ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
DEPARTMENT OF INFORMATICS

DEPARTMENT OF INFORMATICS

M.Sc IN DIGITAL METHODS FOR THE HUMANITIES

Chronological Attribution of Papyri Using Machine Learning

MSc Thesis

ASIMINA PAPARRIGOPOULOU

Athens, November 2021


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY
DEPARTMENT OF INFORMATICS
MASTER OF SCIENCE
IN DIGITAL METHODS FOR THE HUMANITIES

Chronological Attribution of Papyri Using Machine Learning

MSc Thesis

ASIMINA PAPARRIGOPOULOU

Supervisor: Maria Konstantinidou, Assistant Professor
Co - supervisor: John Pavlopoulos, Adjunct Professor
Reviewers: Maria Konstantinidou, Assistant Professor
John Pavlopoulos, Adjunct Professor
Ion Androutsopoulos, Professor

Athens, November 2021


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Abstract

Dating papyri accurately is crucial not only to editing their texts, but also for our understanding of palaeography and the history of writing, ancient scholarship, material culture, networks in antiquity, etc. Most ancient manuscripts offer little evidence regarding the time of their production, forcing papyrologists to date them on palaeographical grounds, a method often criticized for its subjectivity. In this thesis, with data obtained from the Collaborative Database of Dateable Greek Bookhands (https://www.baylor.edu/classics/index.php?id=958430, Baylor University) and the PapPal (http://www.pappal.info/, University of Heidelberg) online collections of objectively dated Greek papyri, we created two datasets of literary papyri and documents respectively, which can be used by machines for the task of computational papyri dating. By experimenting with this datasets, we showed that deep learning dating models, pre-trained on generic images and fine-tuned on a training subset of the data, can achieve accurate chronological estimates for a test subset (69.93% accuracy for bookhands and 56.76% for documents). To compare the estimates of our models with those of humans, experts were asked to complete a questionnaire with samples of literary and documentary hands that had to be sorted chronologically by century. The same samples were dated by the models in question. This paper presents and analyses the results, which show that in some cases the estimates of our models do not deviate from the actual date more than those of humans.

Keywords: papyri, chronological attribution, machine learning, deep learning, CNN


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Περίληψη

Η ακριβής χρονολόγηση των παπύρων είναι ζωτικής σημασίας όχι μόνο για την επεξεργασία των κειμένων τους, αλλά και για την κατανόηση της παλαιογραφίας και της ιστορίας της γραφής, της αρχαίας επιστήμης, του υλικού πολιτισμού, των δικτύων στην αρχαιότητα κ.λπ. Τα περισσότερα αρχαία χειρόγραφα προσφέρουν ελάχιστα στοιχεία σχετικά με τον χρόνο παραγωγής τους, αναγκάζοντας τους παπυρολόγους να τα χρονολογήσουν με βάση την παλαιογραφία, μια μέθοδο που συχνά επικρίνεται για την υποκειμενικότητά της. Στην παρούσα διπλωματική εργασία, με δεδομένα που ελήφθησαν από τις Collaborative Database of Dateable Greek Bookhands (https://www.baylor.edu/classics/index.php?id=958430, Πανεπιστήμιο Baylor) και PapPal (http://www.pappal.info/, Πανεπιστήμιο της Χαϊδελβέργης), διαδικτυακές συλλογές αντικειμενικά χρονολογημένων ελληνικών παπύρων, δημιουργήσαμε δύο συλλογές δεδομένων με λογοτεχνικούς παπύρους και έγγραφα αντίστοιχα, που μπορούν να αξιοποιηθούν από μηχανές για την εργασία της υπολογιστικής χρονολόγησης παπύρων. Πειραματιζόμενοι με αυτά τις συλλογές δεδομένων, στη συνέχεια, δείξαμε ότι μοντέλα χρονολόγησης βαθιάς μάθησης, προ-εκπαιδευμένα σε γενικές εικόνες και προσαρμοσμένα σε ένα υποσύνολο εκπαίδευσης των δεδομένων, μπορούν να επιτύχουν ακριβείς χρονολογικές εκτιμήσεις για ένα υποσύνολο δοκιμής (67,97% ακρίβεια για τους λογοτεχνικούς παπύρους και 55,25% για τα έγγραφα). Για να συγκρίνουμε τις εκτιμήσεις των μοντέλων μας με αυτές των ανθρώπων, ζητήθηκε από τους ειδικούς να συμπληρώσουν ένα ερωτηματολόγιο με δείγματα λογοτεχνικών χεριών και εγγράφων που έπρεπε να ταξινομηθούν χρονολογικά ανά αιώνα. Τα ίδια δείγματα χρονολογήθηκαν από τα υπό εξέταση μοντέλα. Η παρούσα εργασία παρουσιάζει και αναλύει τα αποτελέσματα, τα οποία δείχνουν ότι σε ορισμένες περιπτώσεις οι εκτιμήσεις των μοντέλων μας δεν αποκλίνουν από την πραγματική ημερομηνία περισσότερο από τις αντίστοιχες των ανθρώπων.

Λέξεις κλειδιά: πάπυροι, χρονολογική απόδοση, μηχανική μάθηση, βαθιά μάθηση, CNN


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

To my beloved family, for always motivating and supporting me


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Acknowledgements

I would like to express my warm gratitude to my supervisor, Maria Konstadinidou, and my con-supervisor, Ioannis Pavlopoulos, for their guidance and support throughout the elaboration of this thesis, their invaluable help, constructive advice and the opportunity they gave me to discover and work in such an interesting field. I would also like to thank those who participated in this study, for the time they devoted and their valuable contribution that gave an interesting and deeper insight to my research.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

The object of papyrology is reading, studying, interpreting and exploiting ancient texts preserved on papyrus (Παπαθωμάς, 2016). In reality, however, we cannot define this discipline based on their writing material (Bagnall, 2012), considering that a papyrologist also studies texts surviving on parchment, ostraca, wood, bone, stone and fabric (but not inscriptions, therefore the writing medium must be portable). These texts are exactly the same as the ones surviving on papyrus and they come from the same societies and date to the same periods of time (Παπαθωμάς, 2016). Therefore, it would be more appropriate to adopt Bagnall's definition (2012) that "papyrology is a discipline concerned with the recovery and exploitation of ancient artifacts bearing writing and of the textual material preserved on such artifacts".

In terms of content, we can define two main categories of papyri: literary papyri, bearing texts of literary interest, and documentary ones, bearing texts of various topics of daily life, such as contracts, tax receipts, business letters, etc. (Παπαθωμάς, 2016).

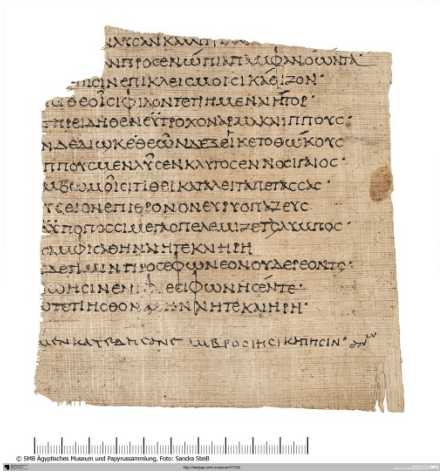

Figure 1: Literary Papyrus[1]

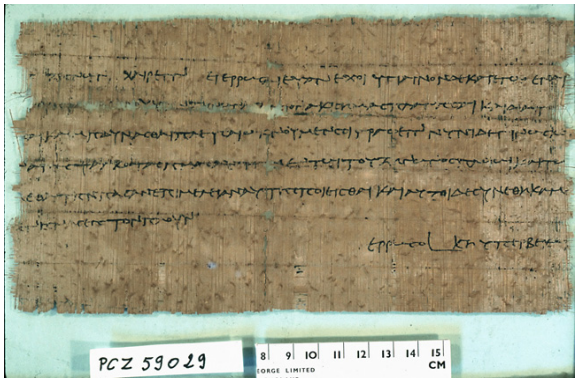

Figure 2: Documentary Papyrus[2]

Dating papyri is considered particularly important for the interpretation and the assessment of their content (Παπαθωμάς, 2016). Documents are often much easier to date, since they frequently bear a date or some reference to known people, institutions, offices or other evidence helpful to that direction. Nonetheless, chronological attribution is not always straightforward: the writers of private letters for the most part did not record dates, while literary texts remain dateless (Turner, 1987). So what methods do papyrologists apply in these cases?

Turner (1987) in his work "Greek Manuscripts of the Ancient World" describes some of the methods employed for papyrus dating. In some cases, archaeological evidence may

[1] P. 6845: Homer, Ilias 8, 433–447. Source: https://berlpap.smb.museum/01720/, Berlin Papyrus Database (BerlPap)
[2] P.Cair.Zen. 1 59029: letter from Antimenes to Zenon. Source: http://ipap.csad.ox.ac.uk/4DLink4/4DACTION/IPAPwebquery?vPub=P.Cair.Zen.&vVol=1&vNum=59029

be of assistance, like the papyri from Herculaneum, which we know were written before 79 BC., when the volcano of Vesuvius erupted. Furthermore, when a document and a literary papyrus are found together in a mummy cartonnage, we can trust the date of the dated documents as a terminus ante quem for the literary text, since both papyri were discarded at the same moment as useless paper. More trustworthy are the dates we can extract when the backside of a papyrus is reused. More specifically, when there is a dated document on the front side (the recto side), then we know that the text on the back (the verso side) was written or copied after the date of the dated document. Conversely, if the dated document is on the back of the papyrus, we know that the text on the front was written or copied before the date of the document. However, in this case we cannot be sure of the time gap between the two. In the event that none of the above evidence is offered for dating, we can take into account the content, such as events that are described or "exploit fashions in 'diplomatic' usage, such as the use of and form taken by abbreviations" (Turner, 1987).

The method used predominantly to get more accurate results, especially when all the other criteria are absent, is based on palaeography, i.e. the study of the script. Dating on palaeographical grounds is based on the assumption that graphic resemblance implies that the two manuscripts are contemporary (Mazza, 2019), as literary papyri are written in elaborate and conservative more formal writing styles that remain unchanged for decades or even centuries, whereas documentary papyri are almost always written in cursive scripts[3] that can be dated with relative accuracy (Παπαθωμάς, 2016). However, this distinction is not absolute, considering that, as stated by Choat (2019), "many dated documents, and the scripts of some of these are sufficiently similar to those of literary papyri for them to form useful comparanda to the latter" and, as Mazza (2019) adds, frequently documentary papyri are written in literary scripts and vice versa literary papyri are copied in documentary scripts. Therefore, it is obvious that relying solely on palaeography is a great challenge that presents plenty and considerable difficulties. For the chronological attribution of a papyrus, the papyrologist should have "a wide range of potential comparanda and have them available for easy consultation" (Choat, 2019). This is not an easy task nor can be achieved without the proper training. Besides, as stated above, the fact that literary texts almost never bear a chronological indication, results in a very small number of literary papyri, securely dated, that can form a basis for comparison. On the other hand, to estimate the date of a papyrus one should take into account all the parameters, like the provenance, the context, the content, the language, the dialect, the codicology, the page layout, the general appearance of the script, the specific letter shapes of the papyrus under examination. (Choat, 2019). Lastly but most importantly, we should not overlook the subjectivity of the whole method, a parameter to which, according to Choat (2019), is given less regard than should be.

Given all these difficulties on the one hand, and the growing development and application of computational means and tools on various disciplines of the humanities

[3]Turner (1987) gives a rather explicit definition of the cursive scripts stating that "the term 'cursive' derives from the concept of a scribe writing in a 'running' movement and lifting his pen as infrequently as he can. Normally he is thought of as applying this running movement to a group of several letters which he will write in a single sequence."

on the other, in recent years many attempts have been made to date manuscripts with the help of computational means. In reality, what these tools and techniques are trying to achieve is the chronological attribution of the manuscripts, based on the palaeographic assumption of the affiliation of scripts, described above, trying, nonetheless, to eliminate the subjective element of this method and to assist the work of palaeographers-papyrologists.

In this thesis, with data obtained from two online collections of securely dated papyri (one set consists of literary papyri and the other of documents), we create two machine-actionable datasets, suitable for the task of computational papyri dating. Then, we exploit machine learning methods on these datasets to train classifiers for chronological classification of papyri. The training of the classifiers aims to achieve successful date estimates of undated papyri. We must emphasize that in no case is our goal to replace traditional dating methods, but to create an additional tool for papyrologists and, along with their experience and specialized knowledge, to achieve more secure dating of the papyrus manuscripts.

This thesis begins with an overview of relevant studies to date and the methods applied for the task of computational manuscript dating. It is followed by a brief description of the datasets created for the purposes of the thesis and a detailed description of the experiments and their results. Finally, the results of a questionnaire given to experts with samples that had to be sorted chronologically are analyzed and compared with those of our models, while the last part of the dissertation presents useful research findings.

2. Related Work

The use of computer means on manuscript images in order to estimate their date of production, the computational papyri dating, is not a novelty, as it can be seen from the study of the relevant literature, discussed in this chapter (Table 1). Furthermore, it is of particular interest the fact that for the computational papyri dating, similar methods to those used for writer identification are exploited. By writer identification we mean the recognition of the writer/ scribe of a manuscript based on writing styles (Dhali et al., 2017) and, according to Hamid et al. (2019), this problem and that of manuscript dating are related to each other[4]. For this reason, it is considered rather useful to present a few relevant works and the methods displayed in them for the task of writer identification.

[4] This can be easily inferred from the study of the literature. Hamid et al. (2018) maintain that "It is evident from literature survey that most techniques proposed for automated manuscript dating are inspired from writer identification and classification". Also, Wahlberg et al. (2016) say that "lately automatic writer identification techniques have been applied to dating". Similarly, He et al. (2016) as well as Dhali et al. (2020) present techniques (features) for date estimation, previously applied in writer identification problems.

	Pub. Year	Evaluation	Method	Language	Period	Dataset name	# Images	Finding
Baledent et al.	*2020*	*Macro F-measure + Similarity*	*DT/RF*	*Latin/French*	*1600 -1720CE*	*GALLICA + DEFT2010*	*8,000 from GALLICA*	*Char > Word*
Dhali et al.	*2020*	*MAE + CS*	*SVR*	*Hebrew*	*250BC - 135AD*	*Dead Sea Scrolls*	*595*	*Capturing handwriting evolution is promising*
Hamid et al.	*2019*	*MAE + CS*	*CNN+SVM*	*Dutch*	*1300-1550CE*	*MPS*	*3,267*	*DL and TL help*
Adam et al.	*2018*	*Accuracy*	*KNN*	Arabic	*8th - 14thCE*	*KERTAS*	*2,000+*	*Whole images of different sizes were used*
Hamid et al.	*2018*	*MAE*	*SVM/KNN/ DT/LDA*	*Dutch*	*1300-1550CE*	*MPS*	*3,267*	*LDA preferred*
Wahlberg et al.	*2016*	*MSE+ percentiles of the absolute errors*	*CNN+GP/ SVR*	*Swedish/ Latin*	*1050-1523CE*	*Svenskt diplomatariums huvudkartotek*	*10,000+*	*ImageNet & 10% fine-tuning data reach the human baseline*
Li et al.	*2015*	*MAE + Accuracy*	*CNN + Word Embeddings*	*English*	*1500-1900CE*	*Google books corpus*	*4,036 volumes (up to 50 pages from each)*	*An OCR-based predictor beats an image-based CNN; the combination works best.*
Wahlberg et al.	*2015*	*MSE, RMSE + percentiles from of the distribution of absolute estimation errors*	*KNN+GP*	*Swedish/ Latin*	*1050-1523CE*	*Svenskt diplomatariums huvudkartotek*	*10,000+*	*MAE<19 although less than 5-% was used to "train"*
He et al.	*2016a*	*MAE + CS*	*SVM*	*Dutch*	*1300-1550CE*	*MPS*	*2,858*	*They applied writer identification to date.*
He et al.	*2016b*	*MAE + CS*	*SVM*	*Dutch*	*1300-1550CE*	*MPS*	*2,858*	*When same writer appears in both training and test sets results are better*
He et al	*2014*	*MAE + CS*	*SVM*	*Dutch*	*1300-1550CE*	*MPS*	*2,858*	*Linear SVR performs better*
Soumya and Kumar	*2014*	*percentage of correct predictions*	*RF*	*ancient Kannada scripts*	Periods of six dynasties	*Canadian script*	*110*	*Epigraphs were studied*

Table 1: Studies of computational manuscript dating

Baledent et al. (2020) compared character-level methods with token-level ones for dating purposes. They selected about 8000 Latin and French documents from 1600 to 1720 from the French digital library GALLICA. The selected documents have plain text access, which in reality is a bad and non-corrected OCR output. Furthermore, experiments were conducted on a comparable dataset, the DEFT2010 (Grouin et al., 2010), from another period which also had OCR issues. The GALLICA corpus was split into a training (70%) and a test set (30%) with the imbalance between the different classes being maintained, whereas for the DEFT2010 corpora the already separated data between train and test was kept. The results showed that a character-level model can handle noise properly to improve classification results as compared to a classical token-level model. Moreover, it became apparent that Decision Trees give good results and Random Forest even better ones.

Dhali et al. (2020) exploited the Dead Sea Scrolls, a collection of ancient manuscripts written in the Hebrew alphabet (derived from Aramaic script) and mostly from 250 BC to 135 AD to create a model that predicts the date of a manuscript. Their study was based on the assumption that the general writing style of each period can be determined if the change of handwriting style over time is captured. Then the date of a document can be easily found by comparing its handwriting style to the general styles of the periods. On account of this, six textural and one grapheme-based feature extraction methods were employed and compared to transform the writing style into a feature vector. For the date estimation, Support vector regression with a radial basis kernel was used. The evaluation results showed that the grapheme-based method is considerably more efficient (MAE: 23.4 years) than the textural methods (with lowest MAE: 42.4 years).

Hamid et al. (2019) used transfer learning on a number of popular pre-trained Convolutional Neural Network (CNN) models in order to estimate the year of production of sample documents from the MPS dataset. The manuscripts were divided into small patches, from which features were extracted with the employment of pre-trained CNN. Also, a number of well-known CNN was fine-tuned on the set of images. The CNN-extracted features were fed into a support vector machine (SVM) classifier, which returned the year of production of a query document as the result of the combination of the decisions on its individual patches with the use of majority vote. The results showed that the Mean Absolute Error (MAE) was significantly reduced compared to existing work on the same problem.

Adam et al. (2018) proposed a dataset, the KERTAS dataset, consisting of more than 2,000 high-quality and high-resolution images of historical Arabic manuscripts dated from 8th to 14th CE, suitable for testing algorithms for age and authorship detection. In their experimental task of age detection, they applied the sparse representation-based method introduced by Wright et al. (2009) that uses normalization to choose the nearest sub-space to the document being evaluated and compared it with three handwriting style-based features. They used two splits, one with predefined folds and one random, and reported accuracy. Moreover, they employed the k-Nearest Neighbor. For the evaluation of the sparse representation-based method they used whole images with no cropping to study both the writing and the layout style and different image sizes starting from 12x12. The highest accuracy is reported when the image size is 50x50. For the

evaluation of the handwriting style-based features, the segmented and binarized text areas were used and according to the results the sparse representation-based method scored the highest accuracy of all handwriting style-based features (94.77%) on the predefined folds and the lowest (42.31%) on the random splits.

Hamid et al. (2018) performed a comparative analysis of popular textural features for the aim of document dating. They used a combination of individual features (Gabor filters, Uniform Local Binary Patterns and Histogram of Local Binary Patterns) to extract a 345- dimensional feature vector, which was then fed to a number of classifiers: Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Decision Trees (DT) and Linear Discriminant Analysis (LDA). These features were evaluated individually as well as combined on the Medieval Paleographical Scale (MPS) dataset, which contains 3267 images of charters written during the years 1300-1550 CE in Medieval Dutch language. According to the results, the combination of Gabor filter features with histograms of LBP features perform better than individual features and LDA leads to improved classification results.

Wahlberg et al. (2016) proposed the employment of deep convolutional neural network (CNN) to estimate the date of production of hand-written documents. They maintain that a CNN can be used directly for date estimation purposes (full network use) or as a feature learning framework for regression (output layer replacement). In their work they explored the latter approach and used Gaussian Processes regression and Support Vector Regression on the Swedish collection Svenskt Diplomatariums huvudkartotek (SDHK). For the experiments the GoogleNet architecture was employed and a model pre-trained on the Imagenet dataset for 120000 iterations (parameter updates) was used to initialize the model. During the evaluation phase, they showed that, when using the CNN only for feature learning, 10% of the collection is needed for fine-tuning the pre-trained, and that the performance can be on average compared to that of a human expert.

Li et al. (2015) used CNN to estimate the publication date of historical English-language documents printed between the 15th and 19th century. They conducted both classification and regression tasks. They built an Image model with Convolutional Neural Network that takes as input a patch of a gray scale image of the document and a Text model with the help of a bag of words, created by an OCR result and used to represent each document as a vector with values that correspond to the number of times each term occurs and the number of out-of vocabulary words. Then this vector was used as input of the neutral network. Finally, they built a Combined model with a CNN that takes both visual and text features as input. For the classification task, they grouped the data into 4 classes of periods of time and used the accuracy metric for evaluation, whereas for the regression task, they rounded the estimated year, and reported the mean absolute error (MAE). As a baseline, for the classification task, they guessed the most common class, and for the regression task, they guessed the midpoint. All three models outperformed the baseline and it was proved that in both classification and regression tasks the combined model is the best of all and the Text model the second best.

Wahlberg et al. (2015) developed a method for large-scale dating of medieval manuscripts, which takes a grey-scale image of a document and with the implementation of the stroke width transform and a statistical model of the gradient image makes sure that the edge pixels always belong to pen strokes. These edges are then analyzed using the shape context descriptor to create a distribution over common shapes on each manuscript page. The evaluation of the method was carried out on the manuscripts collection "Svenskt diplomatariums huvudkartotek" and consisted of over 10000 medieval charters and written in Swedish and Latin. An unweighted k-NN regression, and then, to improve the results, Gaussian process (GP) regression were employed. Even though a 5% subset for training was used, the median absolute error was below 19 years.

He et al. (2014; 2016a; 2016b) in their works tried to estimate the year on which charters from the MPS collection were written. In their 2014 study (He et al. ,2014), they proposed a global and local regression method, in which several textural features, like the Hinge feature and the Fraglets feature, already successfully employed for writer identification are used. It is a three-step method: Global regression, Local support-set selection and Local regression. They trained a linear SVR and non-linear SVR method both for global and local regression. Their results outperformed a conducted random guess, though the Mean Absolute Error (MAE) of the proposed method is still high (35.4 years). In their 2016 study (He et al., 2016a), they applied a family of local contour fragments (kCF) and stroke fragments (kSF), which are scale and rotation invariant, grapheme-based features, capable of capturing the writing style of the documents. These features were encoded into trained codebooks to form statistical histograms, the normalizations of which are the final representations of handwritten documents. They performed dating by handwriting style identification and by classification. According to the results, features which perform well for writer identification are not necessarily that effective at dating historical documents. Furthermore, the classification task showed that the combination of kCF and kSF achieves optimal results (MAE of 14.9 year when the same writer is not included in both training and test sets, and 7.9 years when data are randomly split) while, the experiments conducted by classification with different sizes of codebooks of the kCF and kSF revealed that the MAE decreases as the size of the codebook increases. In their later work (He et al., 2016b), they used a scale-invariant mid-level Polar Stroke Descriptor (PSD) to extract and describe the meaningful handwritten patterns in historical document images. Then, they mapped these patterns into a common space (named codebook), to form a histogram, the normalization of which is considered as the feature representation of the handwritten document. They trained 11, corresponding to the number of the key years, classifiers using a linear SVM and evaluated their method on MPS dataset, applying two scenarios: they split the dataset in a way that the same writer is not included in both training and test sets and randomly. The results demonstrate that the latter scenario performs better (MAE of 7.8 years, while the first scenario has a MAE of 15.1 years) and that the performance improves as the size of the codebook increases. Also, this method is proved to be the most effective of the three presented.

Similar methods were applied for the period estimation not of a manuscript but of an epigraph by Soumya and Kumar (2014). They developed a system that takes gray scale

images of epigraphs written on ancient Kannada script and belonging to periods defined by 6 dynasties (Ashoka, Satavahana, Kadamba, Chalukya, Rastrakuta and Hoysala).The images are binarized and segmented to characters. For the classification, features are extracted from the segmented characters and used to train the Random Forest (RF) classifier. After training, the user selects the epigraph image whose era is to be estimated. The image is binarized, segmented to characters and RF classifies the feature vectors of each segment to one of the classes. Finally, the era of the input epigraph is the majority of eras of the classified segments. Experiments were carried out on 110 Kannada epigraph images from different eras and the system showed good results with up to 85% accuracy for the era identification.

Nasir and Siddiqi (2020) exploited CNN to extract features from digitized historical papyrus manuscripts and identify their writers. The manuscripts were pre-processed and densely sampled to produce small writing patches, from which machine-learned features were extracted with the use of a number of pre-trained CNN. The CNN was fine-tuned using a large dataset of contemporary writings, the IAM handwriting dataset (Marti and Bunke, 2002), and, then, further tuned on the papyrus images from GRK-dataset. To characterize the writer of a manuscript, patch level decisions were combined to document level, by applying a majority vote on patch level decisions and the results showed identification rates of up to 54% among 10 different scribes.

The problem of writer identification was addressed with the use of similar methods in two studies (Mohammed et al., 2018; Mohammed et al., 2019). In the former (Mohammed et al., 2018) the research team proposed a method against common degradation types of historical manuscripts in order to identify their writers. For this purpose, they applied systematically generated degradation on 100 pages of manuscripts selected from the “Stiftsbibliothek” library of St. Gall collection (a virtual manuscript library of Switzerland). Their method, the Normalized Local Naive Bayes Nearest-Neighbour Classifier, takes into account the particularity of handwriting patterns by adding a constraint to prevent the matching of irrelevant keypoints. Keypoints are spotted using Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Features from Accelerated Segment Test (FAST) (Rosten et al., 2010). Experiments showed that SIFT keypoints can cope better with samples of different resolutions, while FAST keypoints can cope better with samples of a very low contrast or a very low resolution. Some of the authors of this study in 2019 (Mohammed et al., 2019) proposed a dataset of Greek papyri manuscripts suitable for the task of writer identification. The GRK-dataset comprises 50 handwriting samples on contracts written by ten different notaries of the 6th century A.D. and was selected by experts for the application of computational-based methods. During the experiments, two image processing and enhancement techniques were applied to enhance the performance of the Normalised Local NBNN classification with FAST keypoints, which was used as a learning and segmentation-free method. The results demonstrate only 30.0% identification rate on the GRK-Papyri dataset with leave-one-out criteria (a version of all the 50 images together was used) and 26.6% identification rate with training-test criteria (a version with a training folder of 20 images -two for each scribe- and a test folder of 30 images was used).

Dhali et al. (2017) conducted a pilot experiment to identify the writers of the Dead Sea Scrolls (DSS) collection using several hand-crafted features. They binarized the manuscript images using Otsu's method (Otsu, 1975) and applied one grapheme-based and eight textural-based feature extraction methods. The pilot study was based on two distinct sets of writers: a limited sample of 323 labeled regions of interest, the FragmROIs, having been written by 13 scribes and a limited sample of 124 FragmROIs, having been written by 13 scribes. The nearest neighbour classification method was performed using the leave-one-out strategy. The results were not the best to be expected, leading the experimental group to proposing statistical modelling, transfer learning, and data augmentation for this limited collection of ancient manuscripts.

Xing and Qiao (2016) proposed DeepWriter, a deep multi-stream CNN for extracting writer-sensitive features. The experiments were conducted on IAM (consisted of unconstrained handwritten English text from 657 different writers) and HWDB (consisted of handwritten Chinese text from 300 different writers) datasets. Firstly, they resized the image of the text while maintaining the aspect ratio, cropped this resized image into patches and uniformly sampled them for testing. On IAM dataset, the DeepWriter was fine-tuned from the, pre-trained on HWDB1.1 dataset, Half DeepWriter model. On HWDB1.1 dataset, the Half DeepWriter was fine-tuned from the above DeepWriter model. The results showed that the models achieve high identification accuracy with little handwritten text input.: 99.01% on 301 writers and 97.03% on 657 writers with one English sentence input, 93.85% on 300 writers with one Chinese character input.

The majority of the works on chronological attribution of manuscripts or writer identification discussed mainly refer to manuscripts written in scripts other than Greek from the medieval or later periods. This may be explained if we take into consideration that among the Greek papyrological corpus, dated samples are very scarce, since dating a manuscript was not a common practice among Greek scribes in antiquity (unlike the medieval times, where they often mention the date of completion in the colophon). Thus, putting together a training set of objectively dated hands is a challenge. Furthermore, most of the mentioned manuscripts fall into the category of documents. The aim of this thesis is the chronological attribution of both documentary and literary papyri in Greek script from antiquity (3rd century B.C.) to the Middle Ages (9th century A.D.).

3. Data

All the data used in the experiments were drawn from two open online collections of manuscripts: the Collaborative Database of Dateable Greek Bookhands (CDDGB)[5] and the PapPal[6].

The Collaborative Database of Dateable Greek Bookhands (CDDGB) is an online catalogue of ancient Greek manuscripts written in literary script, from the 1st to the 9th century A.D, hosted by Baylor University. The data it contains can be dated based on some kind of objective dating criterion, such as the presence of a document that contains a date on the reverse side, or a datable archaeological context associated with the

[5] https://www.baylor.edu/classics/index.php?id=958430
[6] http://www.pappal.info/

manuscript. The list of papyri included in this dataset could have been more comprehensive, a task already undertaken by two ongoing—and much anticipated—projects (ref. from the intro of the CDDGB website). However, for lack of a better alternative and since the collection of objectively dated bookhands goes beyond the scope of this thesis, the CDDGB dataset is deemed adequately reliable for our purpose. Moreover, it is unlikely that a complete list of securely dated literary papyri would increase the number of specimens beyond the lower hundreds, unlike the documentary ones reaching well into the fourth thousand. Table 2 below shows in detail the distribution by century of the image data taken from the CDDGB dataset. The total number of images used is 255. It is worth mentioning, that all manuscripts written in minuscule script were excluded, due to the fact that minuscule Greek cannot be placed confidently into the script evolution process and it does not fall within the area of interest of this thesis.

Century	Number of images
1stBC	1
1stAD	20
2ndAD	87
3rdAD	71
4thAD	18
5thAD	7
6thAD	13
7thAD	2
8thAD	6
9thAD	30
Total number of images	**255**

Table 2: Distribution of images from CDDGB per century

The PapPal is a collection of ancient papyri dated from the 3rd century B.C. to 8th century A.D., originating mainly from Greco-Roman Egypt. This collection contains documents that aim to be a reliable point of reference for the scientist who aspires to study the evolution of the ancient scripts in time and in some cases in space, and therefore assist

in dating non-dated papyri. Furthermore, the different hands that appear at contemporary level, can demonstrate the variety of writing styles that co-existed in a particular period. It should also be noted that this collection contains, in addition to the papyri images, a number of ostraca (about 400), which for the purposes of this thesis are not useful. The total number of images extracted from the PapPal dataset is 3326 and their distribution per century is shown in Table 3.

Century	Number of images
3rdBC	814
2ndBC	581
1stBC	109
1stAD	341
2ndAD	326
3rdAD	346
4thAD	297
5thAD	162
6thAD	257
7thAD	64
8thAD	29
Total number of images	**3326**

Table 3: Distribution of images from PapPal per century

All images taken from these collections vary in format (images from CDDGB are in jpg and png format, while from PapPal in jpg, png and gif) and resolution. To create two datasets, one for each type of papyrus, that can be used by computational approaches for the purpose of papyrus dating, we downloaded the images manually and then renamed them. Image file names took the form shown in Figure 3, with an indication of the century, underscore and the name of the manuscript as it appears in the dataset.

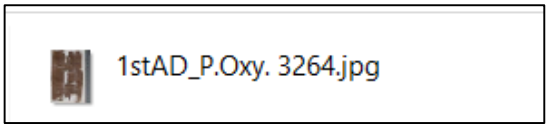

Figure 3: Example of the renamed images

4. Methodology

In this chapter we will outline the computational methods we applied in our experiments. Our purpose is not to describe in detail how each method operates, but to give a simple and understandable overview of them.

4.1. Machine learning (ML)

As mentioned in the introduction to this thesis, we applied machine learning methods to our data for the purpose of chronologically classifying papyri images. In our experiments we opted for supervised learning, which is more suitable in cases of few clearly labeled data (Mahesh, 2020). More specifically, supervised machine learning uses labeled data that is divided into a training and a test set (Mahesh, 2020). Algorithms detect and learn the relationships between patterns in the training set and their labels and, then, try to apply what they have learned by determining labels to the unknown patterns in the test set (Kramer, 2013).

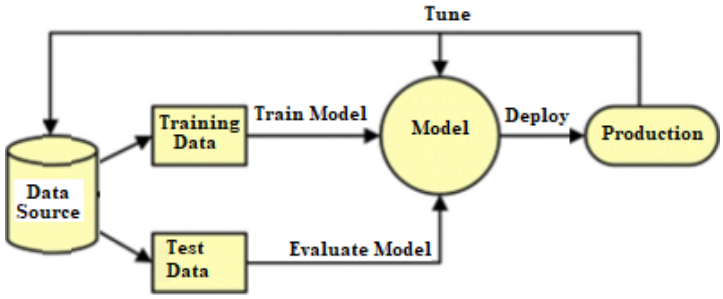

Figure 4: Supervised learning workflow (source: Mahesh, 2020)

Having taken into account the above, we applied the following machine learning algorithms to train models on one part of our data (training set) and to evaluate their performance on chronological attribution of the rest (test set): **K-nearest neighbor classifier (KNN)** [7]**, Support Vector Machines for Classification (SVC)** [8]**, Decision Tree** [9] **, Random Forest** [10] **, Gaussian Processes Classifier** [11] **and Multi-layer perceptrons classifiers (MLP)** [12].

4.2. Deep learning

To improve the performance of the presented machine learning algorithms, we applied deep learning methods on our data. Deep learning is a subset of machine learning and its methods use multiple processing layers to learn representations of data sets (LeCun et

[7] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[10] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[11] https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html
[12] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

al., 2015; Rusk, 2016). A Deep Learning algorithm automatically extracts the features necessary for classification without the need for prior data processing (Lauzon, 2012; Rusk, 2016). In particular, each level learns a concept from the data that subsequent-higher layers build on. The higher the level, the more abstract the concepts that are learned (Rusk, 2016).

However, as already mentioned, Deep Learning methods need large amounts of training data, in order to understand their latent patterns and extract the necessary features. As the number of our data, presented in Chapter 3, is insufficient for a Deep Learning model we applied transfer learning. Transfer learning is a machine learning methodology, which is based on the fact that acquired knowledge can be applied to solve new problems with faster and more effective solutions (Pan and Yang, 2010). In other words, transfer learning is a tool that transfers to the target domain the knowledge contained in related source domains (Zhuang et al., 2020). Domains, tasks, and data distributions of the test and training set do not have to be the same (Pan and Yang, 2010), and the model in the target domain does not need to be trained from scratch. In this way, the problem of insufficient data is significantly addressed, while the training time of the model is reduced (Tan et al, 2018).

The deep learning model we chose to use in our experiments is VGG-16. VGG-16 is a convolutional neural network model (ConvNet[13]) with a depth of 16 weight layers proposed by Simonyan and Zisserman (2014). The model loads a set of weights pre-trained on ImageNet, a dataset of over 14 million images belonging to 1000 classes. The input for VGG-16 is a fixed size of 224 x 224 pixels with 3 channels for an RGB image. The image is passed through a stack of convolutional layers of 3x3 filter with a stride fixed to1 pixel, while five max-pooling layers of 2x2 filter of stride 2 follow some of these conv. layers and carry out spatial pooling. The stack of the conv. layers is followed by three Fully- Connected (FC) layers, the first two of them with 4096 channels each, and the third with 1000 channels. The final layer is the soft-max layer[14]. All hidden layers are equipped with the rectification (ReLU) non-linearity. The architecture of VGG16 is presented in Figure 5. VGG-16 is used in many deep learning image classification problems. This model won the 1st place in the localization task and the 2nd place in the classification in 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

[13] ConvNets are a type of Deep networks designed to process data that come in the form of multiple arrays (LeCun et al., 2015)

[14] The softmax layer is the final layer of some neural networks, which receives values from the previous layer and adjusts the probability of each class (Mueller and Massaron, 2019, 139).

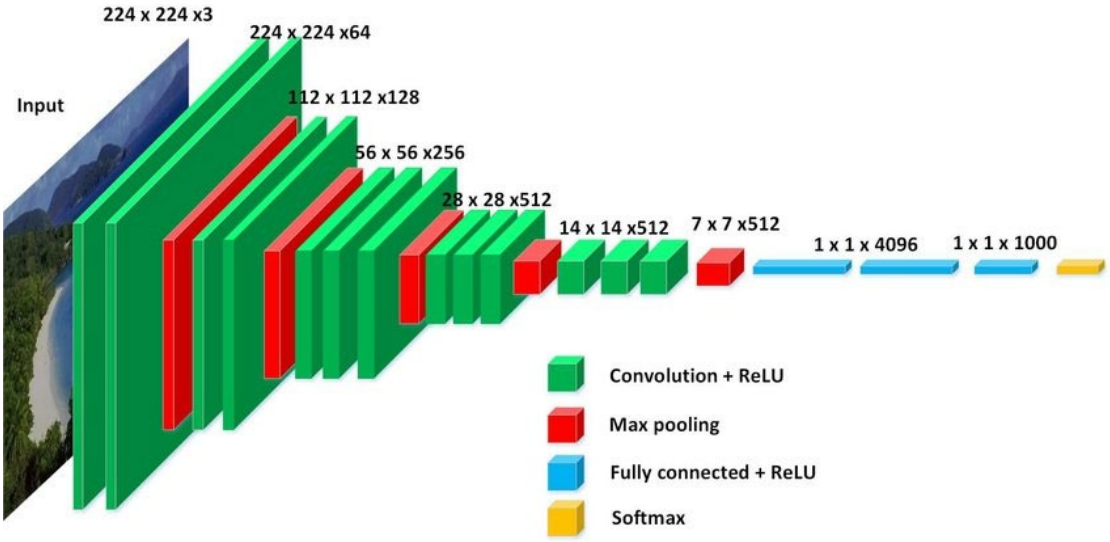

Figure 5: VGG-16 architecture[15]

It should be noted that this particular ConvNet model was used as a feature extractor and not as a predictor. For this purpose, we removed the final two layers (the final fully connected and the softmax), resulting in the last layer having 4,096 output nodes[16]. Then we applied the described machine learning algorithms on the frozen features[17] of this model.

The second deep learning model we chose for our experiments is a deep convolutional neural network model, the Residual Network50 (ResNet50), with a depth of 50 weight layers proposed by He et al. (2016). The ResNet architecture introduced skip or shortcut or residual connections that skip one or more layers to avoid information loss during training (Talo, 2019). The architecture of the model is shown in Figure 6. The model consists of a 7x7 convolutional layer with a stride of 2, followed by a 3X3 max-pooling of stride 2. Then stacks of conv. layers follow and all conv. layers are complied with these rules: for the same output feature map size, the layers have the same number of filters and if the feature map size is halved, the number of filters is doubled. At the end of the network there is a global average pooling layer and a 1000-way fully-connected layer with softmax (He et al., 2016). The ResNet model is pre-trained on the ImageNet 2012 classification dataset that consists of 1000 classes and won the first place at the ILSVRC 2015 classification task and at the ILSVRC & COCO 2015 competitions.

[15] Source: https://www.researchgate.net/figure/VGG-16-network-architecture-for-feature-extraction_fig1_335184836

[16] This method was proposed in https://towardsdatascience.com/how-to-cluster-images-based-on-visual-similarity-cd6e7209fe34.

[17] When we freeze a layer of a model during training, we prevent its weights from being altered, thus we maintain the starting weights of the network layers. This may be necessary to optimize training results (Wu et al., 2019). The features extracted from these layers are called frozen features.

Figure 6: ResNet50 architecture (source: He et al. 2016)

In our experiments we fine-tuned the pre-trained ResNet50 model on a part of our image data and we evaluated on the rest. In other words we employed transfer learning to classify the papyri images into categories-classes of centuries.

5. Experiments and Results

5.1. Data Preprocessing

As mentioned in chapter 3, our data comprise images of various formats and resolution analysis. In order to apply the presented machine learning algorithms on them, we had to do the appropriate preprocessing. First of all, we should note that we worked separately for the literary papyri and separately for the documents. Specifically, we used the Keras library[18] to upload the images from the folder of each type in PIL Format[19] by selecting 224x224 as the target size. Then, we used the function *numpy.array* from the NumPy library[20] to convert each image into an array, as such input is expected by the machine learning algorithms. The results obtained for each image were stored into a dictionary. At the same time, in the same dictionary with "label" as a key we saved the first five characters of each image, which as mentioned in Chapter 3, indicate the date of the papyrus, and with "filename" as a key the whole filename of the image. In this way, we got a dictionary for each type of papyrus including the arrays of their images, the corresponding labels and their respective filenames.

We then took the image arrays and their corresponding labels from the dictionaries, in order to train our models. However, the array of our images had 4 dimensions (number of samples, 224 rows, 224 columns, 3 channels) while the algorithms expect arrays with 2 dimensions. For this purpose, we applied the function *numpy.reshape*[21] from the numPy library, which gives a new shape to the arrays without changing the data and we got a 2 dimensional array (number of samples, 150528 feature vectors).

As the arrays had a very large size, we used the Principal Component Analysis (PCA) technique from the scikit-learn library[22] setting the number of components we wanted to keep to 100. PCA is a technique that performs dimensional reduction. That is to say, it identifies similarities in the structure of data in order to summarize them using less information (Mueller and Massaron, 2019). Thus, it projects the data in a smaller space making it easier for the computer to manage. The final array we obtained was a two-dimensional array (number of samples, 100 feature vectors).

The final pre-processing step we followed was the normalization of the two-dimensional array. Specifically, we divided the array (each feature) by its maximum value in order to scale all the values from -1 to1.

To split the samples into training data and test data, we used from the scikit-learn library the *train_test_split*[23], setting 20% of the samples as a test set. We repeated this split three times, so that for each type of papyrus we have three randomly selected training-test sets.

[18] https://keras.io/api/preprocessing/image/
[19] For PIL format see more on https://pillow.readthedocs.io/en/stable/index.html
[20] https://numpy.org/doc/stable/
[21] https://numpy.org/doc/stable/reference/generated/numpy.reshape.html
[22] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
[23] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

5.1.1. For the application of VGG-16 as feature extractor

As VGG-16 operates with batches of images, after uploading the images, from the folder of each papyrus type, in PIL Format and converting them into an array with 3 dimensions (224 rows, 224 columns, 3 channels) we applied the function *numpy.reshape* in order to add a further dimension that would show the number of images given to the model. The next step was to pass the 4 dimensional array to the *preprocess_input*[24] method imported for Keras library so as to convert the input images from RGB to BGR and zero-center each color channel with respect to the ImageNet dataset, without scaling. Subsequently, from Keras Library we imported VGG16 model[25] removing the final two layers (the final fully connected and the softmax) so that the last layer would have 4.096 output nodes. Finally, to extract the desired features with VGG-16 model we applied the *predict* method[26].

The rest of the process is similar to the one presented above. The results obtained for each image were stored into a dictionary, along with their labels (the first five characters of each image filename), and their filenames. Then, as the array of our images had 3 dimensions (number of samples, 1 row, 4.096 columns), we had to apply the function *numpy.reshape* to get a 2 dimensional array (number of samples, 4.96 feature vectors). Finally, we applied the normalization of the two-dimensional array, already described, in order to scale all the values from -1 to1 and split the samples into training data and test data, by following the same procedure, in order to obtain three randomly selected training-test sets[27].

5.1.2. For the application of ResNet50 model

To prepare our data for the ResNet50 model, we applied similar preprocessing steps to those discussed in section 5.1. Specifically, for each of the two types of papyri, we uploaded the images in PIL Format by selecting 224x224 as the target size and we used the function *numpy.array* to convert each image into an array with 4 dimensions (number of samples, 224 rows, 224 columns, 3 channels). The results (arrays) obtained for each image were, then, stored into a dictionary, along with their labels (the first five characters of each image filename), and their filenames. Then, the images' arrays and their corresponding labels were taken from this dictionary for train the model.

In order to fine-tune ResNet50 we used ktrain[28], which is a lightweight wrapper for the deep learning library TensorFlow Keras. First of all, in order to define the training and the test data for our experiment, we used the *train_test_split*, (from the scikit-learn library) and set 20% of the samples as a test set. Then, we employed the *images_from_array*[29] function from ktrain to get an image generator from training and validation data in NumPy arrays using the training set along with its labels as training data, setting 20% of this data as validation and giving a list of the class names.

---

[24] https://keras.io/api/applications/vgg/#vgg16-function
[25] https://keras.io/api/applications/vgg/#vgg16-function
[26] https://keras.io/api/models/model_training_apis/
[27] The three training-test sets are different from those of the first experiment described in section 5.1.
[28] https://github.com/amaiya/ktrain
[29] https://amaiya.github.io/ktrain/vision/index.html#ktrain.vision.images_from_array

5.2. Experiments

The machine learning classifiers used in our first two experiments (KNN, SVC, Decision Tree, Random Forest, Gaussian Process, MLP) were implemented by scikit-learn library using the default parameters and accuracy, macro precision, macro recall and macro F1 scores, implemented by scikit-learn library[30] too, were used as classification metrics to measure their performance.

During our first two experiments, we also applied the Monte Carlo Cross Validation, a method first proposed by Picard and Cook (1984). Specifically, for both the literary papyri and the documents, with the assistance of each classifier, we trained three models using the three training sets -one for each model- and evaluated them in their respective test sets. Then, we calculated and recorded the average scores obtained from the evaluation of these three models. The process is explicitly presented in Figure 7.

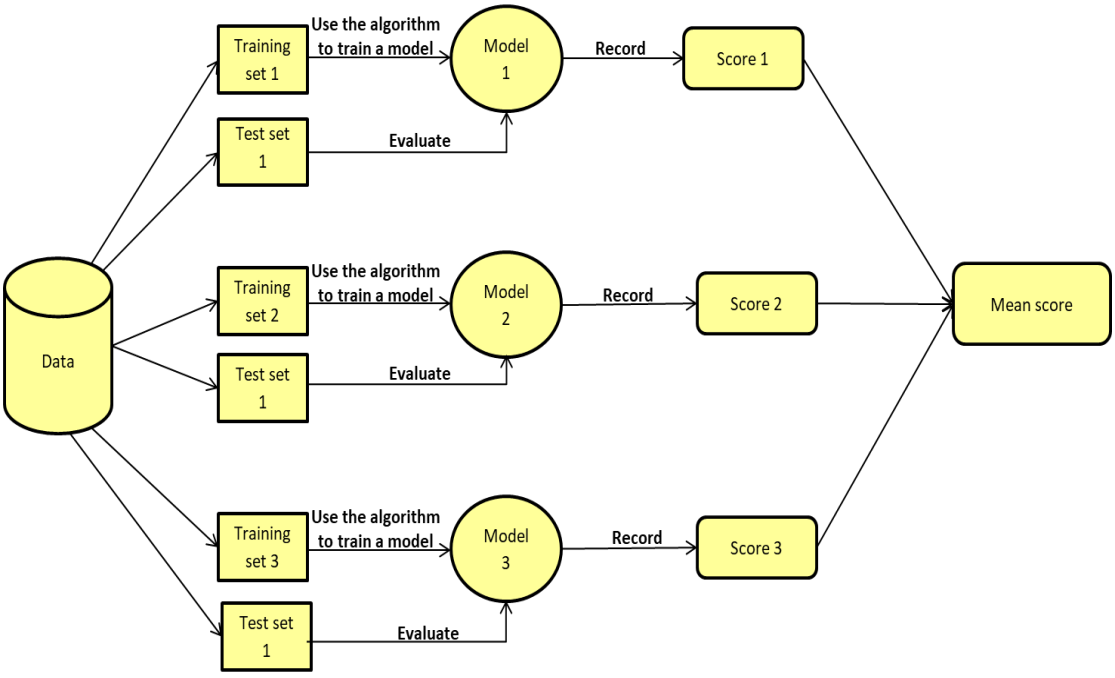

Figure 7: Method applied for each algorithm

5.2.1. Fine-tuning of the ResNet50 model

In ResNet50 experiments, we did not apply the Monte Carlo Cross Validation but, as already discussed in section 5.1.2., we defined only a training- test set, meaning that we used the training set to fine-tuning the model and the test set to evaluate its performance.

[30] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

After the pre-processing phase, we had to select the model to be fine-tuned, which in our case is the, pre-trained on ImageNet, ResNet50 model. Before fine-tuning, we applied the *get_learner*[31] function to get a useful for our task learner instance, in which model and data are wrapped.

In order to fine-tune the model on our data, we froze the first 15 layers (their weighs are applicable as is[32]), with the *learner.freeze* method and trained our model using the *learner.autofit* method. It should be noted that we did not define the number of epochs, thus autofit trained until the validation loss stopped improving[33] after a certain period, defined with the early_stopping argument (in our case was 5). Moreover, the autofit method decreased the learning rate when validation loss stopped reducing, which can be defined with the reduce_on_plateau argument (in our case was 2)[34].

5.3. Baselines

Furthermore, we applied the DummyClassifier[35] from the scikit-learn library, which makes predictions using simple rules. In particular, we chose the most frequent strategy, in which the classifier always predicts the most frequent label in the training set and the uniform, in which the classifier generates predictions at random. Our aim was to use these simple classifiers as baselines and compare their performance with that of the proposed algorithms. Moreover, we should note that for the evaluation of the performance of the Dummy Classifiers we followed the same procedure as with the rest, that is we fitted the Dummy Classifiers using the three randomly selected training sets and we evaluated them on the corresponding test sets. Their final scores resulted from the calculation of the average of these three evaluations.

5.4. Results

The results of the experiments per classifier, including that of the Dummy Classifiers (baselines), for literary papyri are presented in Table 4.

Classifiers	F1-score	precision	recall	accuracy
KNeighborsClassifier	33.56%	34.30%	36.78%	45.75%
SVC	15.59%	20.10%	18.02%	41.18%
DecisionTreeClassifier	14.55%	15.65%	14.65%	28.10%
RandomForestClassifier	30.43%	38.71%	28.85%	41.83%

[31] https://amaiya.github.io/ktrain/index.html#ktrain.get_learner
[32] https://nbviewer.org/github/amaiya/ktrain/blob/master/tutorials/tutorial-03-image-classification.ipynb.
[33] https://nbviewer.org/github/amaiya/ktrain/blob/master/tutorials/tutorial-01-introduction.ipynb.
[34] https://nbviewer.org/github/amaiya/ktrain/blob/master/tutorials/tutorial-01-introduction.ipynb.
[35] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

GaussianProcessClassifier	8.00%	8.51%	12.65%	33.99%
MLPClassifier	10.06%	14.58%	13.37%	33.99%
Baseline: most frequent	5.73%	3.79%	11.80%	32.03%
Baseline: random	5.64%	7.82%	6.03%	9.15%

Table 4: Results by classifier for literary papyri

Figure 8 shows all scores per classifier for literary papyri. What we observe is that most classifiers scored higher accuracy scores compared to the other metrics. In general, all classifiers scored low and in some cases they barely outperformed the baselines. Furthermore, it can be seen that the KNN Classifier outperformed the other classifiers in all metrics except the precision metric at which it is in second place behind the Random Forest. We also notice that the Gaussian Process, the MLP and the Decision Tree Classifier scored lower than the other classifiers (Decision Tree Classifier scored lower accuracy even than the baseline: most frequent).

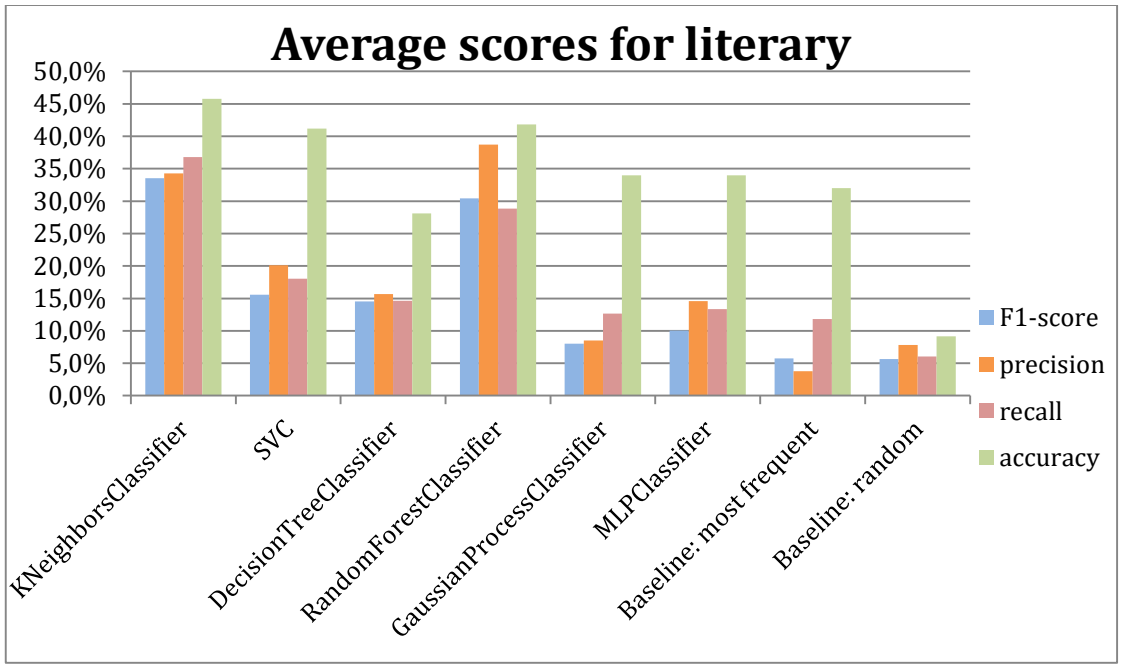

Figure 8: Average scores for literary papyri

Table 5 shows the results of the experiments per classifier and the baselines, for document papyri.

Classifiers	F1-score	precision	recall	Accuracy
KNeighborsClassifier	27.95%	30.14%	28.17%	39.59%
SVC	24.76%	31.45%	26.22%	44.24%
DecisionTreeClassifier	19.30%	19.19%	20.14%	26.93%
RandomForestClassifier	23.71%	43.66%	23.36%	41.74%
GaussianProcessClassifier	16.93%	18.58%	19.68%	38.29%
MLPClassifier	24.80%	28.27%	25.59%	40.49%
Baseline: most frequent	3.69%	2.31%	9.09%	25.43%
Baseline: random	7.74%	8.96%	8.18%	9.36%

Table 5: Results by classifier for documentary papyri

Figure 9 summarizes all of the above scores per classifier for documentary papyri. We notice that the models did not score high with the exception of their performance in accuracy that presents a slightly better image compared to that of the other metrics. Behind the accuracy, most classifiers scored higher precision scores compared to other metrics, while F1 and recall scores move at similar levels for each classifier. The performance of the classifiers per metric shows gradations with the Decision Classifier and Gaussian Process Classifier recording the lowest scores. However, in the case of the documents all models in all metrics outperformed the baselines.

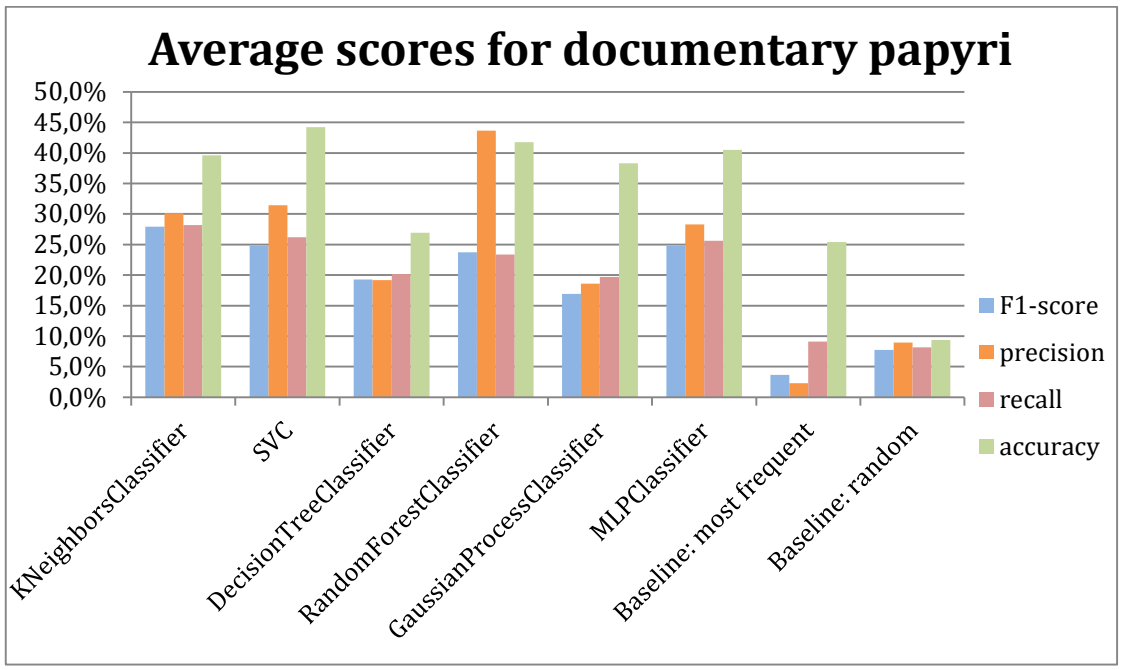

Figure 9: Average scores for documentary papyri

5.4.1. Deep Learning Results

As the results from our first experiments were not very impressive, we tried to apply deep learning methods to our data. Specifically, in our new experiments we used the VGG-16 ConvNet as feature extractor, after removing the last two layers, and then applied the machine learning algorithms on the frozen features, as already described in our methodology chapter. Finally, as a last experiment we used transfer learning and tried to fine-tune a pre-trained on image data deep learning model, the ResNet50 model.

Table 6 shows the results of the experiments conducted with the use of the VGG-16 model by classifier, including the Baselines, for the literary papyri.

Classifiers	F1-score	precision	recall	accuracy
KNeighborsClassifier	39.93%	40.71%	46.39%	52.29%
SVC	27.69%	29.25%	29.71%	55.56%
DecisionTreeClassifier	28.58%	30.88%	29.37%	42.48%
RandomForestClassifier	33.32%	40.39%	32.38%	50.98%
GaussianProcessClassifier	62.44%	72.07%	61.89%	69.93%
MLPClassifier	60.02%	67.94%	57.71%	63.40%
Baseline: most frequent	7.29%	5.05%	13.10%	38.56%
Baseline: random	8.66%	9.95%	10.32%	12.42%

Table 6: Results by classifier for literary papyri with VGG-16

Figure 10 summarizes all the above scores per classifier after the use of the VGG-16 model for literary papyri. We notice that the performance of all classifiers is better than baselines with the exception of the accuracy metric, in which the baseline: most frequent scored quite a high score, which the Decision Tree Classifier barely outperformed. In all other cases the differences in the performance of classifiers-baselines are quite high. We can also see that the Gaussian Process Classifier and the MLP Classifier recorded the highest performance, reaching or even exceeding 60% in some metrics. Conversely, the SVC and the Decision Tree Classifier scored the lowest. Finally, we observe that most classifiers, with the exception of the MLP and the Gaussian Process Classifier, scored higher accuracy compared to other metrics.

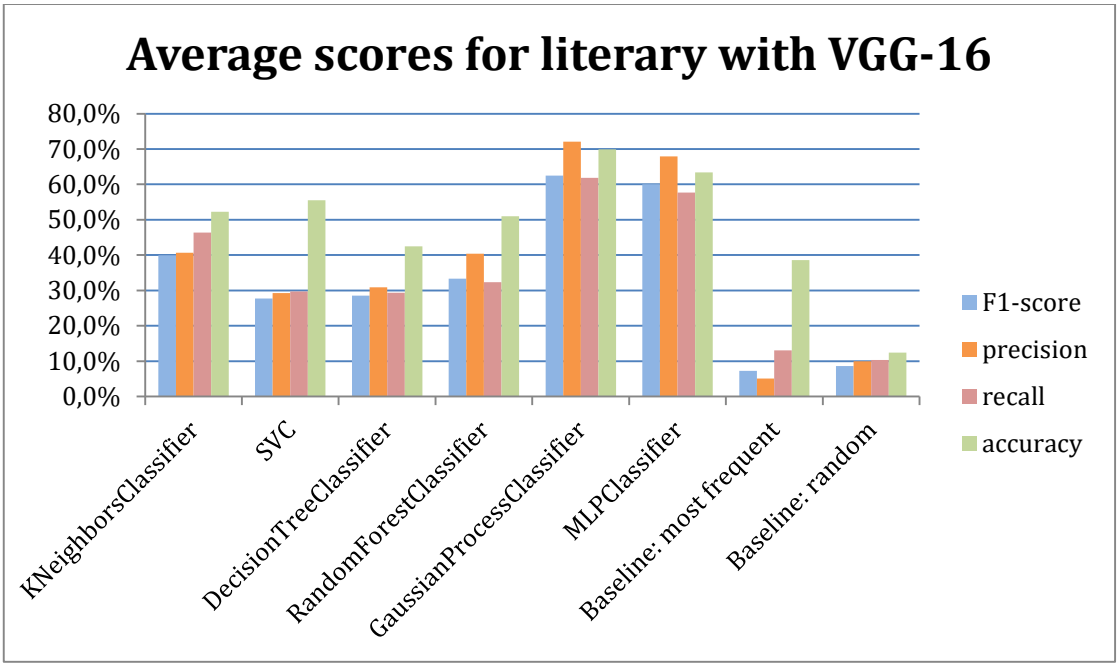

Figure 10: Average scores for literary papyri with VGG-16

Table 7 shows the results of the experiments conducted with the use of the VGG-16 model by classifier, including the Baselines, for documents.

Classifiers	F1-score	precision	recall	accuracy
KNeighborsClassifier	37.86%	45.32%	36.91%	47.00%
SVC	34.75%	43.89%	34.50%	52.80%
DecisionTreeClassifier	22.33%	22.23%	22.69%	33.03%
RandomForestClassifier	31.18%	43.99%	30.42%	49.40%
GaussianProcessClassifier	42.58%	55.46%	40.22%	53.10%
MLPClassifier	47.91%	52.06%	46.09%	56.76%
Baseline: most frequent	3.74%	2.36%	9.09%	25.93%
Baseline: random	7.85%	8.76%	8.25%	9.21%

Table 7: Results by classifier for documentary papyri with VGG-16

Figure 11 shows all scores per classifier for documentary papyri after the use of the VGG-16. It can be seen that the performance of all classifiers in all metrics outperformed the baselines to a significant extent. All classifiers scored higher accuracy and precision, with the exception of the Baselines: most frequent. Moreover, we observe that F1 and recall scores move for each classifier at similar levels. The differences in the performance of the classifiers per category do not present a special gradation with the

highest scores being achieved by the Gaussian Process Classifier and MLP Classifier and the lowest in most cases by the Decision Tree Classifier and Random Forest Classifier.

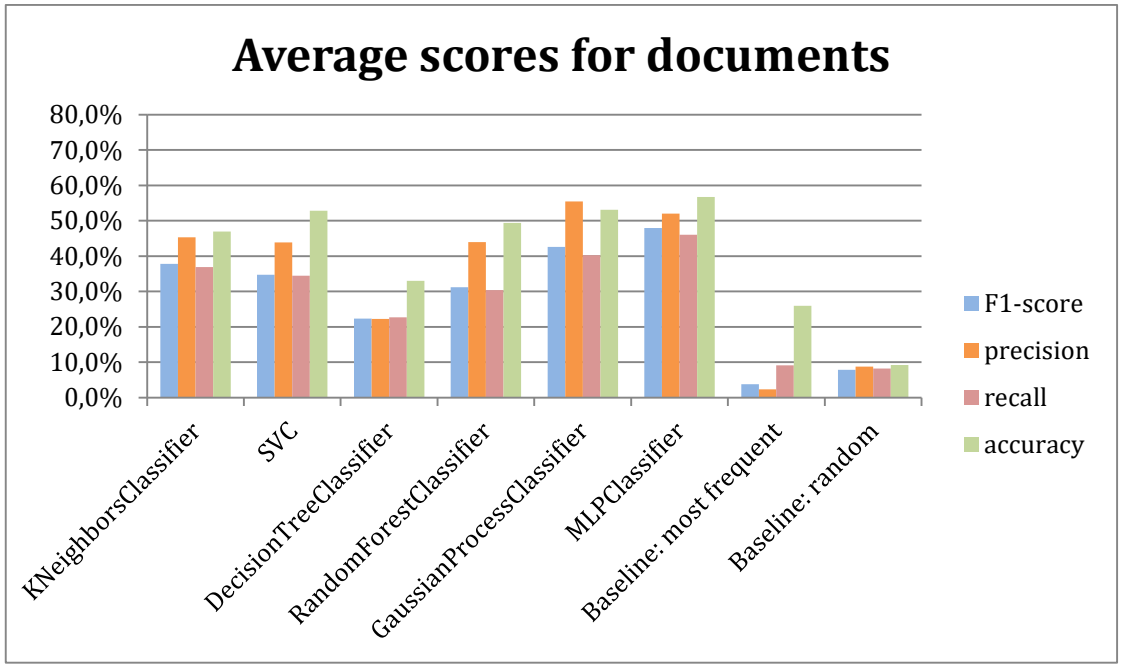

Figure 11: Average scores for documentary papyri with VGG-16

Table 8 presents the results of the experiments conducted with the use of ResNet50 model for literary papyri.

Classifier	F1-score	precision	recall	accuracy
ResNet50	29.75%	22.71%	22.86%	35.29%

Table 8: : Results for literary papyri with the use of the ResNet50 model

Figure 12 illustrates the F1, precision, recall and accuracy score of the ResNet50 model on the literary papyri. It can be observed that the scores are quite low (below 50%) and the model scored highest in accuracy metric.

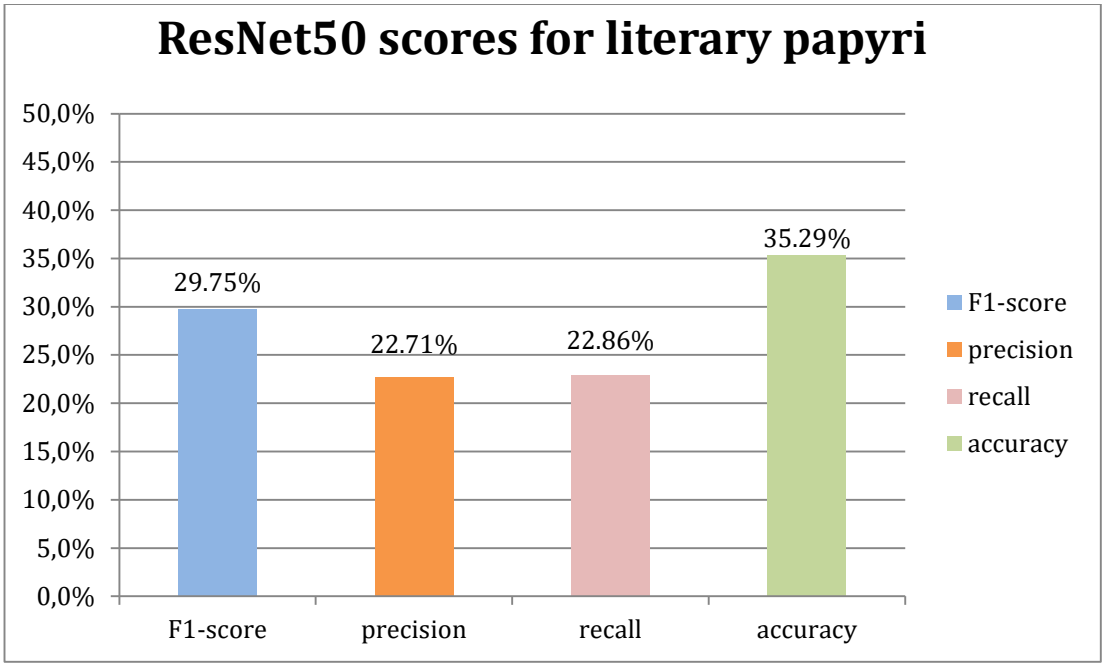

Figure 12: ResNet50 scores for literary papyri

Table 9 presents the results of the experiments conducted with the use of ResNet50 model for documetary papyri.

Classifier	F1-score	Precision	recall	accuracy
ResNet50	39.63%	32.49%	33.93%	46.85%

Table 9: Results for documentary papyri with the use of the ResNet50 model

Figure 13 shows the F1, precision, recall and accuracy score of the ResNet50 model on the documents. It can be noticed that the scores are quite low (below 50%), though better than those on the literary papyri. Furthermore, the accuracy score and then the F1- score are the highest scores.

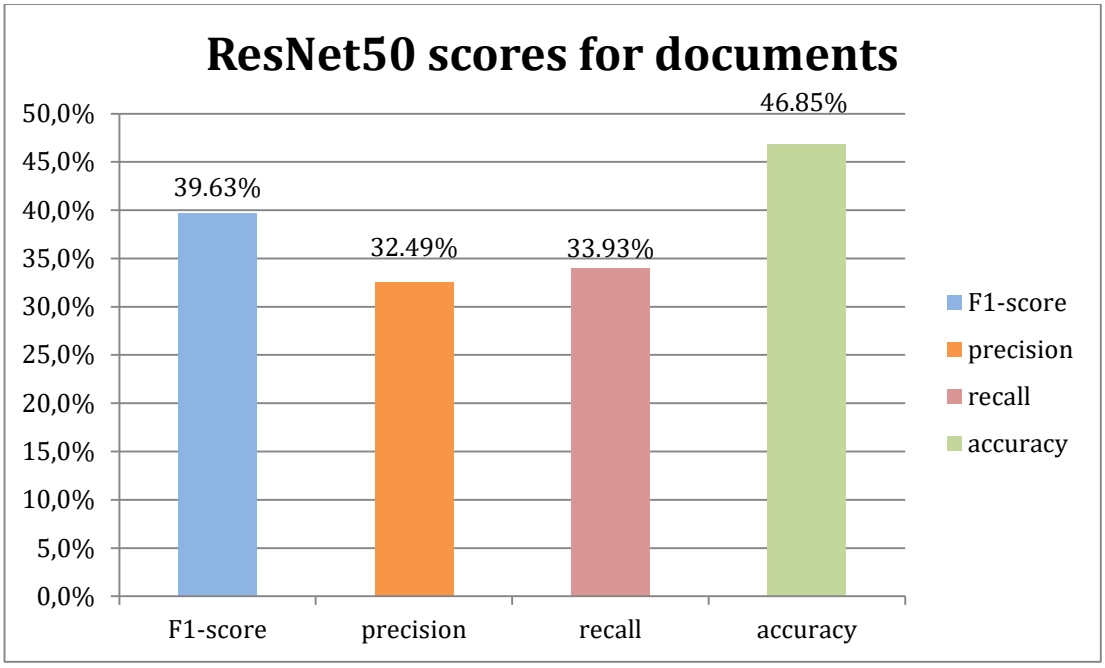

Figure 13: ResNet50 scores for documents

5.5. Learning Curves

Another task we carried out in the context of our experiments was the plotting of learning curves. Learning curves are displays of the performance of an algorithm with respect to the quantity of data used for training (Mueller and Massaron, 2016). With the help of learning curves we can determine the extent to which the models suffer from high variance or bias. In case of high variance, the models make complex hypotheses, i.e. they fit training data very closely (fit noise along with real training data tends), and as a result they cannot generalize to future data, a phenomenon called overfitting. On the other hand, in case of high bias the models make simple and inflexible hypotheses, i.e. they fit the data too loosely (miss real trends along with noise), a phenomenon called underfitting (Briscoe and Feldman, 2011).

In general, when we have high variance we can reduce it either by adding more data or by reducing the number of features or fixing the parameters of the algorithm. On the other hand, in cases of high bias we should increase the number of features or use a more complex algorithm (Mueller and Massaron 2016). Ideally, the learning curves - the one validated on the training set and the one validated on the test set- should start at different points and as we add data they should become close to a common accuracy score (Mueller and Massaron, 2016).

In our experiments, we plotted the learning curves of the models for the documents. First of all, we utilized the division of data into three training-test sets, described in section 5.1 and created for each of the three training sets portions of growing size (500, 1000, 1500, 2000 and 2660 which is the whole training set). We then trained three models on these portions of the training sets - each model on the portions of a training set - and recorded their accuracy score on both the same training data and on the entire

corresponding test set. Finally, we calculated the average accuracy score of the three models validated on the training data and the one validated on the test set and we plotted them on two curves, one for the training sets results and the other for the test sets results[36].

Figure 14 illustrates the learning curves per classifier. For each algorithm we have made a curve that represents the average accuracy score of the algorithm on the test set (solid red line) and on the training data (solid blue line). Also, for both curves we have plotted with dashed lines of the same colors the estimates for their course if we trained the algorithm on more data.

The curves of the KNN classifier seem to rise as we train the algorithm on more data without converging. Even the estimated curves seem to come slightly closer, but their distance remains quite considerable. In a case like this, we say that we have high variance. A similar occurrence can be detected in the case of SVC, in which the curves are noticeably close to each other but, according to the estimates, they will further converge towards a common accuracy score if more data are added. This too is a case of high variance that can be addressed by adding data.

The curves of Decision Tree Classifier and Random Forest Classifier are similar. In these cases, the curves of the training set remain stable at maximum score regardless of the amount of the training data and the curves of the test set rise as we add data- in the case of Decision Tree slightly to not at all, whereas in the case of Random Forest they rise considerably. Both classifiers suffer from the phenomenon of overfitting.

In the case of the Gaussian Process Classifier, the curves seem to converge, before adding the estimated curves. However, their convergence point shows a very low accuracy score, which indicates high bias.

Finally, in the case of the MLP Classifier, the curves appear to become close to each other, but there is still quite a distance between them. Nonetheless, if we add more data, the curves become closer, and their distance is stabilized at the desired point. This is again a case of high variance that can be addressed by adding data.

[36] The process is described by Mueller and Massaron (2016).

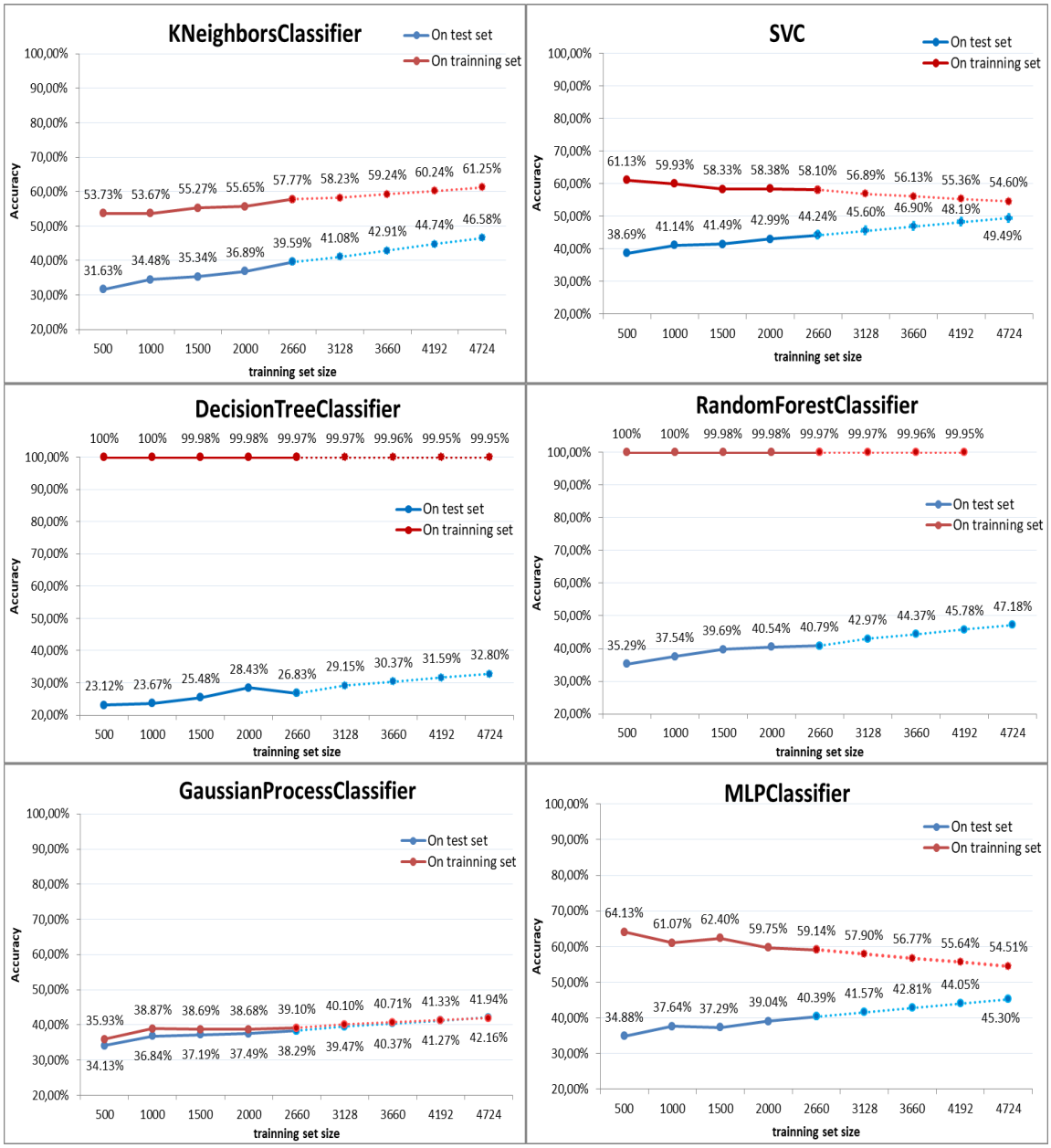

Figure 14: Learning curves per classifier

5.5.1 Learning Curves after the use of VGG-16

Similarly, to our first experiments, after using the deep learning model, VGG-16, we designed learning curves for each algorithm for the documents and presented them in Figure 15. The design follows the same logic as that of our first experiments: for each algorithm the solid red curve represents the average accuracy score of the algorithm on the test set and the solid blue curve on the training data. Additionally, for both curves we have plotted with dashed lines of the same colors the estimates for their course if we trained the algorithm on more data.

The curves of the KNeighbors Classifier have a rising course as we train on more data and, as it can be observed from the estimates, the distance between them decreases at a

very slow pace. This is a case of high variance that can be resolved by increasing the amount of data. Likewise, the distance between the two curves of SVC is quite large. Nonetheless, the estimates show that adding more training samples will help them become close to a common score. And here, we have the phenomenon of high variance that can be addressed by adding data.

The curves of the rest algorithms are significantly similar to each other. In all cases the curves of the training set remain almost invariable - with a slight decrease or increase - at the maximum score regardless of the amount of the training data. On the other hand, the curve of the test set for the Random Forest Classifier, the Gaussian Process Classifier and the MLP Classifier increases with the addition of data, while for the Decision Tree Classifier increases slightly, even after the estimated data addition. All this cases indicate overfitting (high variance).

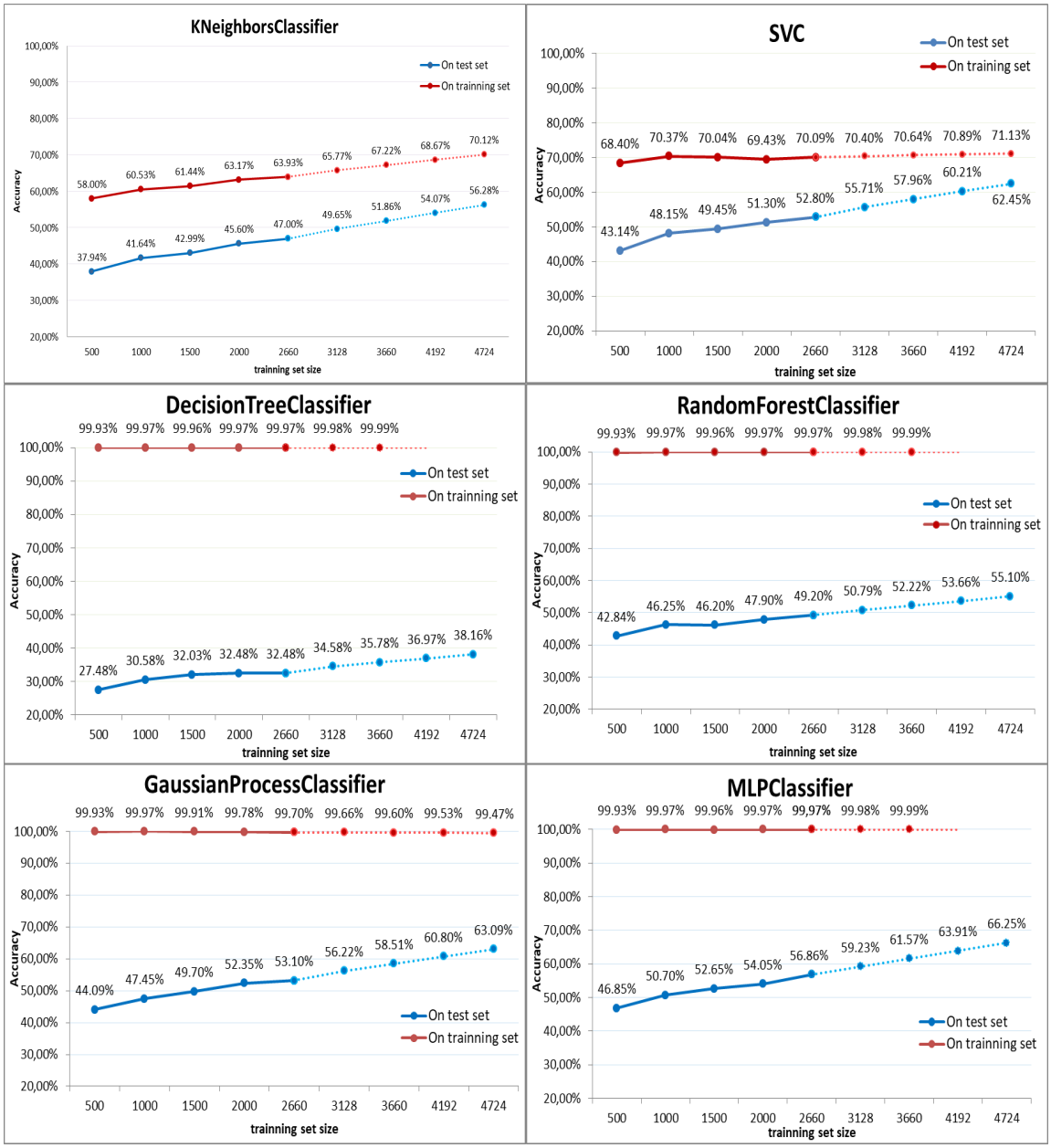

Figure 15: Learning curves per classifier after the use of VGG-16

6. Comparison between the results of the experiments

In this chapter we present comparatively the results of our experiments with the use of the VGG-16 features, without it and with the use of the fine-tuned ResNet50 model for the literary and the documentary papyri. Our purpose is to demonstrate in an apparent way the difference between the scores before and after the application of the pre-trained deep learning models.

In Figure 16 we present the F1, precision, recall and accuracy scores for the literary papyri with and without the use of the VGG-16 model as well as with the use of the ResNet50 model. It can be noticed that in all metrics the VGG-16 enhanced the performance of all classifiers. The Gaussian and the MLP, which scored the lowest (along with the Decision Tree) before the application of the deep learning model, demonstrated the greatest improvement and the best performance after its application in all metrics. On the other hand, the KNN and the Random Forest showed the lowest enhancement on their performance. As for the ResNet50 model, it can be observed that in F1, precision and recall metrics, outperformed most of the classifiers without the use of VGG-16, but only in F1 metric scored higher than some of the classifiers (SVC and Decision Tree) with the use of VGG-16 features.

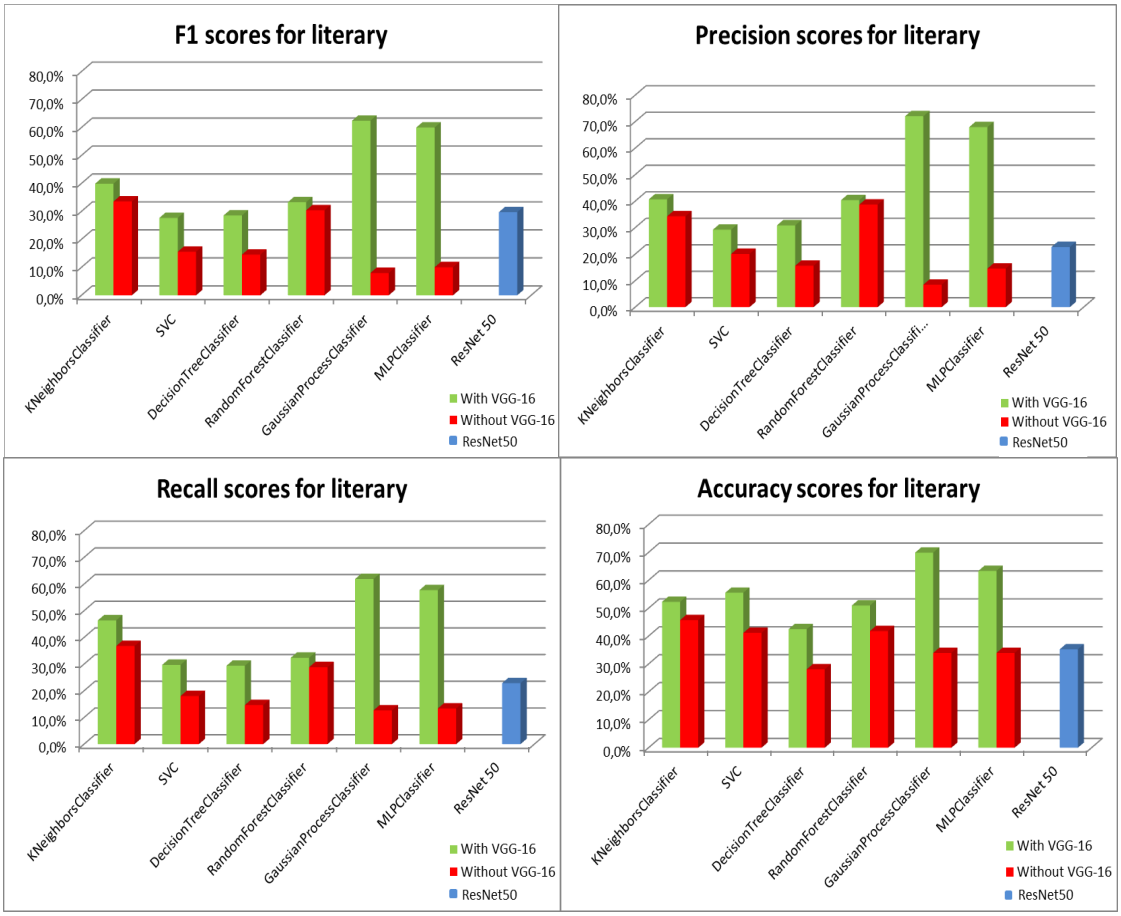

Figure 16: Comparison of scores for literary papyri

Figure 17 illustrates the F1, precision, recall and accuracy scores for the documents with and without the use of the VGG-16 model as well as with the use of the ResNet50 model. In all metrics the application of the VGG-16 improved the scores of all classifiers that did not employ deep learning methods. Furthermore, we can notice that in F1, precision and recall metrics the Gaussian Process had the worst performance without the use of the VGG-16 and the best or the second best after its use. The fine-tuned ResNet50 model outperformed most of the classifiers that did not use deep learning and in some metrics scored higher than some of the classifiers that employed VGG-16. However, it should be noticed that all models in all metrics had a mediocre performance, as they scored lower than 60%.

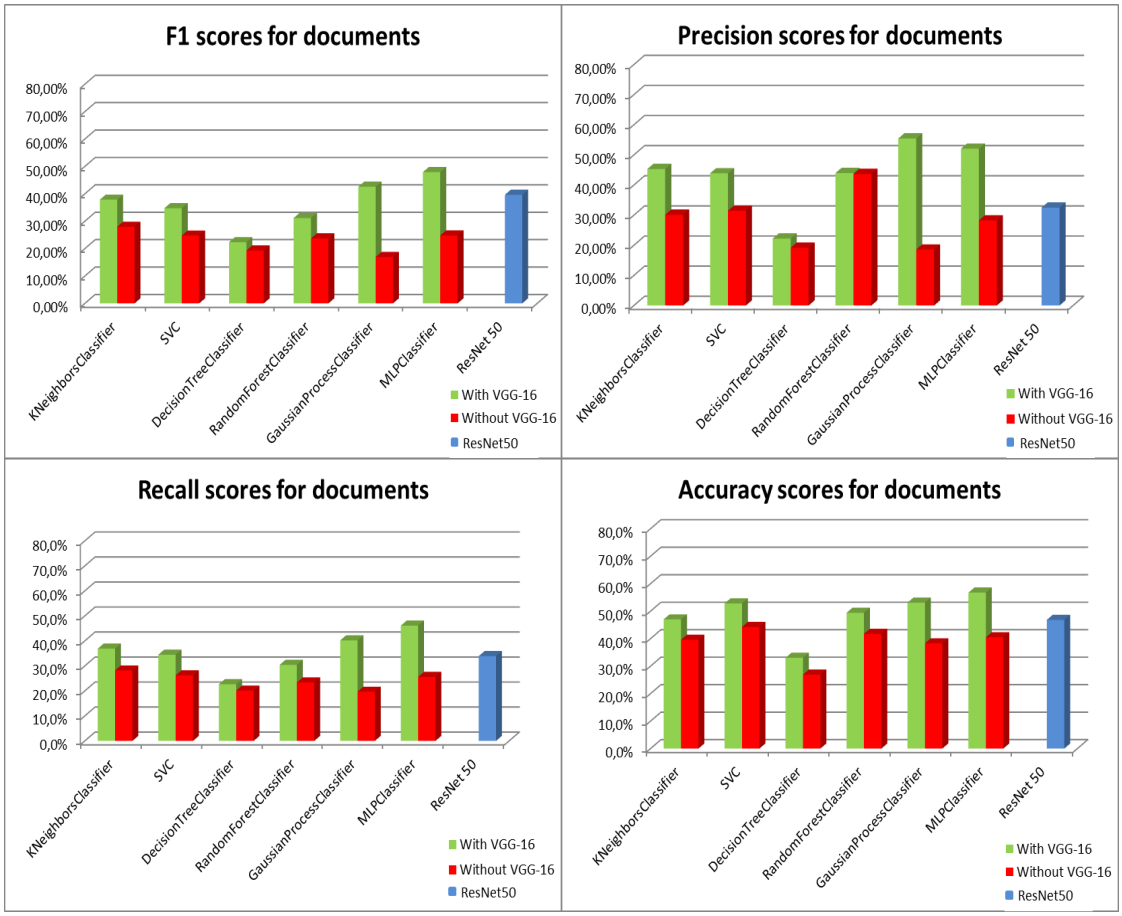

Figure 17: Comparison of scores for documents

7. Results using MAE and MSE metrics

In this chapter we present the results of the evaluation of all our models using the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) metrics, implemented by scikit-learn library[37]. To calculate the MAE, we add the absolute of the difference of the predicted value from the ground truth of all the samples of the test set, and divide it by the total number of samples. Thus the MAE is defined as:

$$\text{MAE}\,(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N-1}|y_i - \hat{y}_i|$$

where N is the number of samples, y is the ground truth and ŷ is the predicted value[38]. To calculate the MSE we add the square of the difference of the predicted value from the ground truth of all the samples of the test set, and divide it by the total number of samples. Thus the MSE is defined as:

$$\text{MSE}\,(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2$$

where N is the number of samples, y is the ground truth and ŷ is the predicted value[39]. As can be seen from their definition, the MAE and MSE metrics show how far a model has failed in its estimates (in our case how many centuries) in relation to ground truth, something that cannot be seen from the metrics used in the previous chapters. Moreover, if these metrics show the failure of the models it is obvious that a lower MAE and MSE indicate a better performance. Finally, we should mention that in our first two experiments (Machine Learning and Deep Learning with the employment of VGG-16) in which we trained three models for each classifier, we calculated the average of the MAE and MSE of these three models.

In Figure 18 the Mean Absolute Error (MAE) of all the models used in our experiments for the literary papyrus is presented. What it can be easily noticed, is that the MAE is lower after the application of VGG-16 model features. Gaussian Process and then MLP, both with VGG-16, outperformed the rest classifiers. Moreover, ResNet50 scored lower MAE than almost all classifiers that did not employ deep learning methods, but higher than those who employed VGG-16 features.

[37] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html.
[38] https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error.
[39] https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error.

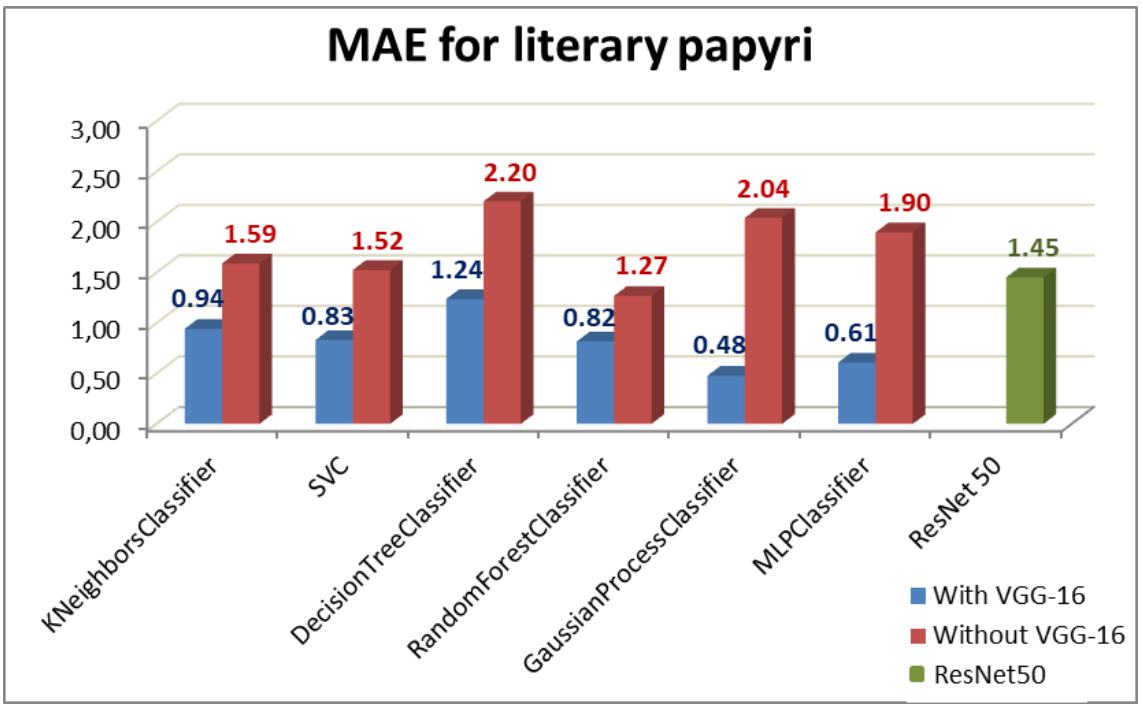

Figure 18: MAE for literary papyri. The numbers represent centuries

In Figure 19 the Mean Absolute Error (MAE) of all the models used in our experiments for the documentary papyrus is illustrated. First of all, we can see that MLP and then Gaussian Process along with SVC, all with the use of VGG-16, scored the lowest MAEs. Furthermore, it is quite obvious that the use of the VGG-16 model enhanced the performance of the classifiers. As for the ResNet50 model, it can be observed that it scored lower MAE than all classifiers that did not employ deep learning methods, and lower even than Decision Tree and Random Forest after the use of VGG-16.

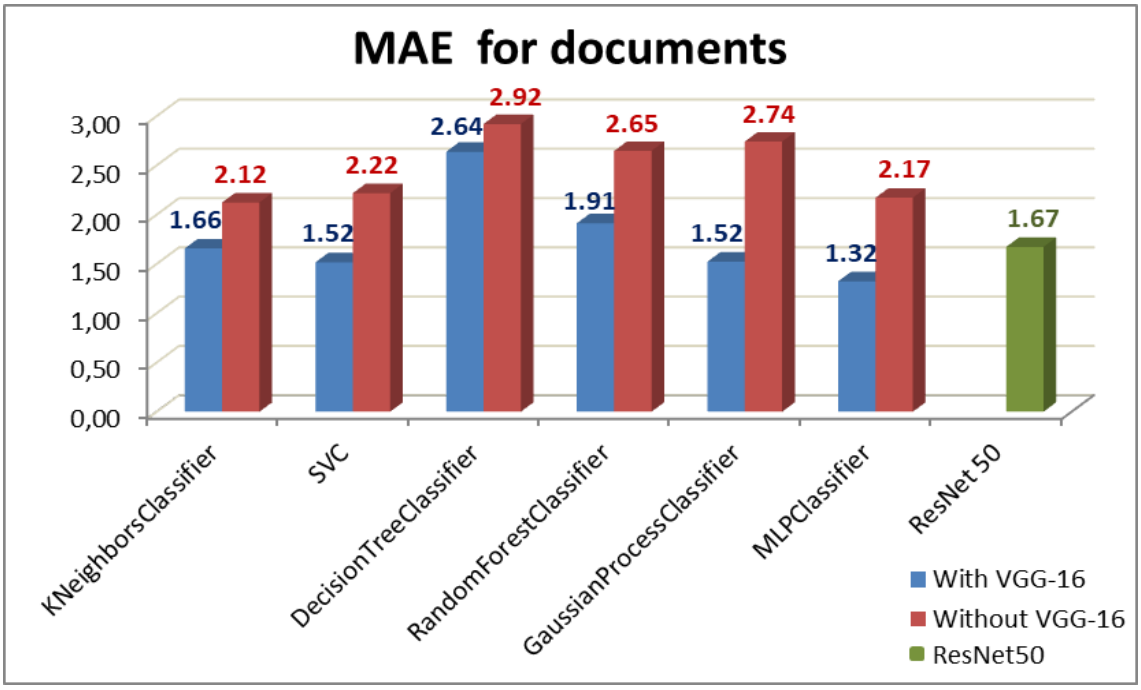

Figure 19: MAE for documents. The numbers represent centuries

Figure 20 shows the Mean Squared Error (MSE) of the models used in our experiments for the literary papyri. What we notice is that the Gaussian Process and the MLP Classifier, both with the employment of VGG-16 model, scored the lowest MSEs. Additionally, we can see that the performance of all classifiers without the use of deep learning methods is quite poor compared to the one after the employment of deep learning models. The fine-tuned (on our data) ResNet50 model scored lower MSE than almost all classifiers that did not use the VVG-16 model (except for the Random Forest) but higher than those who used VGG-16's features.

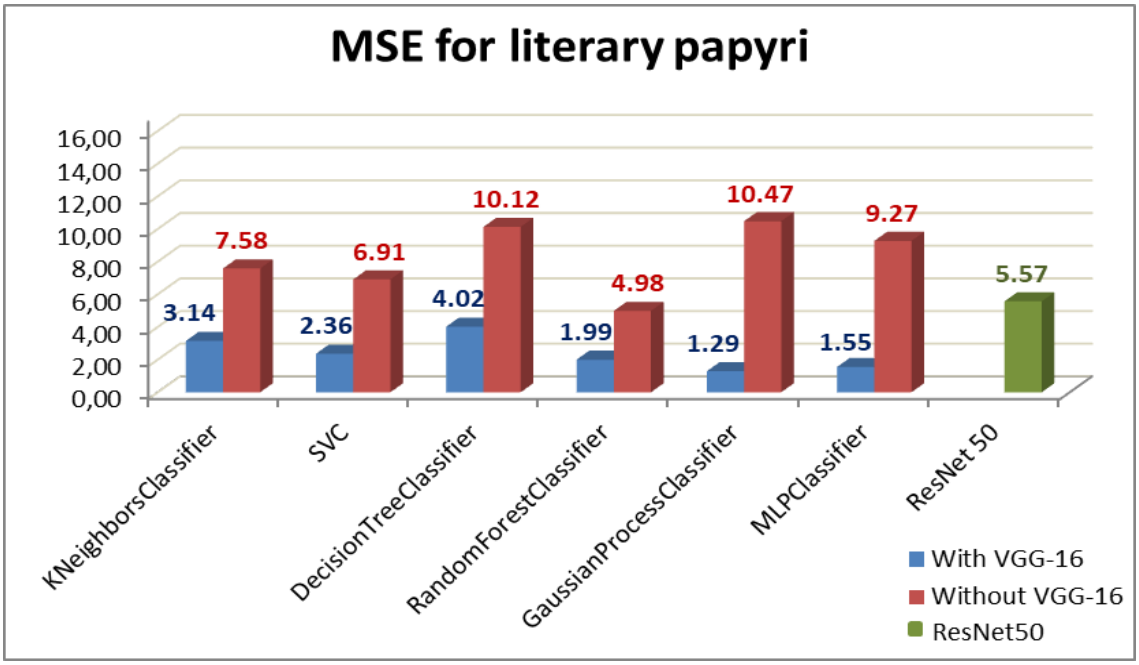

Figure 20: MSE for literary papyri. The numbers represent centuries

Figure 21 presents the Mean Squared Error (MSE) of the models used in our experiments for the documentary papyri. The Gaussian Process and the MLP Classifier with the use of VGG-16 again had the best performances, whereas the Gaussian Process Classifier without VGG-16 scored the highest MSE and, consecutively, had the poorest performance of all the models. On the other hand, the ResNet50 model scored lower MSE error than all models without VGG-16 and the Decision Tree and Random Forest with VGG-16.

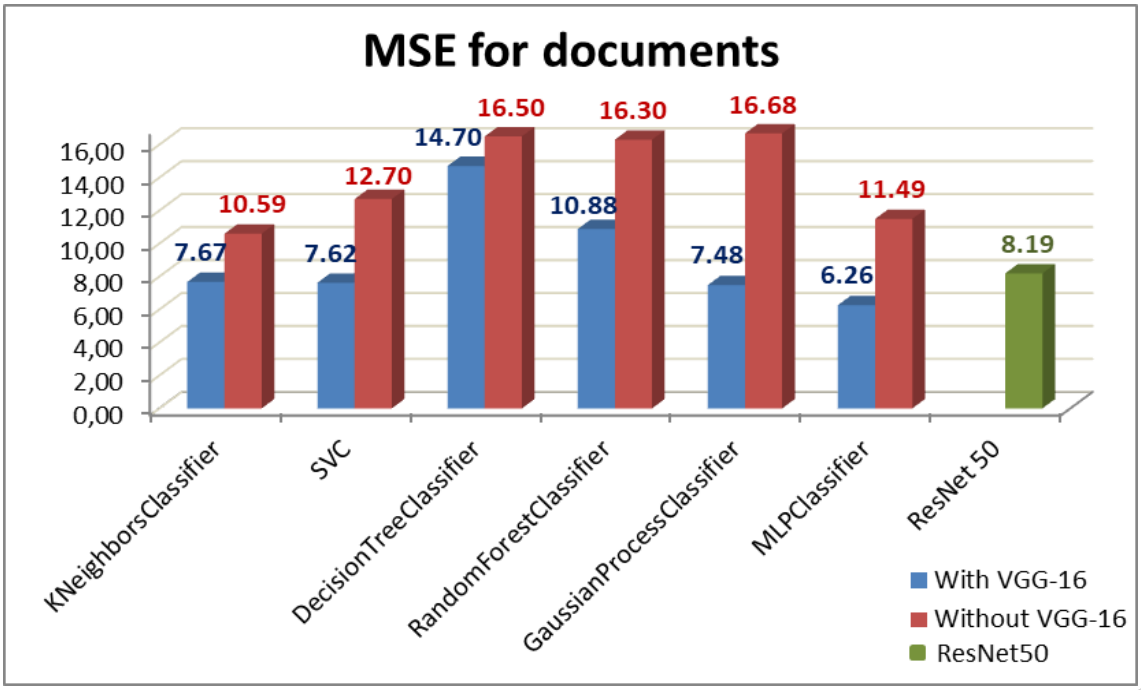

Figure 21: MSE for documents. The numbers represent centuries

8. Questionnaire

8.1. Sample Selection

To compare the estimates of the models with those of the papyrologists, we created a questionnaire (for the questionnaire click here) addressed to people specialized in the field of papyrology, doctoral researchers and experts. The survey involved 1 Ph. D. student and 4 experts, who were asked to estimate the date of a total of 20 papyrus samples (10 literary and 10 documents) without consulting any online or printed source. Respondents were asked to state their level of education, from which derives their degree of familiarity with the papyri, and to give one or more date estimates per sample, along with the method on which they were based for the estimate/s. The three methods of chronological estimation given as options were: the prior knowledge of the papyrus sample, the discovering of a hint on the date in the papyrus text and the recognition of the writing style.

The samples included in the questionnaire were selected as follows: we selected five random samples from the total of 255 images of literary papyri from the CDDGB database and five random samples from the total of 3326 images from the PapPal database.

Then, as we wanted to see the opinion of the experts on the papyri that the best of our models failed to successfully classify (models that employ VGG-16 features), we selected 5 images of literary papyri following the procedure shown in Figure 22: for each of the three test sets of literary papyri, we collected the samples whose date all 6 classifiers failed to estimate correctly. We compared these samples with each other to find common images. As for the literary papyri no common samples of the test sets were found between the failures of the models, we joined the samples (those from each test set whose date all 6 classifiers failed to estimate correctly) and we got five of them at random.

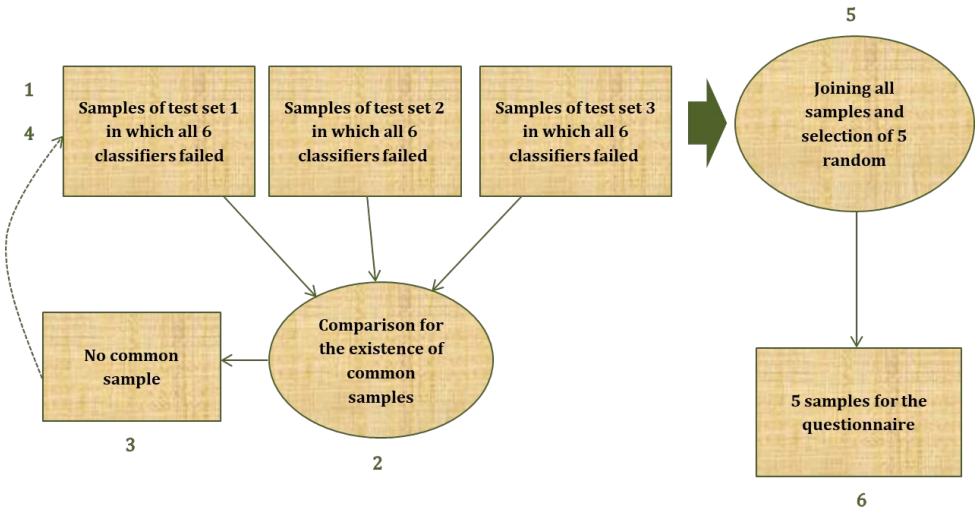

Figure 22: Selection of literary papyrus samples for the questionnaire

Similarly, we selected 5 document images following the procedure shown in Figure 23. For each of the three test sets of documents, we collected the samples whose date all 6 classifiers failed to estimate correctly and, then, we compared the samples with each other to find common images. As 8 common document samples were found among the models' failures, we selected five of them at random.

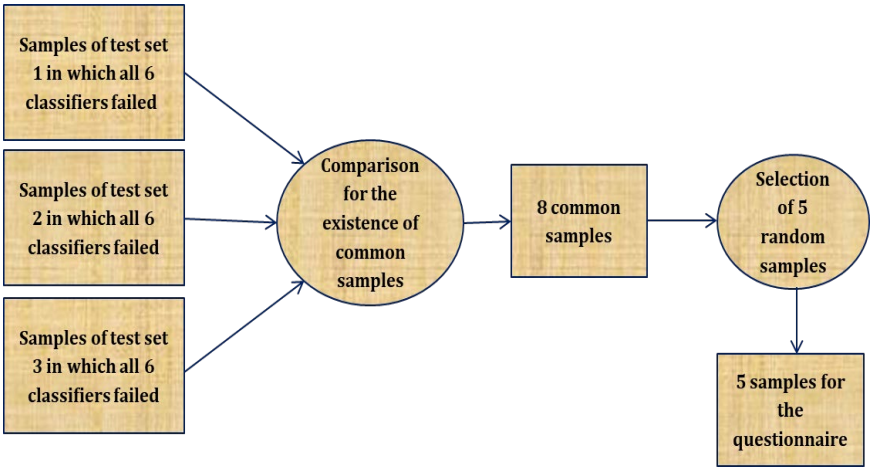

Figure 23: Selection of documentary papyrus samples for the questionnaire

8.2. Results

While collecting the results of the questionnaire, in cases of duplicate answers we decided to apply the following:

·If one of the answers-estimates was correct, we accepted only this one.

·If none of the answers-estimates were correct, we accepted the one closest to the correct century.

Following this, for the ten samples of literary papyri the results of the answers of the respondents are summarized below.

First of all, the average percentage agreement, the degree to which all pairs of respondents agree on average, is only 37%, a score that could be said to some extent confirms the wording for the subjectivity of the method. Table 10 presents the accuracy, F1, MAE and MSE scores per respondent along with their average. What should be mentioned is that Respondent 3 did exceptionally well on the estimates, whereas Respondent 2 scored the lowest scores in all metrics, though the MAE is only 2.1 centuries. The average accuracy of all respondents is not quite high, while F1 is a bit low.

Metrics	Respondent 1	Respondent 2	Respondent 3	Respondent 4	Respondent 5	Average
Accuracy	40.00%	20.00%	90.00%	60.00%	40.00%	**50.00%**
F1-score	23.81%	11.43%	73.33%	42.50%	19.05%	**34.02%**
MAE	1.3	2.1	0.1	1.2	1.4	**1.22**
MSE	3.7	8.5	0.1	4.2	4.4	**4.18**

Table 10: Scores of respondents for literary papyri. On MAE and MSE the numbers represent centuries.

As for the ten samples of documents, the results of the answers of the respondents are summarized below:

Their average percentage agreement is 26% lower than that of literary papyri. The accuracy, F1, MAE and MSE scores per respondent along with their average is illustrated in Table 11. It can be noticed that Respondent 3 had again the best performance with 80% accuracy and a MAE of only 30 years. Nevertheless, the average accuracy and the F1 are not very high.

Metrics	Respondent 1	Respondent 2	Respondent 3	Respondent 4	Respondent 5	Average
Accuracy	20.00%	50.00%	80.00%	60.00%	10.00%	**44.00%**
F1-score	11.22%	43.89%	68.89%	54.76%	4.17%	**36.59%**
MAE	2.8	1.1	0.3	0.7	2.6	**1.5**
MSE	14.2	3.5	0.5	1.9	10.4	**6.1**

Table 11: Scores of respondents for documents. . On MAE and MSE the numbers represent centuries.

8.3. Comparison of the results of the questionnaire with our models

In order to compare the estimates of the respondents for the papyrus samples with those of our models, we decided to train our models on all the samples of the literary papyri leaving out only the ten samples we used in the questionnaire and for which we would get estimates from the models. Similarly, for the documents we trained the models using all the data except the ten samples of the questionnaire. As the best scores for the literary papyri and for the documents were given by the Gaussian Process Classifier and the MLP Classifier, both with the use of VGG16 as feature extractor, we decided to train our models with these two algorithms. The reason why we chose to retrain models was to ensure that none of the samples would be in the classifiers' training set and therefore known to the models.

After training the models and evaluating them in the ten images of the papyri, initially for the literary papyri we found the following: our two models agree 80% with each other, though without scoring high accuracy and F1, as can be observed from Table 12. Nonetheless, their MAE and MSE are quite low, something that indicates that the two models do not fail much in their estimates.

Metrics	Gaussian Process Classifier	MLP Classifier	Average
Accuracy	40.00%	50.00%	**45.00%**
F1-score	36.00%	48.00%	**42.00%**
MAE	1.0	1.0	**1.0**
MSE	3.0	3.2	**3.1**

Table 12: Scores of models on questionnaire literary samples. . On MAE and MSE the numbers represent centuries.

Table 13 presents the results of our models on the ten documentary papyri images. First of all, we should note that the two models agree only 50% with each other. Moreover, we can notice that our model did not perform well in any of the metrics and the MLP Classifier scored lower than the two.

Metrics	Gaussian Process Classifier	MLP Classifier	Average
Accuracy	30.00%	20.00%	**25.00%**
F1-score	20.83%	15.24%	**18.04%**
MAE	2.9	3.5	**3.2**
MSE	14.5	18.1	**16.3**

Table 13: Scores of models on questionnaire documentary samples. . On MAE and MSE the numbers represent centuries.

In Figures 24-27 we present the accuracy, F1, MAE and MSE of each respondent along with the average of all respondents for the literary papyri. The scores are presented in comparison with those of the MLP and Gaussian Process Classifiers, shown with a dark and a light green line respectively (except for Figure 26 where the two models had the same MAE shown with a single line). We notice that MLP scored better accuracy than the majority of the respondents and same to the average, whereas Gaussian Process scored lower than the average. In Figure 25 we can clearly see that both models scored higher F1 than the average of the respondents, and only Respondent 3 performed better.

than MLP Classifier. In Figures 26 and 27 we observe that both MLP and Gaussian Process scored the lowest MAE and MSE of all the respondents with the exception of the third respondent who did exceptionally well in all metrics.

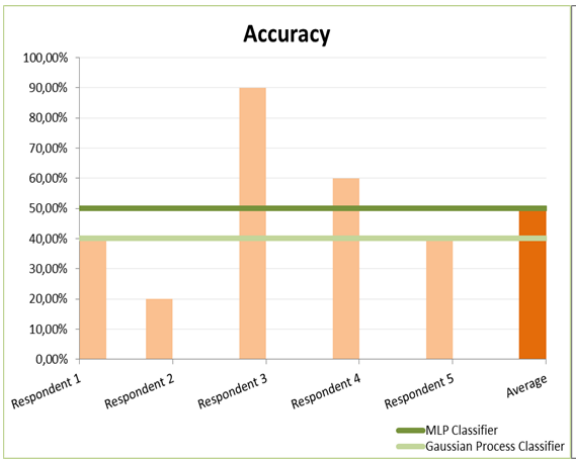

Figure 24: Respondents' accuracy for literary papyri

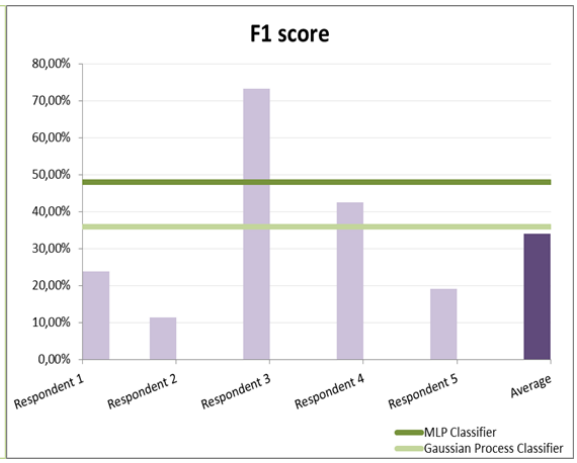

Figure 25: Respondents' F1 score for literary papyri

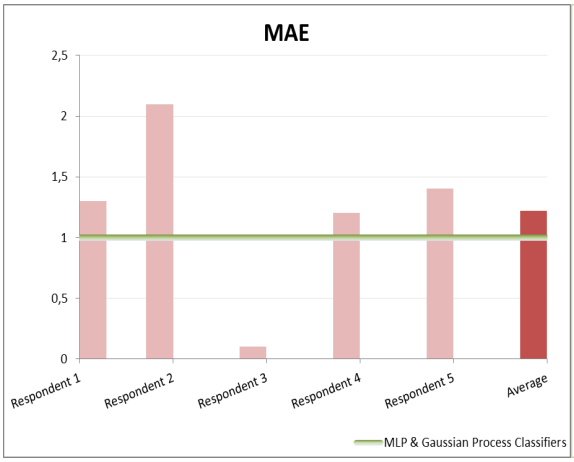

Figure 26: Respondents' MAE for literary papyri

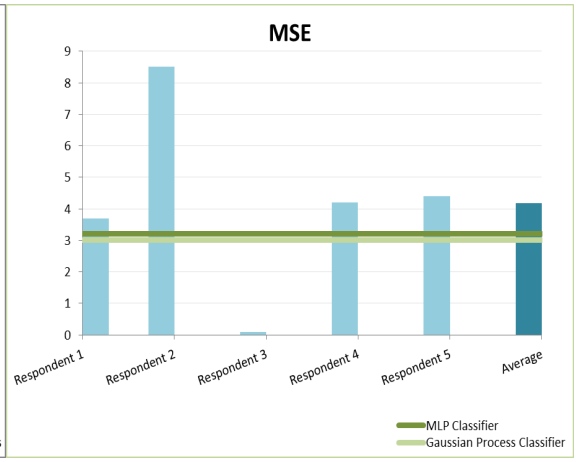

Figure 27: Respondents' MSE for literary papyri

Similarly, Figures 28-31 illustrate the accuracy, F1, MAE and MSE of the respondents for the documents in comparison with those of our models. In this case, we can see that the accuracy and the F1 scores of our models are below the average of all the respondents, while both classifiers scored the highest MAE and MSE of all the respondents. However, we notice that our models did not perform extremely badly, as they exceeded the accuracy and F1 scores of respondents 1 and 5 (MLP exceeded the accuracy of only the fifth respondent and scored the same as the first one), while the MAE and the MSE of the Gaussian Process Classifier are not much higher than the corresponding of the two specific respondents. Finally, we should take into account that in both parts of the questionnaire (the literary and the documentary), there was a small number of estimates made by the respondents based on their prior knowledge of the papyrus under question or a hint on the date in the papyrus text (3 estimates for the literary samples and 5 for the documents).

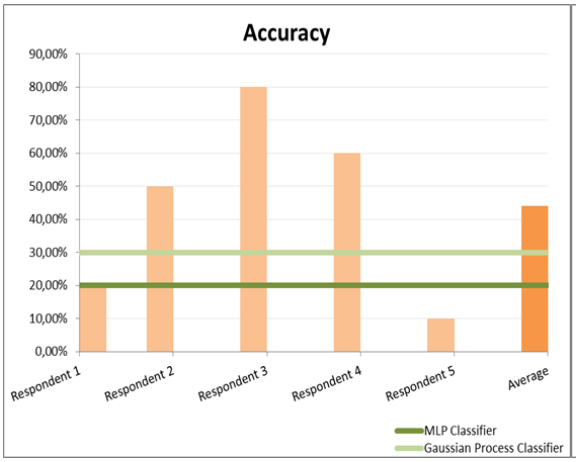

Figure 28: Respondents' accuracy for documents

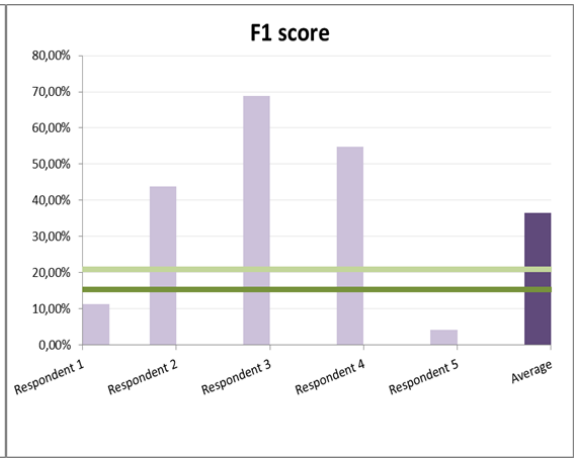

Figure 29: Respondents' F1 score for documents

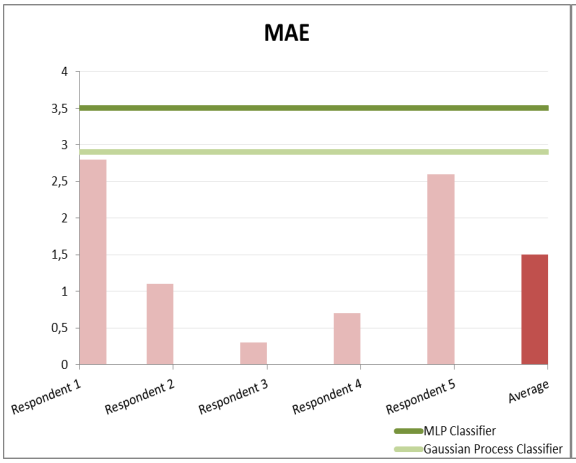

Figure 30: Respondents' MAE for documents

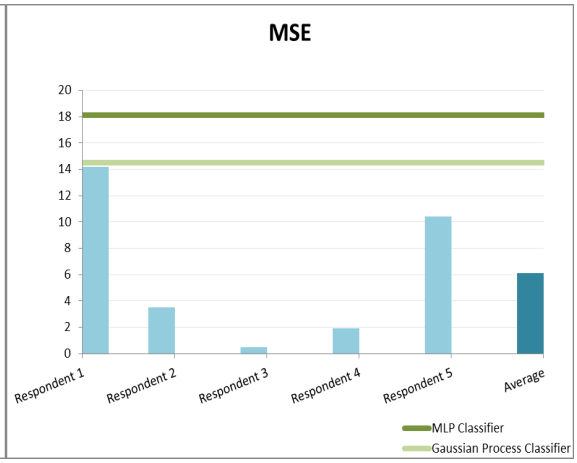

Figure 31: Respondents' MSE for documents

9. Limitations

The results from the experiments we conducted are quite satisfactory (best accuracy score: 69.93, lowest MAE: 0.48) given the limitations and challenges that the task of dating Greek papyri by applying machine learning methods involves.

First of all, the lack of data, i.e. objectively dated papyri that could be used for model training is an important and great challenge that we faced. The decisive role of adding more papyrus samples was clearly reflected in the learning curves (Figures 14, 15), a large part of which indicate high variance that can be addressed with the addition of data.

Apart from the need for more manuscripts, a notable limitation is the imbalance of the existing data. As can be seen in Chapter 3 (Table 2 and Table 3), in which we analyzed our data, the distribution of papyri per century is very heterogeneous, to the point that some centuries have few to almost no samples, while others have a significant representation (1 sample of literary papyrus for the 1st century BC and 2 for the 7th AD while the 2nd has 87 and 29 samples of documents for the 8th century AD when the 3rd BC has over 800). This imbalance naturally affects the results and leads to a poor performance of the models when they are called to date manuscripts of the centuries that have minimal representation from samples.

Another issue that should not be overlooked is the fact that the dates given to the papyri by objective criteria (see the Introduction) is often not entirely precise or accurate, but results from an estimate with an error probability of about 50 years. This means that many manuscripts attributed to a century may, if we take into account the difference of 50 years, be written in the previous or the following century. If such discrepancies have been passed on to the training data and consequently to our models, then we may have a bias problem. In other words, as our models depend on their training data which they consider to be correct and reliable, if the latter contain errors then the models will be trained incorrectly and will fail to give correct estimates.

Finally, a limitation concerns in general Machine Learning and whether it can assist people in their work. The concern has to do with the question of whether any mistake made by a machine learning model is transmitted to man, that is, whether man, in our case the palaeographer-papyrologist, can be influenced by the results of machine learning and based on them be led to an error, which might have been avoided if a model had not been consulted.

Conclusion

In this thesis, we employed machine learning methods with the aim to assist papyrologists in their challenging task of dating papyri. By using data from two online collections of objectively dated papyri, we proposed two machine- actionable datasets that can be used for this task. Our experiments show that the training of machine learning algorithms on the above datasets for their classification into classes of centuries gives good, but not very satisfactory results. On the contrary, the application of deep learning and transfer learning methods had a promising impact. More specifically, the fine-tuned (on a part of our data) ResNet50 model performed quite well but comparably to our first experiments, whereas the use of the pre-trained model VGG-16 with frozen layers and then the training of classifiers gave the highest accuracy and the lowest MAE. Our study shows that our best classifier for documents is the MLP Classifier with 56.76% accuracy and a MAE of 1.32, while for literary papyri the Gaussian Process Classifier with 69.93% accuracy and a MAE of 0.48. Finally, we conducted a research by distributing to experts in the field of papyrology a questionnaire consisting of papyrus samples that had to be chronologically sorted in centuries and by giving our models the same samples for dating, in order to compare the estimates of the experts with those of our models. The results show that in the case of literary papyri our models have a lower MAE than the average expert, and that in the case of documents their performance does not significantly differ from that of some experts.

Our study gives prospects for future endeavours. A particularly interesting proposal would be the study of the results that the chronological attribution of the papyri by experts after the consultation of our models would give. Furthermore, the enrichment of our datasets is considered helpful and necessary for the improvement of our models' performance, especially in papyri dated to centuries that have low representation in samples.

References

Adam, K., Baig, A., Al-Maadeed, S., Bouridane, A., & El-Menshawy, S. (2018). KERTAS: dataset for automatic dating of ancient Arabic manuscripts. *International Journal on Document Analysis and Recognition (IJDAR), 21,* 283-290. https://doi.org/10.1007/s10032-018-0312-3

Bagnall, R.S. (Ed.). (2012). The Oxford Handbook of Papyrology. Oxford University Press.

Baledent, A., Hiebel, N., & Lejeune, G. (2020). Dating Ancient texts: an Approach for Noisy French Documents. *2020 Language Resources and Evaluation Conference (LREC)*. 17-21. Retrieved November 24, 2021, from https://hal.archives-ouvertes.fr/hal-02571633

Briscoe, E., & Feldman, J. (2011). Conceptual complexity and the bias/variance tradeoff. *Cognition,* 118(1), 2-16. https://doi.org/10.1016/j.cognition.2010.10.004

Choat, M. (2019). Dating Papyri: Familiarity, Instinct and Guesswork. *Journal for the Study of the New Testament, 42(1)*, 58-83. https://doi.org/10.1177/0142064X19855580

Dhali, M., He, S., Popović, M., Tigchelaar, E., & Schomaker, L. (2017). A Digital Palaeographic Approach towards Writer Identification in the Dead Sea Scrolls. *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM),* 693-702. https://doi.org/10.5220/0006249706930702

Dhali, M., Jansen, C. N., Willem de Wit, J., & Schomaker, L. (2020). Feature-extraction methods for historical manuscript dating based on writing style development. *Pattern Recognition Letters, 131*, 413-420. https://doi.org/10.1016/j.patrec.2020.01.027

Grouin, C., Forest, D., Da Sylva, L., Paroubek, P., & Zweigenbaum, P. (2010). Présentation et résultats du défi fouille de texte DEFT2010 où et quand un article de presse a-t-il été écrit? 17th Conférence sur le Traitement Automatique des Langues Naturelles.

Hamid, A., Bibi, M., Moetesum, M., & Siddiqi, I. (2019). Deep Learning Based Approach for Historical Manuscript Dating. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 967-972. https://doi.org/10.1109/ICDAR.2019.00159

Hamid, A., Bibi, M., Siddiqi, I., & Moetesum, M. (2018). Historical Manuscript Dating using Textural Measures. *2018 International Conference on Frontiers of Information Technology (FIT)*, 235-240. https://doi.org/10.1109/FIT.2018.00048

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 770-778. https://doi.org/10.1109/CVPR.2016.90

He, S., Samara, P., Burgers, J., & Schomaker, L. (2014). Towards Style-Based Dating of Historical Documents. *2014 14th International Conference on Frontiers in Handwriting Recognition*, 265-270. https://doi.org/10.1109/ICFHR.2014.52

He, S., Samara, P., Burgers, J., & Schomaker, L. (2016a). Image-based historical manuscript dating using contour and stroke fragments. *Pattern Recognition, 58*, 159–171. https://doi.org/10.1016/j.patcog.2016.03.032

He, S., Samara, P., Burgers, J., & Schomaker, L. (2016b). Historical manuscript dating based on temporal pattern codebook. *Computer Vision and Image Understanding, 152*, 167-175. https://doi.org/10.1016/j.cviu.2016.08.008

Kramer, O. (2013). *Dimensionality Reduction with Unsupervised Nearest Neighbors* (Intelligent Systems Reference Library, 51). Springer. https://doi.org/10.1007/978-3-642-38652-7

Lauzon, F. Q. (2012). An introduction to deep learning. *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA),* 1438-1439. https://doi.org/10.1109/ISSPA.2012.6310529

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature, 521(1),* 436-444. https://doi.org/10.1038/nature14539

Li, Y., Genzel, D., Fujii, Y., & Popat, A. C. (2015). Publication Date Estimation for Printed Historical Documents using Convolutional Neural Networks. *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing (HIP '15),* 99–106. https://doi.org/10.1145/2809544.2809550

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, *9(1),* 381-386. Retrieved November 24, 2021, from https://www.ijsr.net/get_abstract.php?paper_id=ART20203995

Marti, U.-V., & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1), 39–46. https://doi.org/10.1007/s100320200071

Mazza, R. (2019). Dating Early Christian Papyri: Old and New Methods – Introduction. *Journal for the Study of the New Testament, 42(1)*, 46-57. https://doi.org/10.1177/0142064X19855579

Mohammed, H., Märgner, V., & Stiehl, H. S. (2018). Writer Identification for Historical Manuscripts: Analysis and Optimisation of a Classifier as an Easy-to-Use Tool for Scholars from the Humanities. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 534-539. https://doi.org/10.1109/ICFHR-2018.2018.00099

Mohammed, H., Marthot-Santaniello, I., & Märgner, V. (2019). GRK-Papyri: A Dataset of Greek Handwriting on Papyri for the Task of Writer Identification. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 726-731. https://doi.org/10.1109/ICDAR.2019.00121

Mueller, J. P., & Massaron, L. (2016). *Machine Learning for Dummies*. Wiley.

Mueller, J. P., & Massaron, L. (2019). *Deep Learning for Dummies*. Wiley.

Nasir, S., & Siddiqi, I. (2020). Learning Features for Writer Identification from Handwriting on Papyri. *Pattern Recognition and Artificial Intelligence, MedPRAI, 2020, Communications in Computer and Information Science, (1322)*, 229-241, https://doi.org/10.1007/978-3-030-71804-6_17

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296), 23–27.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering, 22(10)*, 1345-1359. https://doi.org/10.1109/TKDE.2009.191

Picard, R. R., & Cook, R. D. (1984). Cross-Validation of Regression Models, *Journal of the American Statistical Association,* 79(387), 575-583. https://doi.org/10.1080/01621459.1984.10478083

Rusk, N. (2016). Deep learning. *Nature Methods, 13(1),* 35. https://doi.org/10.1038/nmeth.3707

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR 2015,*1-14. Retrieved November 24, 2021, from arXiv:1409.1556v6

Soumya, A., & Kumar, G. H. (2014). Classification of Ancient Epigraphs into Different Periods Using Random Forests. *2014 Fifth International Conference on Signal and Image Processing*, 171-178. https://doi.org/10.1109/ICSIP.2014.33

Rosten, E., Porter, R., & Drummond, T. (2010). Faster and Better: A Machine Learning Approach to Corner Detection. IEEE transactions on pattern analysis and machine intelligence, 32(1), 105–119. https://doi.org/10.1109/TPAMI.2008.275

Talo, M. (2019). Convolutional Neural Networks for Multi-class Histopathology Image Classification. *ArXiv*. Retrieved November 24, 2021, from https://www.semanticscholar.org/paper/Convolutional-Neural-Networks-for-Multi-class-Image-Talo/f2086b6c8f55c7a4ec8e52f13133ac9abfee8035

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A Survey on Deep Transfer Learning. *Artificial Neural Networks and Machine Learning – ICANN 2018. Lecture Notes in Computer Science, 11141*, 270-279. https://doi.org/10.1007/978-3-030-01424-7_27

Turner, E. G. (1987). *Greek Manuscripts of the Ancient World* (P. J. Parsons, Ed.; Revised and Enlarged ed.). Institute of Classical Studies.

Wahlberg, F., Mårtensson, L., & Brun, A. (2015). Large scale style based dating of medieval manuscripts. *In Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing (HIP '15),* 107–114. https://doi.org/10.1145/2809544.2809560

Wahlberg, F., Wilkinson, T., & Brun, A. (2016). Historical Manuscript Production Date Estimation Using Deep Convolutional Neural Networks. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 205-210. https://doi.org/10.1109/ICFHR.2016.0048

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 31(2), 210–227. https://doi.org/10.1109/TPAMI.2008.79

Wu, E., Hadjiiski, L. M., Samala, R. K., Chan, H., Cha, K. H., Richter, C., Cohan, R. H.,. Caoili, E. M., Paramagul, C., Alva, A., & Weizer, A. Z. (2019). Deep Learning Approach for Assessment of Bladder Cancer Treatment Response. *Tomography*, 5(1), 201-208. **https://doi.org/10.18383/j.tom.2018.00036**

Xing, L., & Qiao, Y. (2016) DeepWriter: A Multi-stream Deep CNN for Text-Independent Writer Identification. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 584-589. https://doi.org/10.1109/ICFHR.2016.0112

Zhuang, F., Qi, Z., Duan, Z., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE, 109(1),* 43-76. https://doi.org/10.1109/JPROC.2020.3004555

Παπαθωμάς, Α. (2016). *Εισαγωγή στην παπυρολογία (*3η Επαυξημένη Έκδοση). Αμφιλόχιος Παπαθωμάς.