit{Size of the training dataset} \tag{2.7}$$

This increase in the number of samples to which the model is exposed, would significantly add to the generalization power of the model. We have used the following augmentation techniques.

- Mean subtraction (Zero centrings)

  To make data zero-centred, the mean of the data will be subtracted from each of the data points (Figure 2.20). Consider a situation in which a neuron's (unit) inputs are all positive or all negative. The gradient measured during back propagation would be either positive or negative in this scenario (the same as the sign of inputs). As a result, parameter changes are limited to a few unique paths, making convergence inefficient. Zero centrings will conquer this issue (Parmar, 2018);

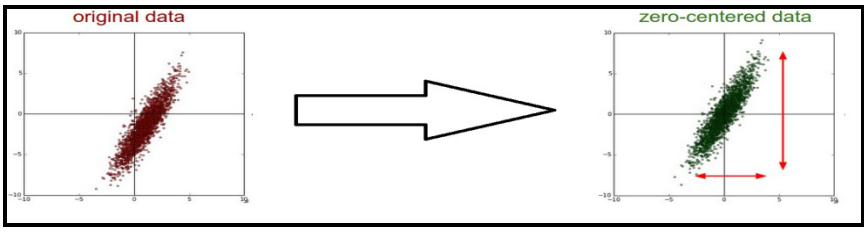


Figure 2.20 Mean subtraction

- Rotation

  The image is rotated right or left on an axis between 1° and 359° for rotation augmentations. The rotation degree parameter has a significant impact on the safety of rotation augmentations. Slight rotations, such as 1 to 20 or 1 to 20, have yielded positive outcomes. (Shorten & Khoshgoftaar, 2019) In this work, we have used the same degree range for rotation augmentation;

- Width and Height shifting

  To prevent positional bias in data, shifting images left, right, up, or down can be a highly effective transformation. (Shorten & Khoshgoftaar, 2019) We have used width and height shifting in which images will be translated vertically or horizontally at random;

- Horizontal flipping

  Horizontal flipping is more frequently used than vertical flipping. It's used to turn half of the pictures horizontally at random (Figure 2.21).

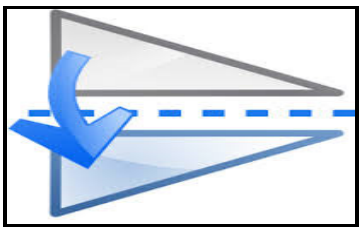


Figure 2.21 Horizontal flip

This is one of the simplest augmentations to use, and it worked well on image datasets like CIFAR-10 and ImageNet. (Shorten & Khoshgoftaar, 2019)

2.8 Specifications of the proposed approach

The concepts and techniques implemented in our proposed model are described in detail in the previous sections. In this section, we present an review of our model specifications. Figure 2.22, shows the architecture of our proposed model.

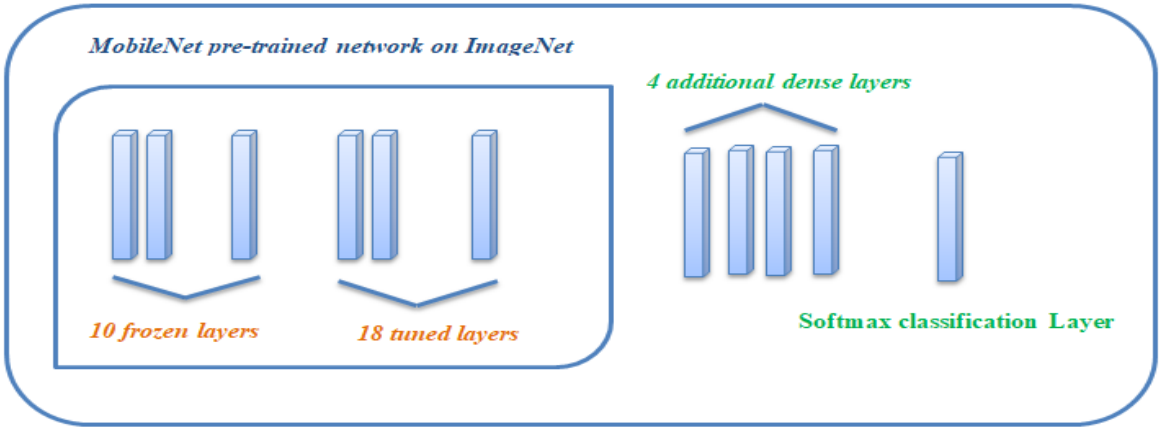


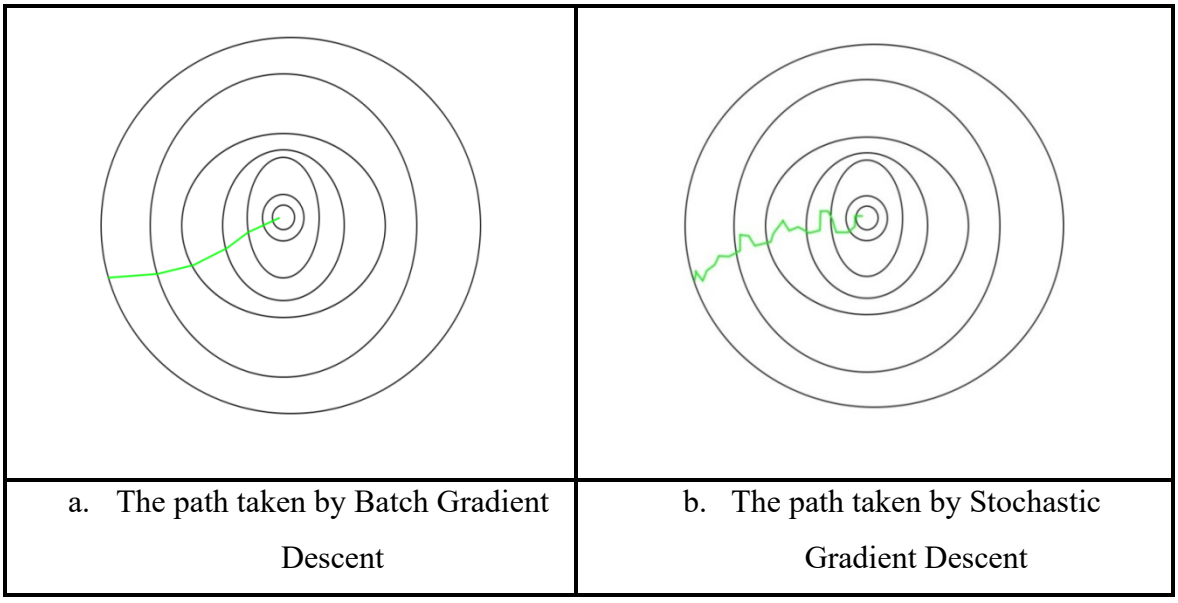
Figure 2.22 The overall architecture of our proposed model

As it is depicted in Figure 2.22 and also as described in previous sections, we have used the MobileNet (Howard, et al., 2017), 28 layers CNN based network, as our base model. To leverage the transfer learning, the base model we have used is pre-trained on ImageNet (Stanford Vision Lab, 2020), a well-known large image classification dataset. We configured the first 10 layers of the base model to be fixed and dedicated these layers to detect the lower level image features. The last 18 layers, on the other hand, are tuned during the training to be more specifically focused on the learning features related to our target datasets. We also added four additional dense layers on the top of the based model to increase the learning capacity of our model. Finally, the last layer of our network which is the output layer is equipped with a Softmax activation function.

To conquer overfitting, our model is equipped with L1 and L2 regularization techniques.

We used SGD (Stochastic Gradient Descent) algorithm as optimizer in our model. It is a variation of gradient descend algorithm that is largely used in cases in which there are large amounts of data. SGD works remarkably well on tasks with large datasets (Murphy, 2012). In each iteration of SGD algorithm, a few random samples from the dataset are picked to consider in the calculations of updating model parameters which result in less time complexity in comparison with the common gradient descent algorithm that go through all the dataset samples in each iteration (Brownlee, 2016).

In SGD, in each iteration, few random samples are considered, so the path to the minimum cost would not be as smooth as typical gradient descend algorithms and there will be some noises. However, it is worthwhile when the minimum cost can be meet within significantly less training time (Roy, 2020). In Figure 2.23, the path to minimum cost is compared in batch gradient descend and stochastic gradient descend. Besides higher speed, due to adding some noise, SGD is less likely to get stuck in local minima (Murphy, 2012).



a. The path taken by Batch Gradient Descent	b. The path taken by Stochastic Gradient Descent

Figure 2.23 Comparison of the path to minimum cost in gradient descend algorithms Adapted from Roy (2020)

We have selected the learning rate of '0.01' to train our model. The learning speed of the network is regulated by the learning rate (Brownlee, 2019). Since the learning rate determines the step size with which the parameters of the model are updated, specific considerations should be taken into account in setting its value. Too large values would result in a higher speed of learning with the cost of probable stuck in local minima. Small learning rates, on the other hand, make the learning process very slow and demands more computational resources and time. We have tried learning our model with different values of learning rate and found the value of '0,01' best fitted to our task.

As mentioned earlier, to alleviate the scarcity of available labelled training data, we empowered our model with augmentation techniques. This way, we exposed our model to more data and also increased the generalization power of the model.

Our two main tasks as 'Illustration Detection' and 'Diagram Detection' on ECCO and NAS datasets are depicted in Figure 2.24.

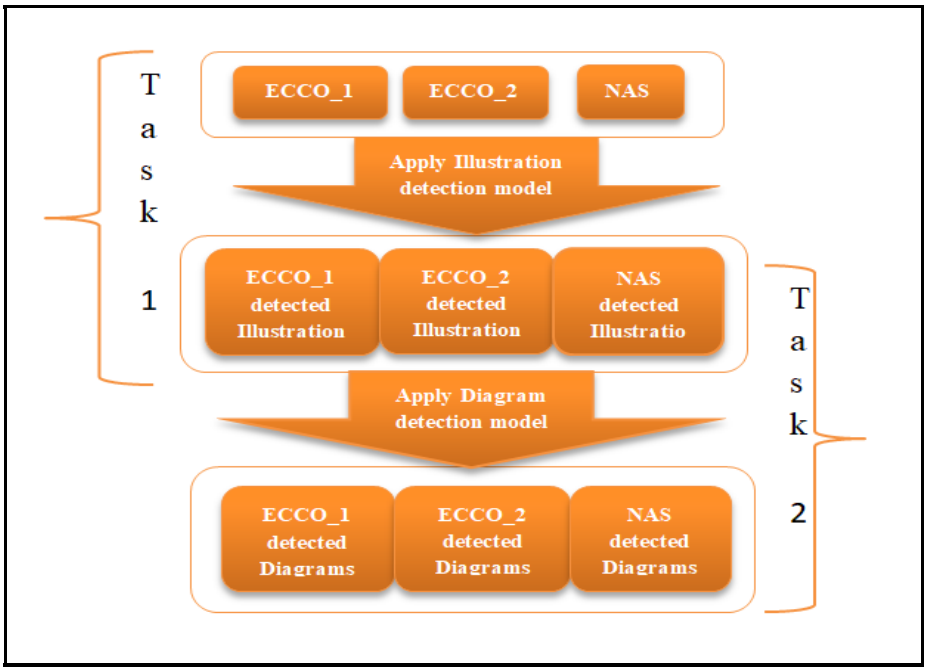


Figure 2.24 Two main tasks of this thesis

In the next chapter, the results of applying our proposed model on two large datasets of historical document images, ECCO and NAS, are presented and discussed.

CHAPTER 3

EXPERIMENTS AND RESULTS

3.1 Evaluation metrics

Results of classification problems can be divided into four categories of 'true positive', 'true negative', 'false positive' and 'false negative' (Figure 3.1). These parameters are usually integrated into a matrix called 'Confusion Matrix' whose purpose is to present the performance of a classification model.

		Predicted Values	
		Positive	Negative
Actual Values	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 3.1 Confusion Matrix

To evaluate the classification model from different perspectives, there are four common metrics of 'accuracy', 'precision', 'recall' and 'F1-score'.
The number of correct predictions divided by the total number of samples equals Accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

The ratio of correctly predicted positive samples to total predicted positive samples is known as Precision.

$$Precision = \frac{TP}{TP+FP} \tag{3.2}$$

Recall is the proportion of positive samples that were correctly predicted to the total number of samples that should have been predicted as positive.

$$Recall = \frac{TP}{TP+FN} \tag{3.3}$$

F1-score is determined by taking the Harmonic Mean between precision and recall. We can determine both how precise our classifier is (how many instances it classifies correctly) and how reliable it is (does it miss a significant number of instances).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

In this thesis, we are more focused on Precision and Recall to evaluate our model. Precision refers to the percentage of relevant results which are results that we are looking for. Alternatively, recall is the percentage of relevant results classified correctly by the algorithm. We haven't considered accuracy for our domain because our domain includes an imbalanced dataset. Accuracy is not a proper indicator in the context of imbalanced data-sets, since it fails to distinguish between the numbers of correctly classified examples of different classes (Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012).

3.2 Dataset Specification

In this thesis, our target datasets are ECCO[5] and NAS datasets which are used for the Visibility of Knowledge (VOK Project, 2016) project. ECCO aims to preserve historical documents printed from 1701 until 1800 in Britain. The manuscript collection consists of more than 32 million pages from over 155,000 manuscripts. The manuscripts are categorized into 7 subjects of "Religion and Philosophy" (RelAndPhil), "Literature and Language" (LitAndLang1, LitAndLang2), "History and Geography" (HistAndGeo), "Social Science and

[5] Eighteenth Century Collections Online

Fine Arts" (SSAndFineArt), "Medicine, Science, and Technology" (MedSciTech), "Law" and "Reference" (GenRef).

This repository of historical document images is divided into two parts:

- ECCO_1: consists of around 150,000 manuscripts and more than 26,000,000 pages;
- ECCO_2: consists of more than 52,000 manuscripts and around 7,000,000 pages.

Figure 3.2 illustrates the distribution of the pages through years and subjects in ECCO_1 and ECCO_2.

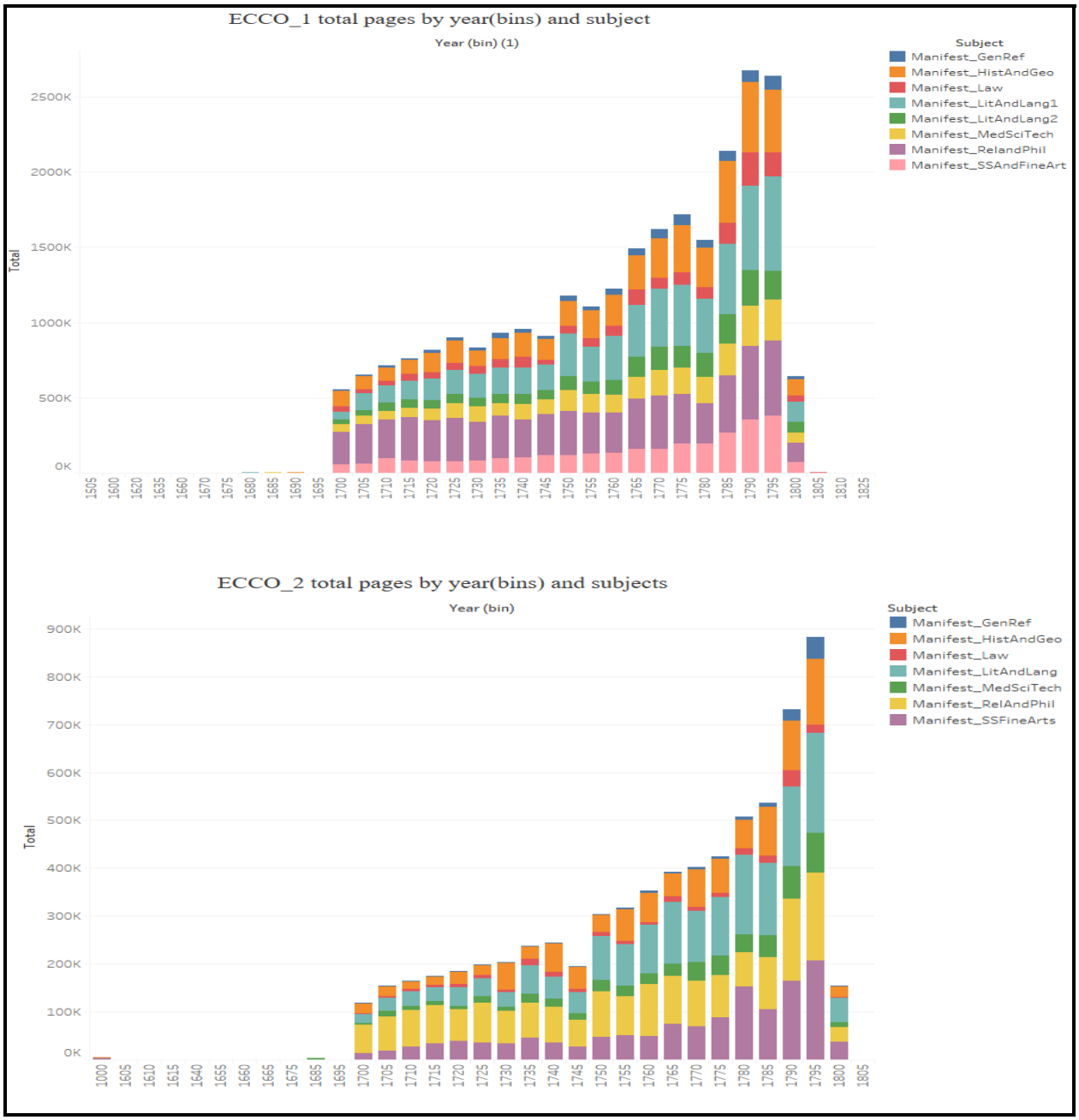


Figure 3.2 Distribution of ECCO document pages over year and subject

In this thesis, we also used another dataset of historical documents called NAS. The manuscripts in this dataset are categorized based on the language in which they were published into five groups: 'DE_KoenigAkamie_Berlin', 'FR_JournalScavans_Paris', 'RU_AcademieImperiale_Petersburg', SE_KonglSwenska_Stockholm' and

'UK_RoyalSociety_London'. NAS consists of 827 journals and 530,544 images.

Figure 3.3, illustrates the distribution of the pages through years and subjects in the NAS dataset.

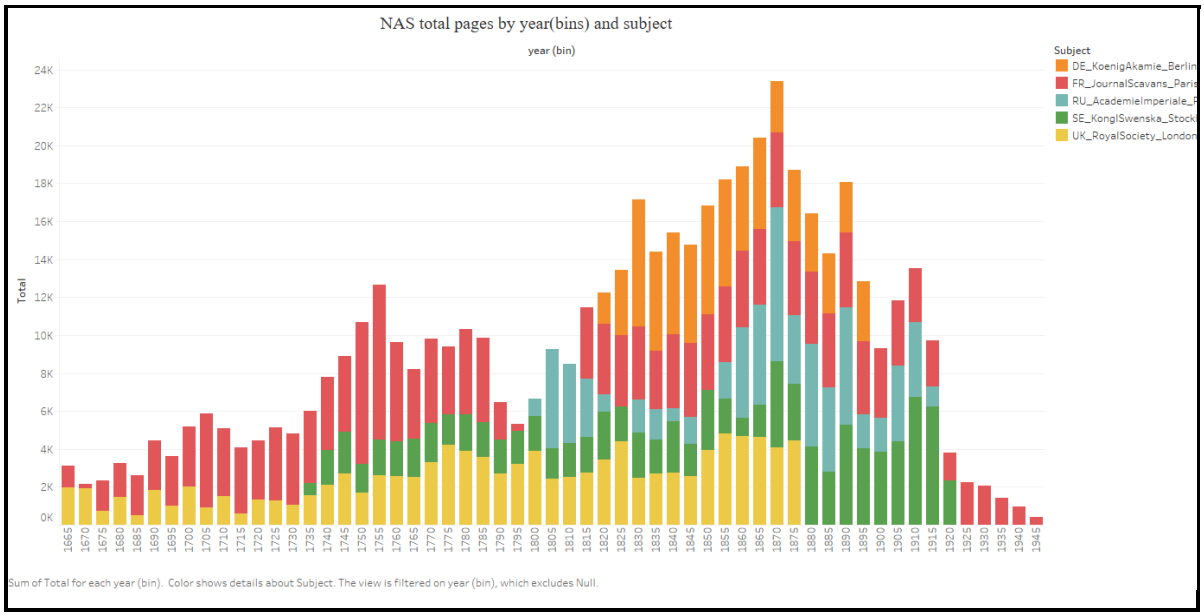


Figure 3.3 Distribution of NAS document pages over year and subject

3.3 Illustration detection results

As mentioned in section 1.2, in this work, in the first step we have applied our proposed model on the ECCO and NAS dataset to categorize the document images into two categories of 'pages with illustrations' and 'pages without illustrations'. The results and discussion regarding these two-class classification tasks are presented in the following sections.

3.3.1 Illustration detection training dataset specification

The specification of the datasets used for training and evaluating our proposed model for Illustration detection on ECCO and NAS datasets are presented in Table 3.1 and Table 3.2.

Table 3.1 ECCO Illustration detection labelled dataset specification

class	**Number of labelled document images**	
	Training set	Testing set
Illustration	1,343	133
NON (Table, Footnote, Text)	18,095	9,852

As illustrated in the above table, in the ECCO dataset, we have access to a total of 29,000 labelled document images which are categorized into two classes of 'Illustrations' and 'NON'. NON category consists of pages containing Tables, Footnotes or pure texts. About 10,000 of these images have been used for final evaluating the model and the rest have been used for training the model. Moreover, out of the training set, 20% is used as a validation set to evaluate the model during the training and per each epoch.

Table 3.2 NAS Illustration detection labelled dataset specification

class	**Number of labelled document images**	
Class	Training set	Testing set
Illustration	2,421	328
NON (Table, Footnote, Text)	16,977	8,853

In the NAS dataset, there are 28,000 labelled document images, from which 9000 images have been used for the final test process of the model and the remainder for training. The same protocol as the ECCO dataset has been respected regarding the evaluation set.

3.3.2 Results of Illustration Detection

Our proposed model, introduced in detail in Chapter 3, has been applied to the ECCO and NAS datasets. The results are presented in the next two sections.

3.3.2.1 ECCO dataset Illustration detection results

Our proposed model is trained through 50 epochs on the ECCO Illustration dataset. The loss diagram through these 50 epochs is illustrated in Figure 3.4.

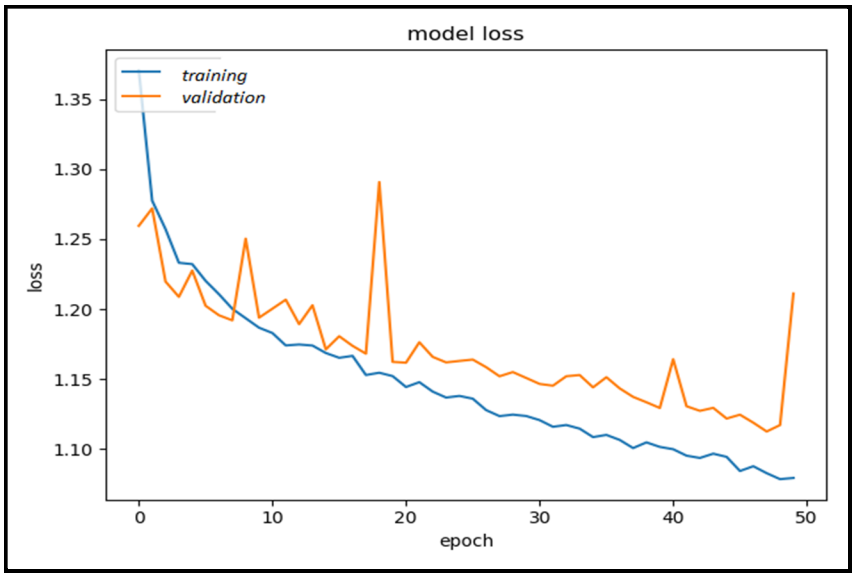


Figure 3.4 Loss diagram of ECCO Illustration detection task

As we can see in Figure 3.4, the general trend of the loss diagram both in the train and validation set is descending and this is what we expected. There are some fluctuations in the validation loss diagram. It can occur when in one batch we have mislabelled samples which can lead to an update that does not reduce, but rather increases the global loss.

In Table 3.3 and Table 3.4, the results of the ECCO illustration detection regarding the evaluation metrics of precision, recall and F1-score are reported. Moreover, Table 3.4 also shows the results in detail for each class.

Table 3.3 ECCO Illustration detection evaluation metrics

	precision	**recall**	**F1-score**	**support(number of images in the validation set)**
Macro average	0.97	0.97	0.97	9998

Table 3.4 ECCO Illustration detection evaluation metrics per class

	precision	**recall**	**F1-score**	**Support(number of images in the validation set)**
Illustration	0.94	0.95	0.94	146
NON	0.99	0.99	0.99	9852

As depicted in Table 3.4, the precision in detecting the illustration class is 0.94 in percent which means out of the total number of illustration predicted images, 94 percent of them are correctly predicted. Regarding recall, the illustration class recall ratio is 0.95 that shows from the total number of images which should be classified as illustrations, 95 percent of them are correctly detected as illustrations.

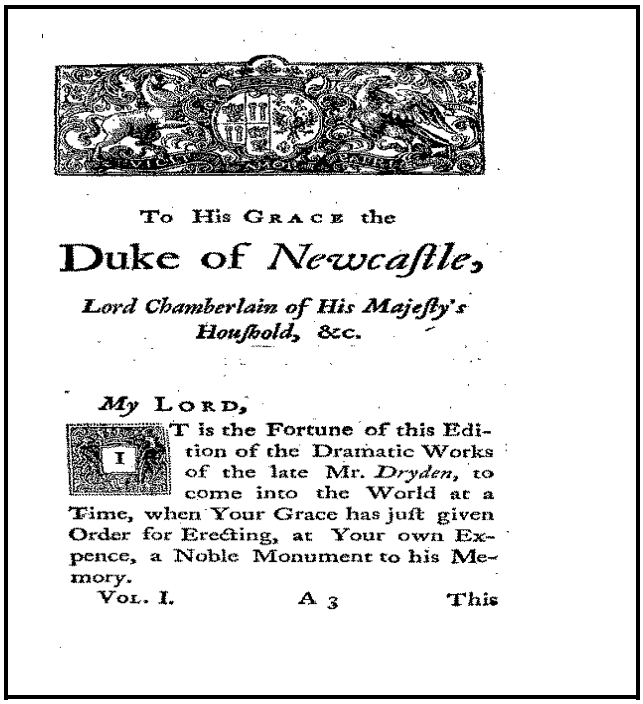
To be more specific in results, the confusion matrix of ECCO Illustration Detection is presented in Table 3.5.

Table 3.5 ECCO Illustration detection Confusion Matrix

		Predicted	
		ILLUS	NON
Ground Truth	ILLUS	138	8
	NON	9	9843

As illustrated in the confusion matrix, 8 out of 146 images from the illustration category in the test set are mislabelled as the NON category by the model. In the NON category, 6 images out of 9843, in other words, only 0.06% of the data are misclassified as ILLUS. As expected, due to enough training images in the NON category, the model is quite effective in detecting this category.

Most of the False Negative samples have a structure similar to the illustration in Figure 3.5.



To His GRACE the

Duke of *Newcaſtle*,

Lord Chamberlain of His Majeſty's Houſhold, &c.

My LORD,

IT is the Fortune of this Edition of the Dramatic Works of the late Mr. *Dryden*, to come into the World at a Time, when Your Grace has juſt given Order for Erecting, at Your own Expence, a Noble Monument to his Memory.

VOL. I. A 3 This

Figure 3.5 A False Negative sample in the Illustration detection in ECCO

According to the ground truth images, we expected the image in Figure 3.5 to be predicted as illustration but the model predicted it as NON. When investigating the labelled images in the NON category, we found that there are also some images with the same structure as the image in Figure 3.5 which are wrongly labelled as NON in the ground truth. This made the model confused to predict the correct label of this kind of image. The image in Figure 3.6 is in the False Positive samples. This image is labelled as NON in ground truth but it seems that it was mislabelled and it should have been labelled as illustration. However, the model was predicted as an illustration based on the patterns that had been learned through the training step.

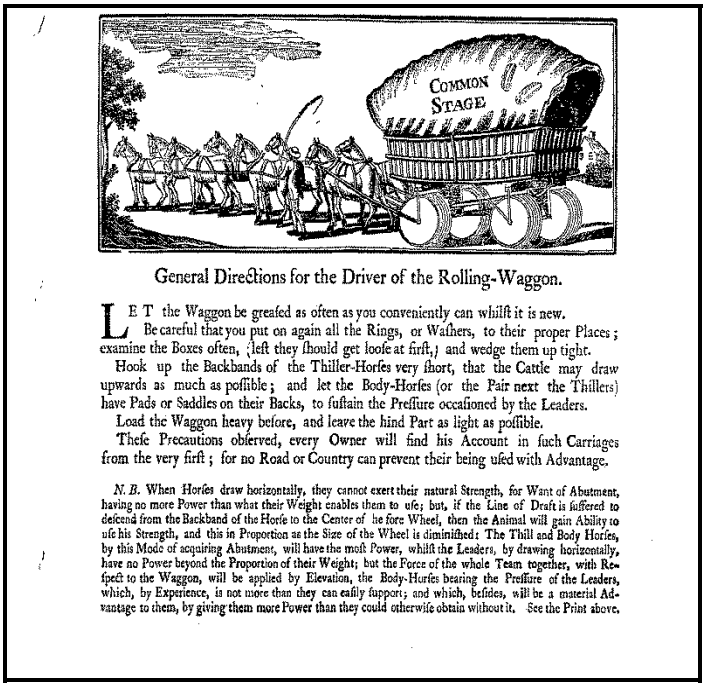


Figure 3.6 A False Positive sample in the Illustration Detection in ECCO

3.3.2.2 NAS dataset Illustration detection results

The results of experimenting with the illustration detection task on the NAS dataset are given in this section. Figure 3.7 shows the loss curve of the illustration detection model during the training of the model on the NAS dataset.

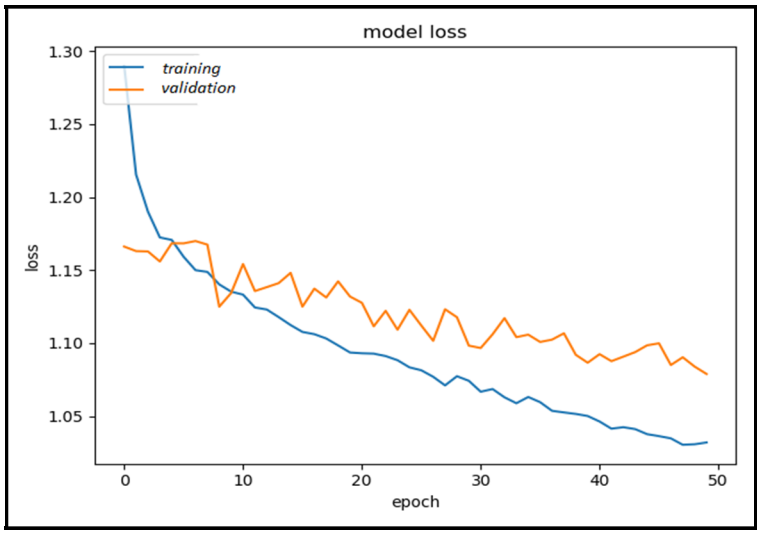


Figure 3.7 Loss diagram of NAS Illustration detection

The same as the ECCO dataset, the overall trend of the loss diagram is descending and it yields that the training process is reasonable. Table 3.6, Table 3.7 and Table 3.8, demonstrate the results of the NAS illustration detection from different perspectives.

Table 3.6 NAS Illustration detection evaluation metrics

	precision	**recall**	**F1-score**	**Support (number of images in the validation set)**
Macro average	0.89	0.96	0.92	9179

Table 3.7 NAS Illustration detection evaluation metrics per class

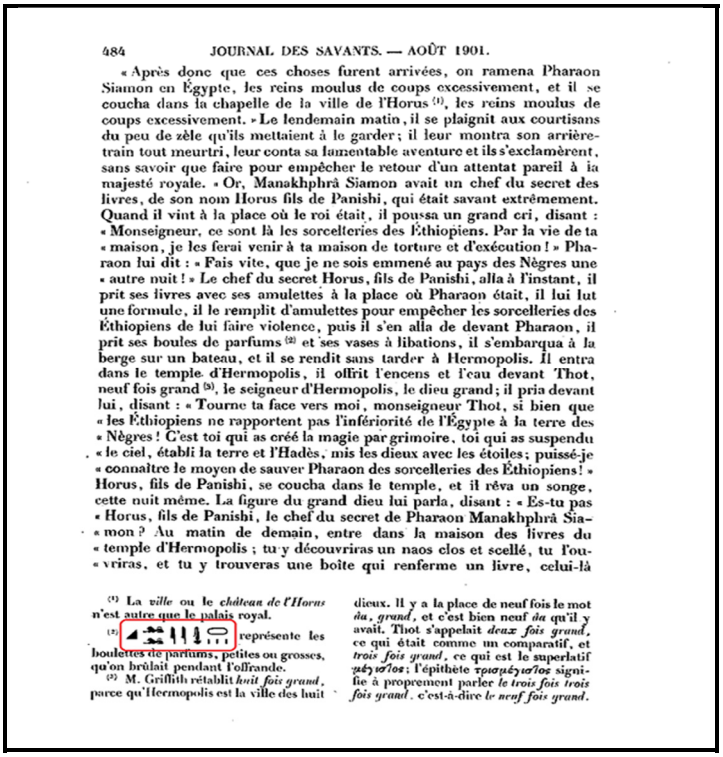
	precision	**recall**	**F1-score**	**Support (number of images in the validation set)**
Illustration	0.77	0.93	0.84	327
NON	0.99	0.99	0.99	8852

As it is illustrated in the tables, in the NAS dataset, illustrations are detected with 0.77 percent of precision and 0.93 percent of recall. Although the precision is less in comparison with the ECCO dataset, the reported recall rate is promising. It indicates that 93 percent of the actual illustrations are detected correctly.

Table 3.8 NAS Illustration detection Confusion Matrix

		Predicted	
		ILLUS	NON
Ground Truth	ILLUS	303	24
	NON	88	8764

According to the confusion matrix, in our class of interest, Illustration, 24 images were misclassified as NON. On the other hand, 88 images from the NON category were incorrectly indicated as illustrations by the model. In Figure 3.8, an instance of the False Negative images is illustrated. As highlighted in the image, there is a very small illustration in the image surrounded by text and it seems that the model found it difficult to predict it as an illustration due to the small size of the illustration in comparison to the text surrounding it.



484 JOURNAL DES SAVANTS. — AOÛT 1901.

« Après donc que ces choses furent arrivées, on ramena Pharaon Siamon en Égypte, les reins moulus de coups excessivement, et il se coucha dans la chapelle de la ville de l'Horus [1], les reins moulus de coups excessivement. » Le lendemain matin, il se plaignit aux courtisans du peu de zèle qu'ils mettaient à le garder; il leur montra son arrière-train tout meurtri, leur conta sa lamentable aventure et ils s'exclamèrent, sans savoir que faire pour empêcher le retour d'un attentat pareil à la majesté royale. « Or, Manakhphrâ Siamon avait un chef du secret des livres, de son nom Horus fils de Panishi, qui était savant extrêmement. Quand il vint à la place où le roi était, il poussa un grand cri, disant : « Monseigneur, ce sont là les sorcelleries des Éthiopiens. Par la vie de ta « maison, je les ferai venir à ta maison de torture et d'exécution ! » Pharaon lui dit : « Fais vite, que je ne sois emmené au pays des Nègres une « autre nuit ! » Le chef du secret Horus, fils de Panishi, alla à l'instant, il prit ses livres avec ses amulettes à la place où Pharaon était, il lui lut une formule, il le remplit d'amulettes pour empêcher les sorcelleries des Éthiopiens de lui faire violence, puis il s'en alla de devant Pharaon, il prit ses boules de parfums [2] et ses vases à libations, il s'embarqua à la berge sur un bateau, et il se rendit sans tarder à Hermopolis. Il entra dans le temple d'Hermopolis, il offrit l'encens et l'eau devant Thot, neuf fois grand [3], le seigneur d'Hermopolis, le dieu grand; il pria devant lui, disant : « Tourne ta face vers moi, monseigneur Thot, si bien que « les Éthiopiens ne rapportent pas l'infériorité de l'Égypte à la terre des « Nègres ! C'est toi qui as créé la magie par grimoire, toi qui as suspendu « le ciel, établi la terre et l'Hadès, mis les dieux avec les étoiles; puissé-je « connaître le moyen de sauver Pharaon des sorcelleries des Éthiopiens ! » Horus, fils de Panishi, se coucha dans le temple, et il rêva un songe, cette nuit même. La figure du grand dieu lui parla, disant : « Es-tu pas « Horus, fils de Panishi, le chef du secret de Pharaon Manakhphrâ Sia-« mon ? Au matin de demain, entre dans la maison des livres du « temple d'Hermopolis ; tu y découvriras un naos clos et scellé, tu l'ou-« vriras, et tu y trouveras une boîte qui renferme un livre, celui-là

(1) La *ville* ou le *château de l'Horus* n'est autre que le palais royal.

(2) [hieroglyphs] représente les boulettes de parfums, petites ou grosses, qu'on brûlait pendant l'offrande.

(3) M. Griffith rétablit *huit fois grand*, parce qu'Hermopolis est la ville des huit dieux. Il y a la place de neuf fois le mot *âa, grand*, et c'est bien neuf *âa* qu'il y avait. Thot s'appelait *deux fois grand*, ce qui était comme un comparatif, et *trois fois grand*, ce qui est le superlatif μέγιστος; l'épithète τρισμέγιστος signifie à proprement parler *le trois fois trois fois grand*, c'est-à-dire *le neuf fois grand*.

Figure 3.8 A False Negative sample in the Illustration Detection in NAS

Figure 3.9 shows some samples of False Positives in Illustration detection on the NAS dataset.

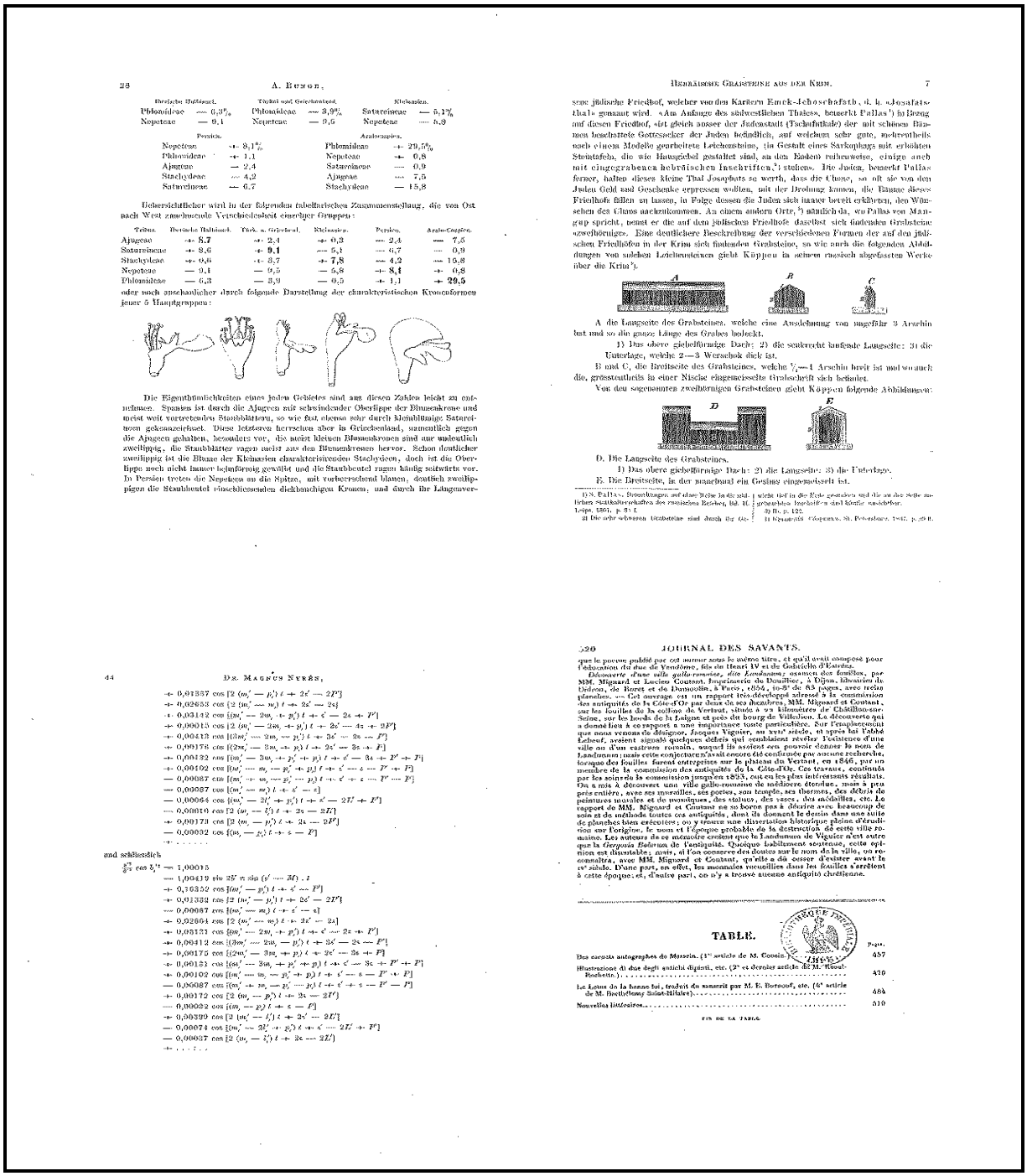


Figure 3.9 Samples of False Positives in Illustration detection on NAS dataset

In Figure 3.9, we can see that the above images contain some small illustrations and are detected as illustrations by the model but at the same time they had been labelled as NON in the ground truth and as a result, they are categorized as False Positives. On the other hand, in Figure 3.9 at the bottom, we can see images of pages that do not contain just pure text; they include Formula and Stamp that leads model to categorize them as illustrations.

3.3.3 Experiment Results of Illustration Detection on large unlabeled historical datasets of ECCO and NAS

As mentioned earlier, in this thesis, our main goal is the classification of document images based on the presence of illustration and diagram on two large datasets of ECCO and NAS which consisting of historical document images from ancient manuscripts. Figure 3.10 shows the overall perspective regarding our aim.

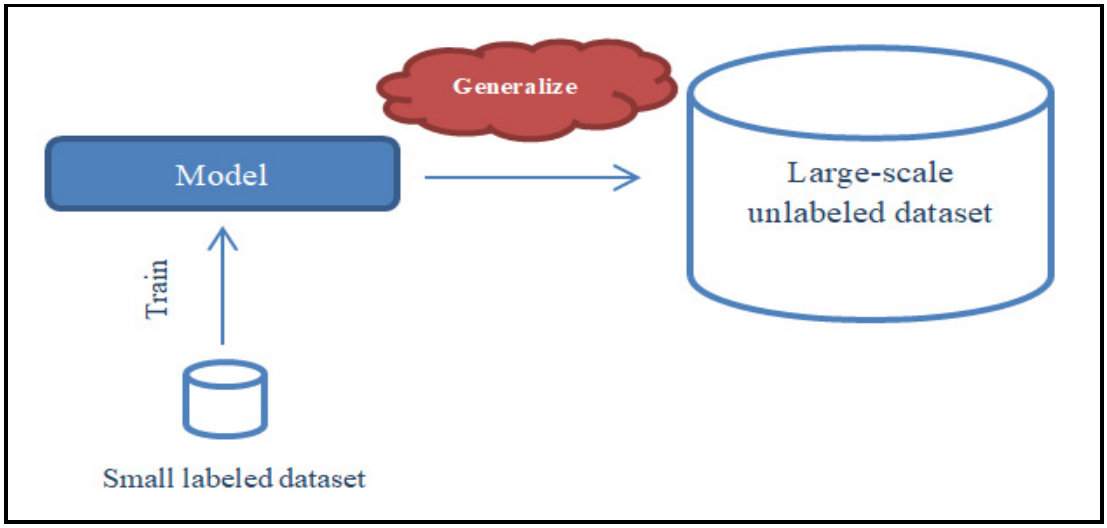


Figure 3.10 Detecting illustrations and diagrams on large-scale unlabeled datasets

After designing, training and evaluating our proposed model on the training and testing datasets which are described in detail in the previous sections, we have applied the model on ECCO and NAS large historical document image datasets. In Table 3.9, the results are summarized regarding the number of illustration pages detected per each dataset.

Table 3.9 Experiment Results of Illustration detection on large historical datasets of ECCO and NAS

dataset	Number of manuscripts	Total number of pages	Number of detected Illustration pages
ECCO_1	154,926	26,019,457	**316,956**
ECCO_2	52,689	6,894,680	**101,416**
NAS	822	527,860	**21,602**

As it is shown in Table 3.9, around 316,000 document images are detected to contain illustrations in ECCO_1. This number in ECCO_2 and NAS is respectively around 100,000 and 21,000. The line chart in Figure 3.11 compares 'the ratio of detected illustrations pages to the total document pages' in ECCO_1 and ECCO_2 over the years. It can be seen that although the total number of document images in ECCO_1 is considerably more than ECCO_2, the ratio of illustration pages in ECCO_2 is more than ECCO_1.

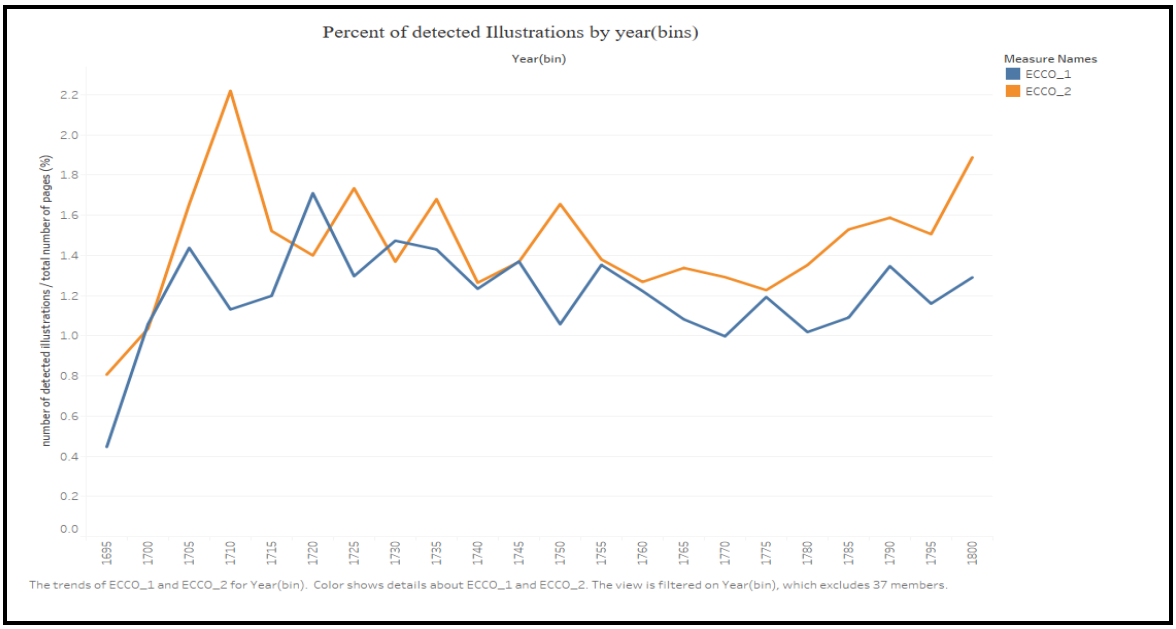


Figure 3.11 Ratio of the detected Illustrations in ECCO_1 and ECCO_2

The distribution of the detected illustration pages in the three repositories of ECCO_1, ECCO_2 and NAS, based on their published year and subject is depicted in the bar charts in Figure 3.12 and Figure 3.13.

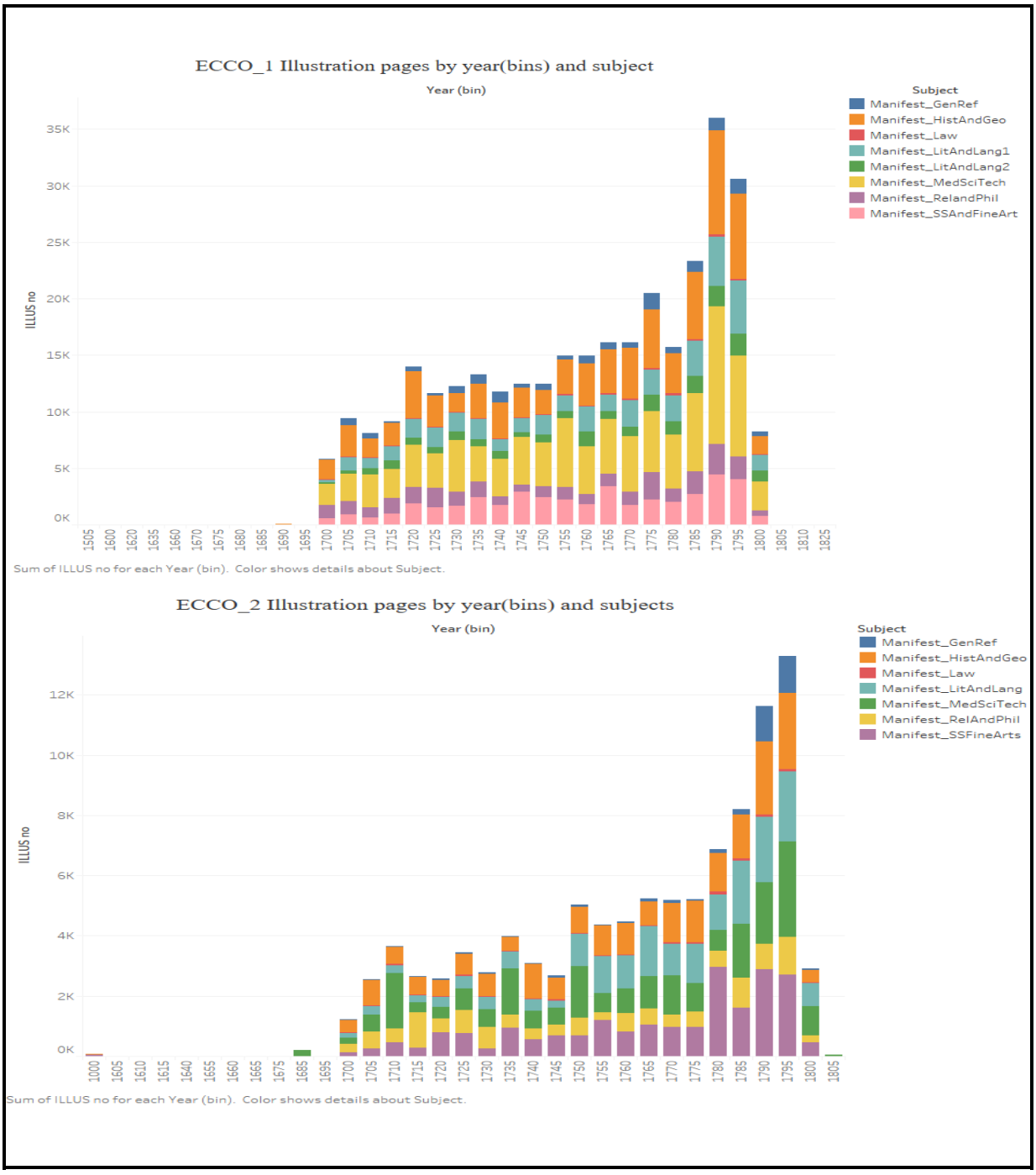


Figure 3.12 Distribution of detected illustration pages in ECCO over year and subject

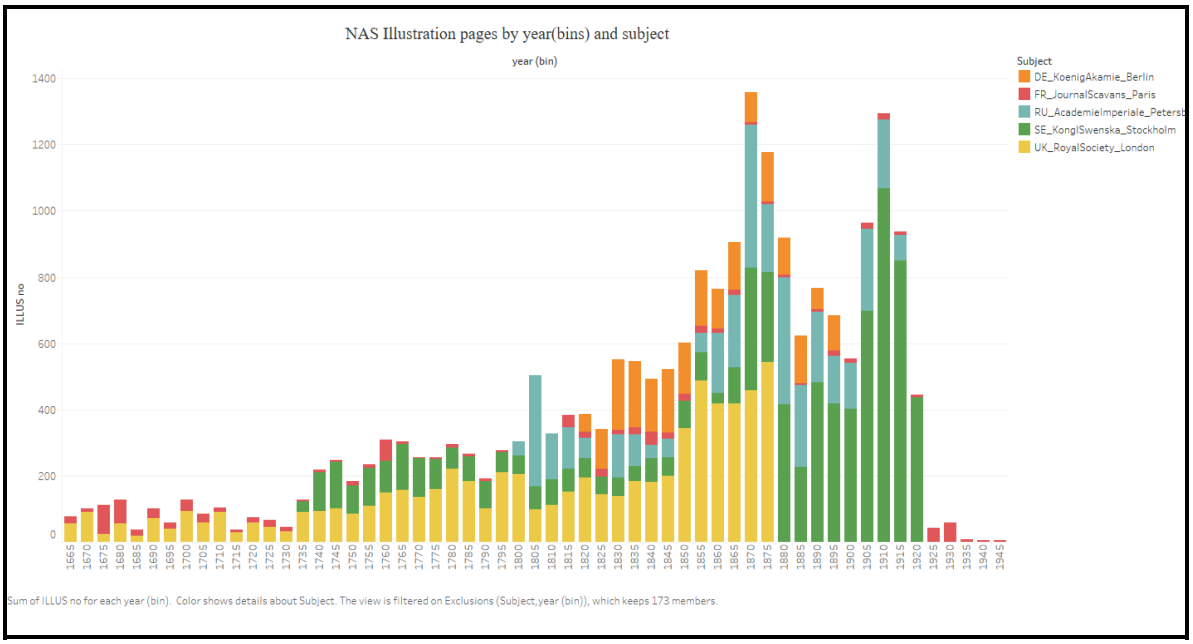


Figure 3.13 Distribution of detected illustration pages in NAS over year and subject

As it is illustrated in Figure 3.12, illustrations detected in the scientific manuscripts outnumbered the other categories and this is what is expected due to the dependency of scientific documents on the illustrations to describe the concepts and theories.

3.3.4 Evaluating the results of Illustration detection in large datasets

In the previous section, we presented the results of applying our model to detect illustrations in full datasets of ECCO and NAS. Large-scale datasets of ECCO and NAS aren't being labeled so the evaluation metrics of Precision and Recall cannot be calculated on the all predicted results on full dataset. To evaluate the results on these large unlabeled datasets, we leveraged the MOE[6] (Margin of error, 2007). MOE calculates the smallest number of samples required to satisfy the defined statistical constraints.

[6] Margin Of Error

$$Sample\ size = \frac{z^2 \times p(1-p)}{\varepsilon^2} \quad (3.5)$$

- *z is z score (initializing based on confidence level value)*
- *p is the population proportion (for instance in illustration detection problem is the proportion of illustration images to all images)*
- *ε is the margin of error*

The results regarding the required sample size calculated for each dataset are presented in Table 3.10.

Table 3.10 Samples sizes to evaluate Illustration detection in large datasets

	ECCO_1		**ECCO_2**		**NAS**	
	Population Proportion	Sample size	Population Proportion	Sample size	Population Proportion	Sample size
Illustration Detection	5%	***1500***	%5	***1500***	9%	***1400***

Based on the calculated sample sizes, the following three steps are performed:

- Samples are selected randomly from the detected illustrations in each dataset;
- Selected samples are labelled manually;
- Assigned manual labels are compared to model predictions to calculate True Positive, False Positive and estimate the Precision on the full datasets.

The results of the above process are summarized in Table 3.11.

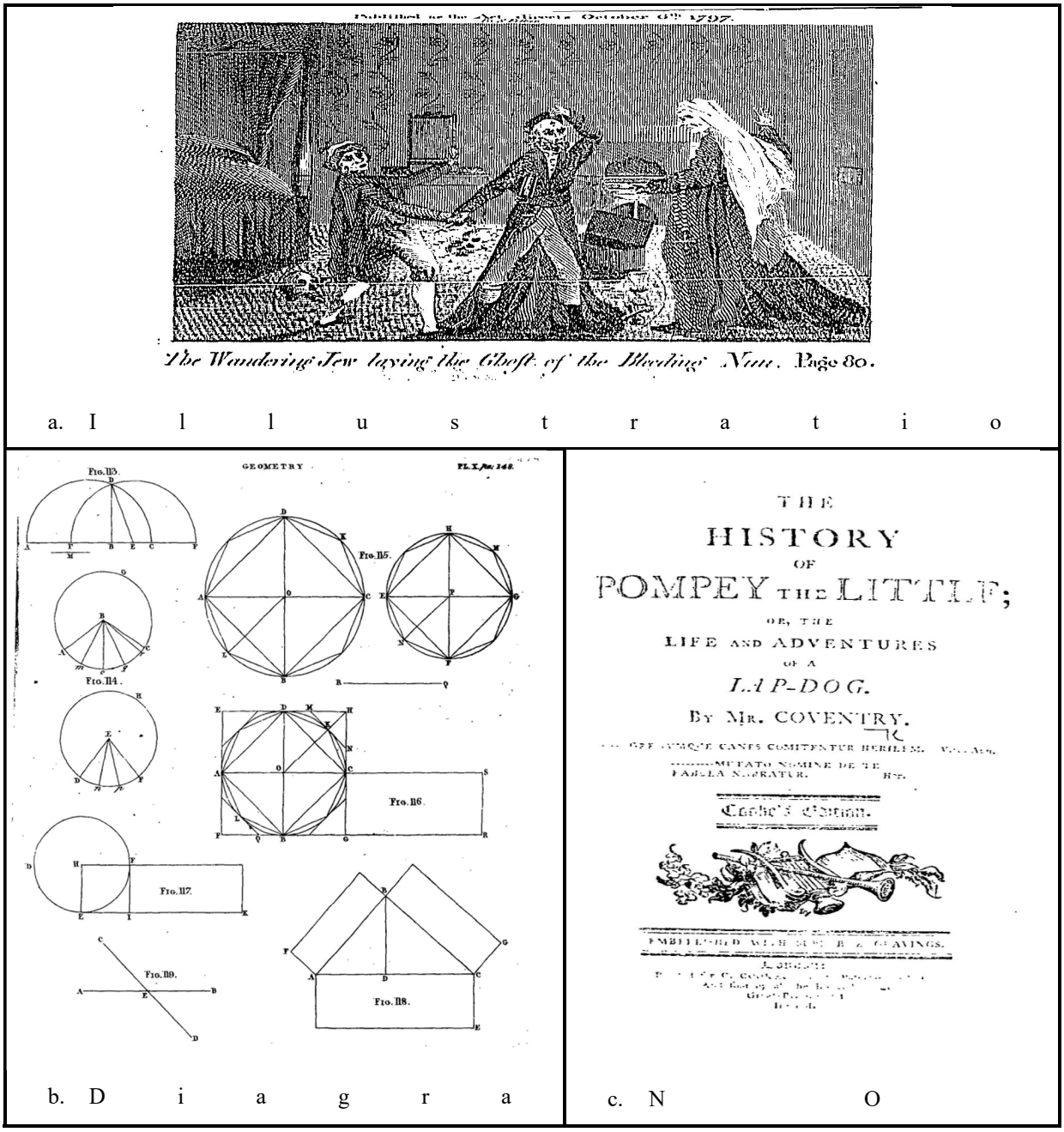
Table 3.11 Performance evaluation of the Illustration detection model on the large datasets

	ECCO_1	ECCO_2	NAS
Sample size	1500	1500	1400
Number of True Positives	1452	1369	1178
Number False Positives	48	131	222
Precision on large dataset	**96.8%**	**91.26%**	**84.1%**
Precision on the small testing dataset	**94%**	**94%**	**77%**

As it is illustrated in the above table, it can be seen that the results on the large datasets of ECCO and NAS are almost as precise as we had predicted previously based on the results on small testing datasets.

3.4 Diagram detection Results

The second step in this thesis, as mentioned earlier in the Introduction chapter, is Diagram detection. In this step, we aim to detect the diagrams out of the illustrations detected in the first step. This is a three-class classification problem with the same network architecture described in Chapter 3. The classes include ‘Illustration’, ‘Diagram’ and ‘NON’. Three samples regarding these three classes are shown in Figure 3.14.



a. Illustratio b. Diagra c. NON

Figure 3.14 Samples of the document image in three classes of Illustration, Diagram and NON

As it is shown, the NON category here is different from the NON category in the first step of illustration detection. In the illustration detection step, the NON category consists of pages containing tables, footnotes or pure text. On the other hand, in the Diagram detection step, the NON category is a subcategory of illustrations detected in the first step which includes

pages without meaningful illustrations and diagrams, such as title pages as shown in Figure 3.14.c.

3.4.1 Diagram detection training dataset specification

In Table 3.12, the labelled datasets used for the Diagram detection in ECCO and NAS are introduced. In each dataset, 20% of the data is put aside untouched for testing and the other 80% considered as the training dataset.

Table 3.12 ECCO and NAS Diagram detection dataset specification

Class	ECCO	NAS
	Number of samples	Number of samples
Illustration	1,486	2,117
Diagram	276	523
NON	637	54

In the Diagram detection task of our project, we had access to almost 2400 labelled images in ECCO and 2200 labelled images in the NAS dataset which are quite small numbers of images required to train a CNN model. That's why we have applied augmentation techniques to our model as described in detail in section 3.7.

3.4.2 Results of Diagram Detection

In the following sections, the results of the Diagram detection on ECCO and NAS datasets are represented in detail.

3.4.2.1 ECCO dataset Diagram detection results

Figure 3.15, shows the loss diagram of the Diagram detection model on the ECCO dataset. As it is illustrated in the figure, the training process of the model during the 100 epochs is quite smooth and according to expectations on a well-behaviour model.

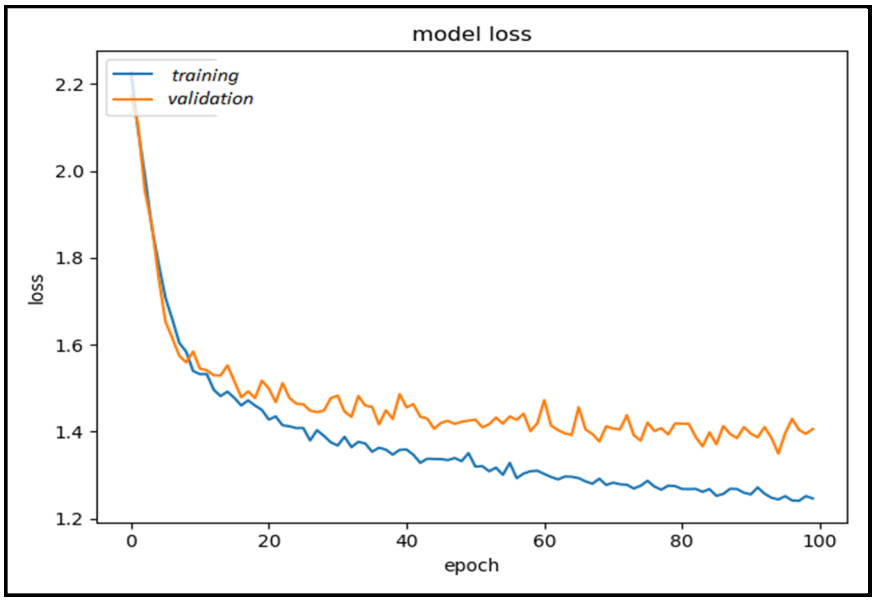


Figure 3.15 Loss diagram of Diagram detection on ECCO dataset

Table 3.13, Table 3.14 and Table 3.15, illustrate the results of the diagram classification on the ECCO dataset based on 'precision', 'recall', 'F1-score' and also 'confusion matrix' metrics.

Table 3.13 ECCO Diagram detection evaluation metrics

	precision	**recall**	**F1-score**	**support(number of images in the validation set)**
Macro average	0.90	0.91	0.90	479

Table 3.14 ECCO Diagram detection evaluation metrics per each class

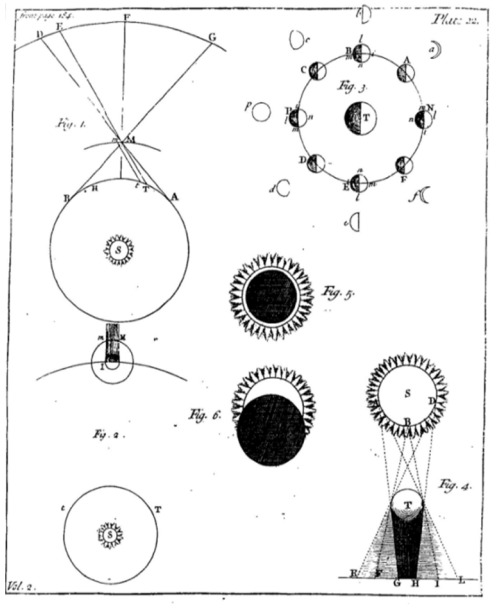
	precision	**recall**	**F1-score**	**Support (number of images in the validation set)**
Diagram	0.84	0.87	0.86	55
Illustration	0.95	0.95	0.95	297
NON	0.92	0.89	0.90	127

The precision and recall in detecting diagrams which is our target class here, in the ECCO dataset are 0.84 and 0.87 respectively. In comparison with Illustration detection, these numbers are smaller. This can be justified by the similarity between the illustrations and diagrams structures that sometimes makes them difficult to be distinguished even by human beings. On the other hand, the fewer number of labelled images to train the model could be another support to describe less evaluation metric's rates of precision and recall.

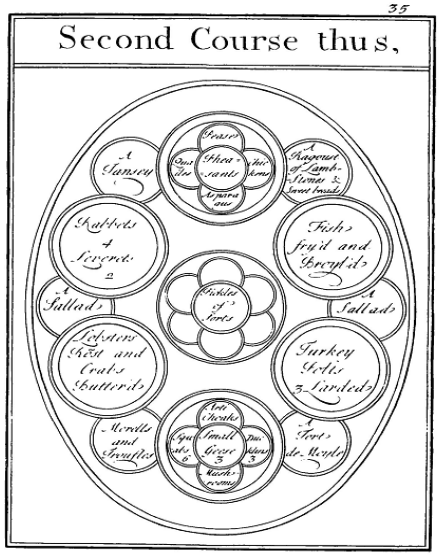
Table 3.15 ECCO Diagram detection Confusion Matrix

		Predicted		
		DIAG	ILLUS	NON
Ground Truth	DIAG	48	6	1
	ILLUS	5	283	9
	NON	4	10	113

As it can be seen, out of the diagram pages, 6 pages are misclassified as illustrations and one image is misclassified as NON. The image in Figure 3.16. a is labelled as diagram in ground truth but the model predicted it as ILLUS. On the other hand, figure 3.16.b is an illustration in the ground truth. As it can be seen, the patterns of these two images are quite similar. They both are made of multiple circles and it seems that is why the model confused to distinguish them.



a: diagram in the ground truth, Predicted as Illustration



b: Illustration in the ground truth

Figure 3.16 Similar structures in DIAG and ILLUS

3.4.2.2 NAS dataset Diagram detection results

The Diagram detection model is then applied to the NAS dataset. The loss diagram can be seen in Figure 3.17 and the results are summarized in Table 3.16, Table 3.17 and Table 3.18.

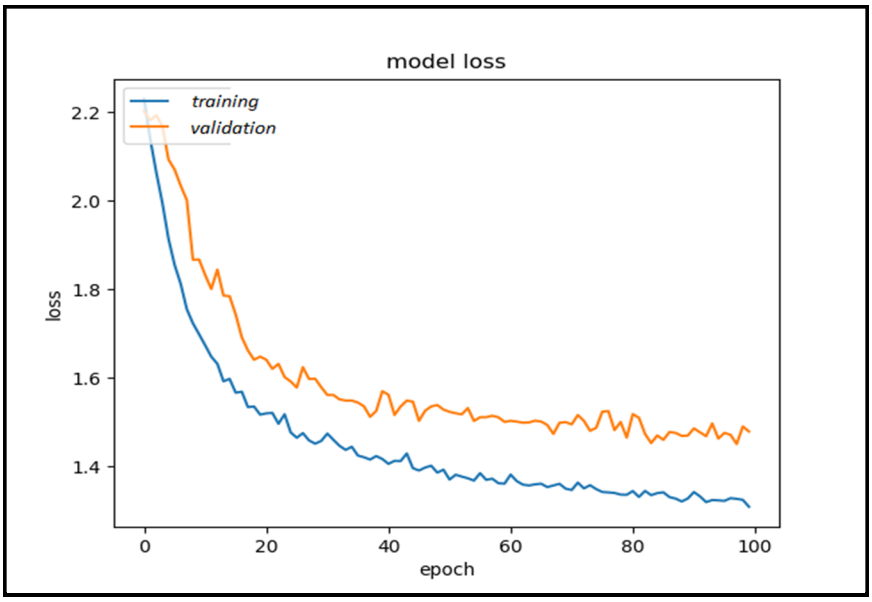


Figure 3.17 Loss diagram of Diagram detection on NAS dataset

Table 3.16 NAS Diagram detection evaluation metrics

	precision	**recall**	**F1-score**	**Support (number of images in the validation set)**
Macro average	0.77	0.83	0.80	523

Table 3.17 NAS Diagram detection evaluation metrics per each class

	precision	**recall**	**F1-score**	**Support (number of images in the validation set)**
Diagram	0.73	0.83	0.77	116
Illustration	0.94	0.90	0.92	395
NON	0.64	0.75	0.69	12

According to the above table, out of the detected diagrams in the NAS dataset, 73 percent of them were detected correctly. If we aim to be more focused in our target class considering our imbalanced dataset, the recall ratio tells us that from all the actual diagram pages, 83 percent of them were correctly identified by the model which is quite an acceptable ratio.

Table 3.18 NAS Diagram detection Confusion Matrix

		Predicted		
		DIAG	ILLUS	NON
Ground Truth	DIAG	96	20	0
	ILLUS	34	356	5
	NON	2	1	9

Two illustration pages that are misclassified as NON are as shown in Figure 3.18.

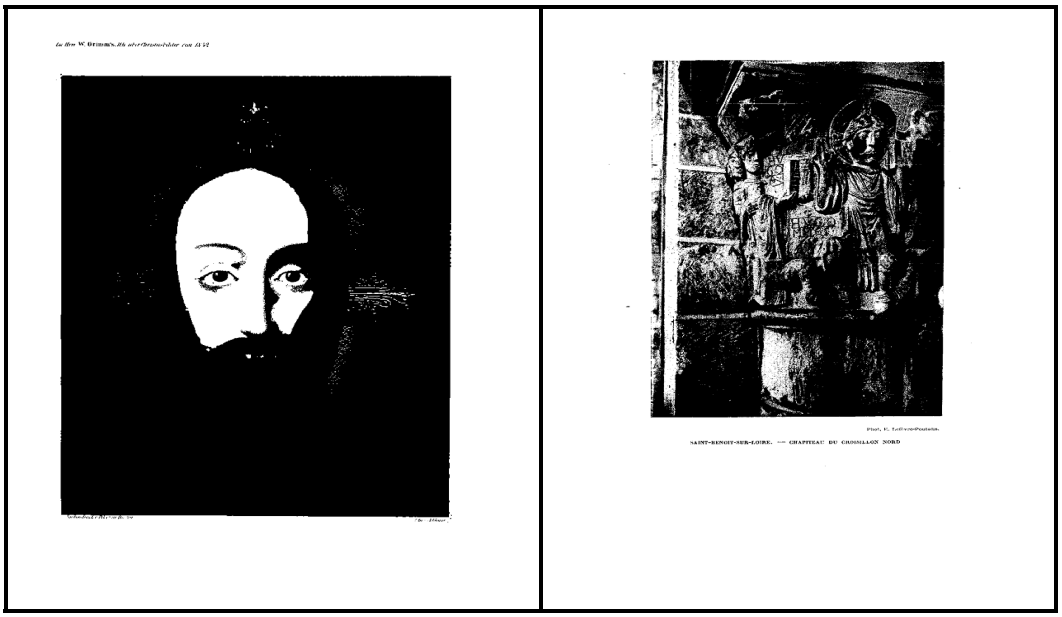


Figure 3.18 Illustrations misclassified as NON

On the other hand, in Figure 3.19, two instances that are labelled as NON in the Ground Truth are shown.

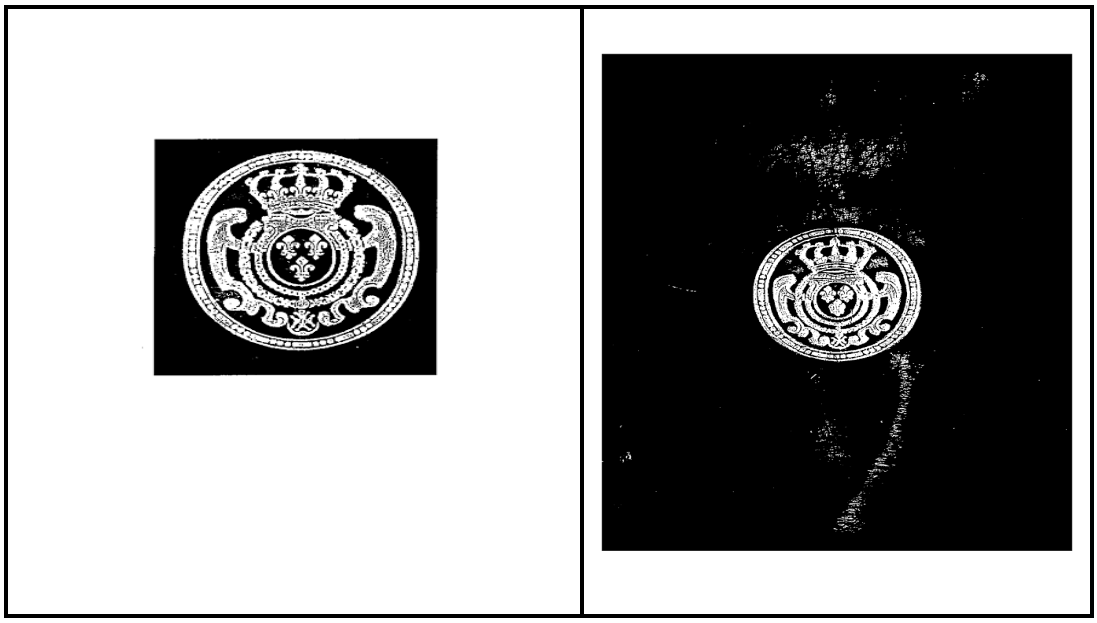


Figure 3.19 Two NON instances in the ground truth

As it can be seen, there are strong similarities between the structure of the images in Figure 3.17 and Figure 3.18 and it makes it difficult for the model to detect them truly.

In Figure 3.20 two diagram pages that are misclassified as illustration are shown.

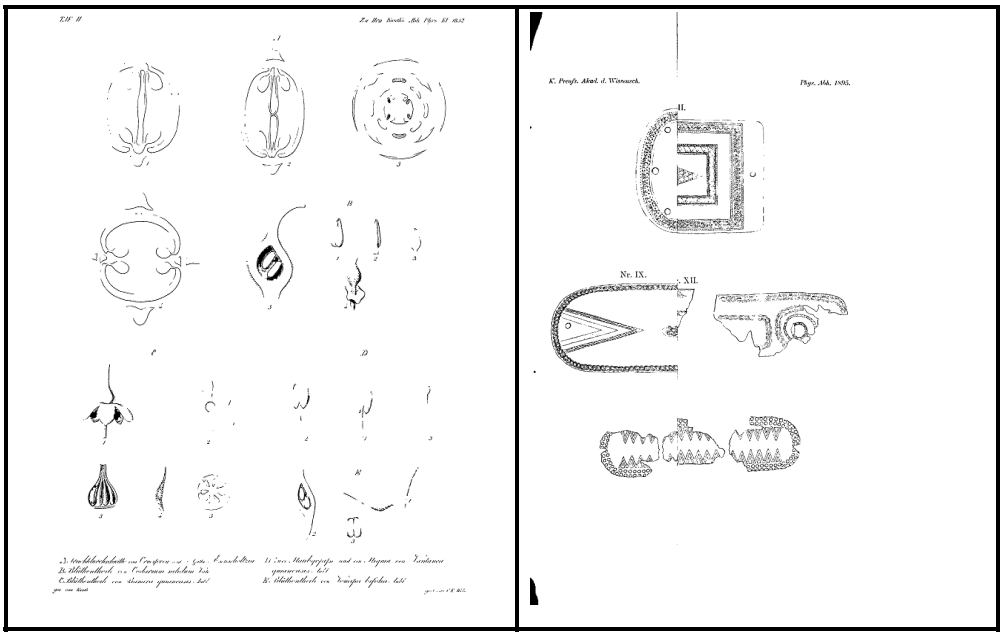


Figure 3.20 Diagram pages misclassified as Illustrations

Also, it can be seen, these diagrams have similar structures as illustrations and that's why the model failed to detect them correctly.

3.4.3 Experiment Results of Diagram detection on large historical datasets of ECCO and NAS

Referring to our main goal of the classification of document images based on the presence of illustration and diagram, in the second step, we have applied our model on the detected Illustrations from the large repository of ECCO and NAS historical document images to detect diagrams from the detected illustrations in the first step. The number of diagram pages detected for each dataset is summarised in Table 3.19.

Table 3.19 Experiment Results of Diagram detection on large historical datasets of ECCO and NAS

dataset	Number of manuscripts	Total number of pages	Number of detected Diagram pages
ECCO_1	39,281	316,956	67,309
ECCO_2	15,034	101,416	16,543
NAS	767	21,602	13,534

Out of the 316,956 illustrations pages in ECCO_1, 67,309 diagrams pages are detected. In ECCO_2 and NAS, 16,543 and 13,534 diagram pages are detected out of the 101,416 and 21,602 total illustration pages.

Figure 3.21 and Figure 3.22 show the distribution of the detected diagram pages in the three subsets of ECCO 1, ECCO 2, and NAS, based on their published year and subject.

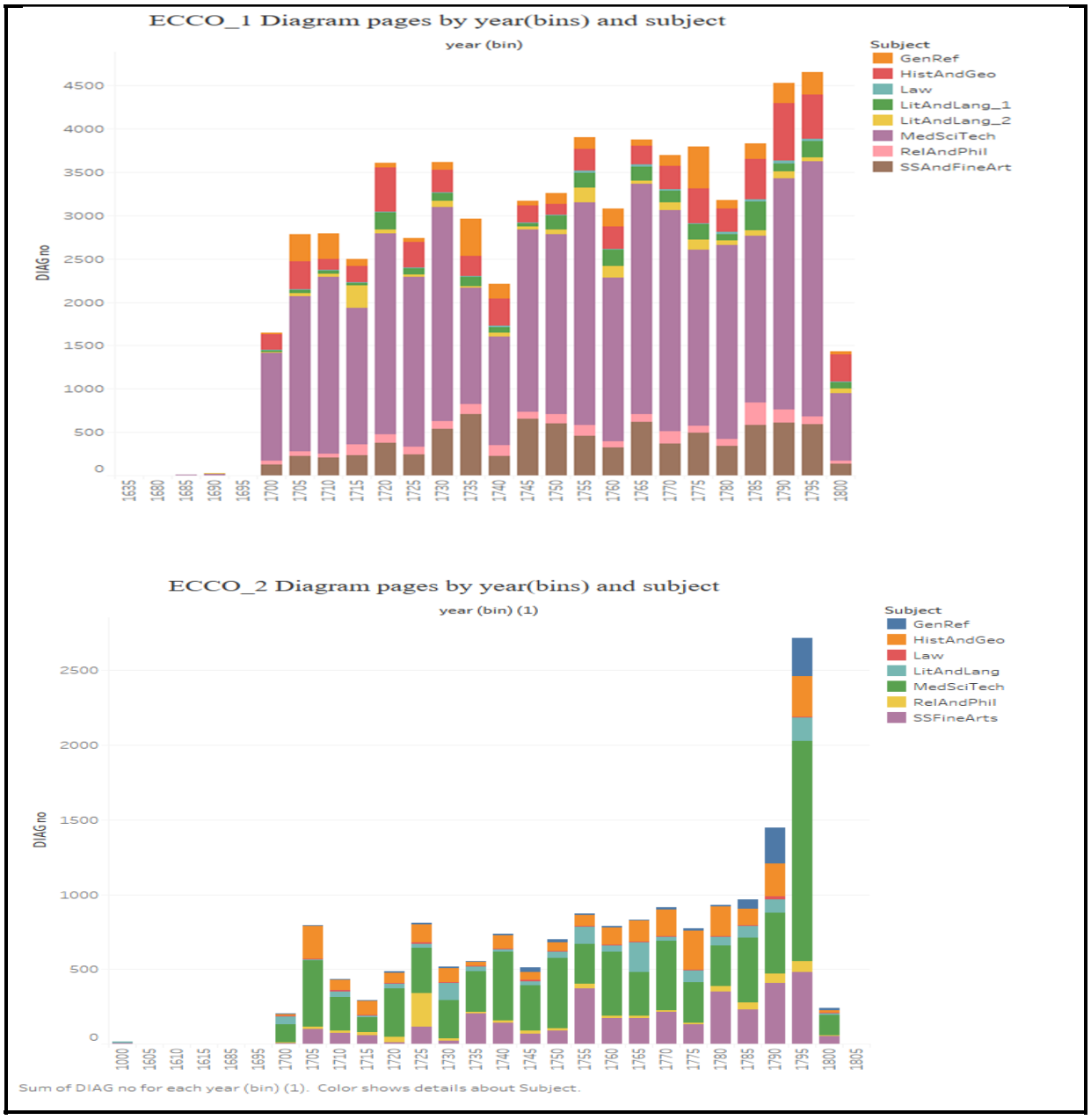


Figure 3.21 Distribution of detected Diagram pages in ECCO over year and subject

As expected, again the Science category is detected to have the most pages containing diagrams in the ECCO dataset.

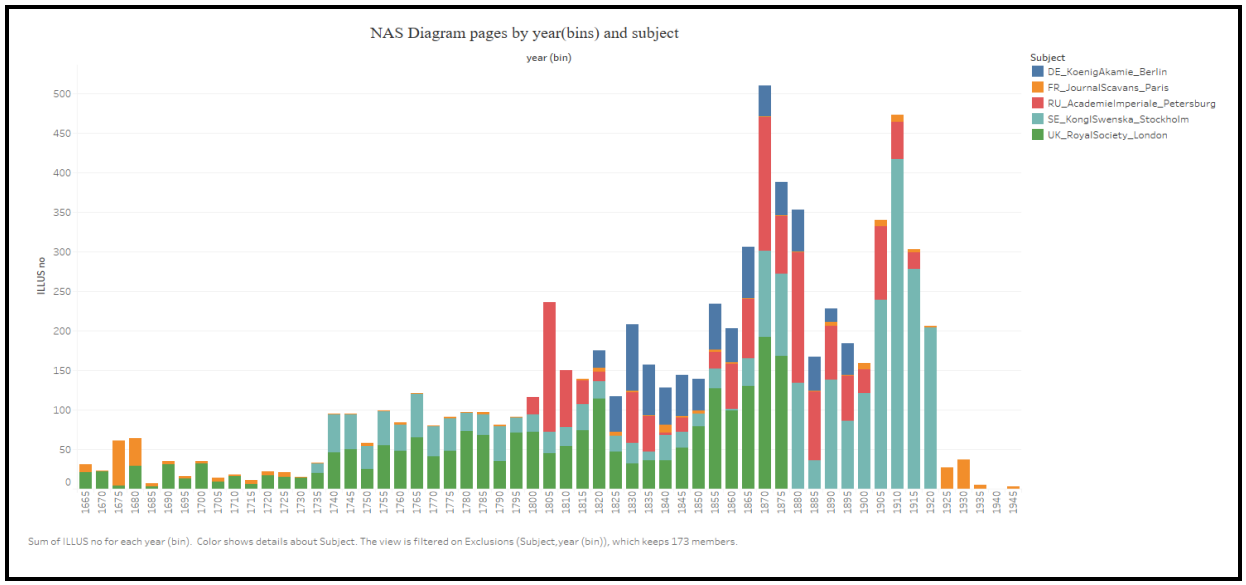


Figure 3.22 Distribution of detected Diagram pages in NAS over year and subject

3.4.4 Evaluating the results of Diagram detection in large datasets

Regarding the diagram detection problem, the same protocol as described in section 4.3.4 is used to evaluate the results on the full datasets of ECCO and NAS.
The calculated sample sizes are presented in Table 3.20.

Table 3.20 Samples sizes to evaluate Diagram detection in large datasets

	ECCO_1		**ECCO_2**		**NAS**	
	Population Proportion	Sample size	Population Proportion	Sample size	Population Proportion	Sample size
Diagram Detection	11%	***1650***	11%	***1650***	22%	***2900***

Table 3.21, represents the results of the evaluation on large datasets.

Table 3.21 Performance evaluation of the diagram detection model on the large datasets

	ECCO_1	ECCO_2	NAS
Sample size	1650	1650	2900
Number of True Positives	1336	1342	2037
Number False Positives	314	308	862
Precision on large dataset	**80.96%**	**81.33%**	**70.26%**
Precision on the small testing dataset	**84%**	**84%**	**73%**

As we can see in Table 3.21, it can be seen that the precision achieved in Diagram detection on the large datasets of ECCO and NAS almost corresponds to the value on small testing datasets.

CONCLUSION

In this thesis, we have proposed a deep learning-based approach to retrieve visual information from large-scale datasets of historical document images. Considering the importance of typographical objects of illustrations and diagrams in supporting the content of the manuscripts and also providing an abstract view to make the complicated texts more understandable, in this work, we aimed to detect document pages that contain illustrations and diagrams.

Due to the diversity of layout and object structures in ancient manuscripts and considering the deficiency of hand-crafted feature extraction-based methods, we have leveraged the capability of automatic feature extraction of CNN-based approaches to deal with these complexities. In our target datasets of ECCO and NAS, we also faced the two common issues the same as other real-world datasets: 'imbalanced dataset' and 'scarcity of labelled data to train the model' which usually would result in overfitting. To address these issues, we have empowered our proposed CNN network with the 'regularization methods of L1 and L2', 'augmentation techniques' and 'transfer learning'. In our implementation of augmentation, during the training of the model, in each epoch, the model is exposed to a new set of randomly augmented image instances from the training set which results in increasing the generalization power of the model. With transfer learning, instead of initializing our model's parameters, randomly, we have initialized them with parameters that are already pre-trained on Image Net dataset. This way, we could enhance the performance of our classification tasks and at the same time decrease the computation complexity and time.

Applying our proposed model on large-scale datasets of ECCO and NAS containing more than 32 Million historical document images with different layouts complexities yields promising results regarding both performance and scalability.

RECOMMENDATIONS

In the process of analyzing the false positives and false negative samples, we find out that despite techniques used for dealing with inadequate training labelled samples, if we had more labelled data that covered more diversity of illustrations and diagrams, we would have achieved more promising results.

There is considerable number of document images that were misclassified as illustrations due to the existence of ‘formula’ and ‘map’ in them which made their layouts different in compare with document images that contain only text contents. It seems that two objects of ‘maps’ and ‘formulas’ could be considered as two categories along with ‘illustration’ and ‘diagram’. In other words, considering the five class classification problem of ‘illustration’, ‘diagram’, ‘formula’, ‘map’ and ‘NON’ would yield more accurate results.

There are considerable numbers of pages that are degraded and contain illegible lines, it seems removing them from both training and full datasets would help the model not be confused in the detection of target classes.

As a future work, combining our model with an NLP module to extract the captions of the illustrations and diagrams create the possibility of extracting an automatic list of illustration pages along with their captions. It could be a precious source of information retrieval for these historical documents.

BIBLIOGRAPHY

Branco, P., Torgo, L., & Ribeiro, R. P. (2015). A Survey of Predictive Modelling under Imbalanced Distributions. New York, United States.

Baluja, S., & Covell, M. (2009). Finding Images and Line-Drawings in Document-Scanning Systems. *2009 10th International Conference on Document Analysis and Recognition* (pp. 1096 - 1100). Barcelona, Spain: IEEE.

Baykal, E., Dogan, H., Ercin, M. E., Ersoz, S., & Ekinci, M. (2019). Transfer learning with pre-trained deep convolutional neural networks for serous cell classification. *Springer Multimedia Tools and Applications*, 15593–15611.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Bouhamed, O., Ghazzai, H., Besbes, H., & Massoud, Y. (2020). Autonomous UAV Navigation: A DDPG-based Deep Reinforcement Learning Approach. *IEEE International Symposium on Circuits and Systems (ISCAS'20)* (pp. 1 - 5). Seville, Spain: IEEE.

Brownlee, J. (2016, March 23). Gradient Descent For Machine Learning. Retrieved from machinelearningmastery.com

Brownlee, J. (2019, January 23). How to Configure the Learning Rate When Training Deep Learning Neural Networks. Retrieved from machinelearningmastery.com

Cheriet, M., Farrahi Moghaddam, R., & Hedjam, R. (2013). Visual language processing (VLP) of ancient manuscripts: Converting collections to windows on the past. *2013 7th IEEE GCC Conference and Exhibition (GCC)* (pp. 407 - 412). Doha, Qatar: IEEE.

Diagram Definition. (2021, June 06). Retrieved from wikipedia.org

Fabio, E. (2018). *Deep Convolutional Neural Networks for Document Classification.* POLITECNICO DI TORINO.

Fashion-MNIST Dataset. (n.d.). Retrieved from github.com

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions on Systems*, 463 - 484.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning - An MIT Press book.* https://www.deeplearningbook.org/.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning (Adaptive Computation and Machine Learning series).* An MIT Press book.

Harley, A. W., Ufkes, A., & Derpanis, K. G. (2015). Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. *13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 991 - 995). Tunis, Tunisia: IEEE.

Harley, A. W., Ufkes, A., & Derpanis, K. G. (2015). Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. *13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 991 - 995). IEEE.

Hastie, T., Tibshirani , R., & Friedman , J. (2009). *The Elements of Statistical Learning.* Springer Series in Statistics.

Howard, G. A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* Google Inc.

Hu, B., Ergu, D., Yang, H., Liu, K., & Cai, Y. (2019). Document images classification based on deep learning. *7th International Conference on Information Technology and Quantitative Management* (pp. 514 - 522). China: ELSEVIER.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology- Wiley Online Library*.

Illustration Definition. (2021, June 30). Retrieved from wikipedia.org

Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2015). Deep Residual Learning for Image Recognition. IEEE.

Kamola, G., Spytkowski, M., Paradowski, M., & Markowska-Kaczmar, U. (2014). Image-based logical document structure recognition. *Springer Pattern Analysis and Applications*, 651–665.

Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014). Convolutional Neural Networks for Document Image Classification. *22nd International Conference on Pattern Recognition* (pp. 3168 - 3172). Stockholm: IEEE.

Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014). Convolutional Neural Networks for Document Image Classification. *22nd International Conference on Pattern Recognition* (pp. 3168 - 3172). Stockholm, Sweden: IEEE.

Kavasidis, I., Palazzo, S., Spampinato, C., Pino, C., Giordano, D., Giuffrida, D., & Messina, P. (2018). A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents. New York, United States: IEEE.

Kavasidis, I., Palazzo, S., Spampinato, C., Pino, C., Giordano, D., Giuffrida, D., & Messina, P. (2018, April 17). A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents. New York, United States: IEEE.

Krahenbuhl, P., & Koltun, V. (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. *24th International Conference on Neural Information Processing Systems.* Curran Associates Inc.

Krahenbuhl, P., & Koltun, V. (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. *24th International Conference on Neural Information Processing Systems.* NIPS.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Springer Progress in Artificial Intelligence*.

Krizhevsky, A. (2009). *CIFAR Dataset.* http://citeseerx.ist.psu.edu. Retrieved from www.cs.toronto.edu: https://www.cs.toronto.edu/~kriz/cifar.html

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *25th International Conference on Neural Information Processing Systems.* USA: Curran Associates Inc.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* Springer.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Li, X., & Plataniotis, K. N. (2020). How much off-the-shelf knowledge is transferable from natural images to pathology images. *PLOS One*.

Liu, Y., Lu, X., Qin, Y., Tang, Z., & Xu, J. (2013). Review of chart recognition in document images. *The International Society for Optical Engineering.*

Ma, C., Huang, J.-B., & Yang, M.-H. (2017). Robust Visual Tracking via Hierarchical Convolutional Features. New York, United States: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Margin of error. (2007). Retrieved from www.cs.mcgill.ca

Max-pooling. (2018, February 27). Retrieved from computersciencewiki.org

Mehri, M. (2015). *Historical document image analysis:.* La Rochelle, France: UNIVERSITÉ DE LA ROCHELLE.

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective.* London, England: The MIT Press Cambridge, Massachusetts.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERIN*, 1345 - 1359.

Parmar, R. (2018, September 11). *towardsdatascience.com*. Retrieved from towardsdatascience.com

Parmar, R. (2018, September 11). *Training Deep Neural Networks*. Retrieved from towardsdatascience.com

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 806 - 813). Stockholm, Sweden: IEEE.

Roy, R. (2020, May 16). Stochastic Gradient Descent (SGD). Retrieved from geeksforgeeks.org:

Saad, A., Tareq, M. A., & Saad, A.-Z. (2017). Understanding of a Convolutional Neural Network. *ICET2017* (pp. 1 - 6). Antalya, Turkey: IEEE.

Saha, R., Mondal, A., & Jawahar, C. V. (2019). Graphical Object Detection in Document Images. *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 51 - 58). Sydney, NSW, Australia: IEEE.

Saha, S. (2018, 12 15). A Comprehensive Guide to Convolutional Neural Networks. Retrieved from towardsdatascience.com

Sarkar, D. (2018, NOV 14). *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning.* Retrieved from https://towardsdatascience.com

Sarkar, D. (2018, November 14). *towardsdatascience.com*. Retrieved from towardsdatascience.com

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Springer Journal of Big Data*.

Stanford Vision Lab, S. U. (2020). *IMAGNET Dataset*. Retrieved from image-net.org

Tripathi, H. (2019, September 24). *What Is Balanced And Imbalanced Dataset.* Retrieved from medium.com

VOK Project. (2016, September 15). Retrieved from txtlab.org

Wang, J., & Perez, L. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *Cornell University Computer Vision and Pattern Recognition*.

Wang, Z. (2015). The Applications of Deep Learning on Traffic Identification. Advances in neural information processing systems.

Xiaohan, Y., Liangcai, G., Yuan, L., Xiaode, Z., Runtao, L., & Zhuoren, J. (2017). CNN Based Page Object Detection in Document Images. *14th IAPR International Conference on Document Analysis and Recognition.*

Zhalepour, S. (2018). *Visual Information Retrieval From Historical Document Images.* Montreal, Canada: Ecole de Technologie Superiere.