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

MSc IN DIGITAL METHODS FOR THE HUMANITIES

Deep Learning-based OCR for Greek Paleographic Manuscripts

MSc Thesis

PARASKEVI PLATANOU

ATHENS, NOVEMBER 2021

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY

DEPARTMENT OF INFORMATICS

MASTER OF SCIENCE

IN DIGITAL METHODS FOR THE HUMANITIES

Deep Learning-based OCR for Greek Paleographic Manuscripts

MSc Thesis

PARASKEVI PLATANOU

Supervisor: Georgios Papaioannou, Associate Professor

Co-supervisor: John Pavlopoulos, Adjunct Professor

Reviewers: Georgios Papaioannou, Associate Professor

John Pavlopoulos, Adjunct Professor

Ion Androutsopoulos, Professor

ATHENS, NOVEMBER 2021

Abstract

Today classicists are provided with a great number of digital tools which, in turn, offer possibilities for further study and new research goals. In this thesis we explore the idea that old Greek handwriting can be machine-readable and consequently, researchers can study the target material fast and efficiently. Previous studies have shown that Optical Character Recognition (OCR) models are capable of attaining good accuracy rates. However, achieving high accuracy OCR results for Greek manuscripts is still considered to be a major challenge. The overall aim of this thesis is to examine the efficiency of OCR software for old manuscript reading and train a deep learning model for this task. To address this statement, we study and use digitized images of the Oxford University Bodleian Library Greek manuscripts. In particular, we follow steps which include image preprocessing, transcription and programming. Our ambition is to go beyond the many challenges we face from one step to the other, taking into consideration that Greek handwritten characters are challenging alone when it comes to machine reading, and develop OCR models using deep learning methods in order to render old Greek handwriting machine readable.

Keywords: deep learning, OCR, handwriting, Greek language, old Greek manuscripts, parchments, Digital Paleography

Περίληψη

Οι φιλόλογοι σήμερα έχουν στη διάθεσή τους μια σωρεία ψηφιακών εργαλείων τα οποία, με τη σειρά τους, προσφέρουν δυνατότητες για περαιτέρω μελέτη και νέους ερευνητικούς στόχους. Στην εργασία αυτή η έρευνά μας βασίζεται στην ιδέα ότι οι παλαιές ελληνικές γραφές μπορούν να γίνουν μηχαναγνώσιμες και εν συνεχεία οι ερευνητές μπορούν να μελετήσουν το υλικό το οποίο ενδιαφέρει άμεσα και αποτελεσματικά. Προηγούμενες μελέτες έχουν αποδείξει ότι τα μοντέλα Οπτικής Αναγνώρισης Χαρακτήρων έχουν τη δυνατότητα να αγγίζουν ικανοποιητικούς δείκτες ακρίβειας. Εντούτοις, η αποτελεσματική Οπτική Αναγνώριση Χαρακτήρων ελληνικών χειρογράφων είναι μέχρι σήμερα μία μεγάλη πρόκληση. Ο στόχος της εργασίας αυτής είναι η εξέταση της αποτελεσματικότητας λογισμικού για την Οπτική Αναγνώριση Χαρακτήρων χειρογράφων κωδίκων και η προπόνηση μοντέλου βαθιάς μάθησης για το σκοπό αυτό. Για να απαντήσουμε στο ερώτημα αυτό, μελετούμε και κάνουμε χρήση των ψηφιοποιημένων εικόνων της συλλογής ελληνικών χειρογράφων της Βοδλιανής Βιβλιοθήκης του Πανεπιστημίου της Οξφόρδης. Συγκεκριμένα, ακολουθούμε μία διαδικασία η οποία περιλαμβάνει επεξεργασία εικόνας, μεταγραφή και προγραμματισμό. Φιλοδοξούμε να αντιμετωπίσουμε τις διάφορες προκλήσεις που συναντούμε κατά τη διαδικασία αυτή, λαμβάνοντας υπόψιν ότι μόνοι οι ελληνικοί γραφικοί χαρακτήρες προσθέτουν εξαιρετική δυσκολία στη μηχαναγνωσιμότητα, και να παρουσιάσουμε μοντέλο Οπτικής Αναγνώρισης Χαρακτήρων με τη χρήση μεθόδων βαθιάς μάθησης με σκοπό να καταστήσουμε τους ελληνικούς χειρόγραφους κώδικες μηχαναγνώσιμους.

Λέξεις κλειδιά: Μηχανική Μάθηση, Οπτική Αναγνώριση Χαρακτήρων, γραφή, Ελληνική γλώσσα, ελληνικοί χειρόγραφοι κώδικες, Ψηφιακή Παλαιογραφία

I would like to dedicate this thesis to my dearly beloved sister, for she has always been there for me, and to my loving parents, for they have given me the greatest gift anyone could give another person; they never stopped believing in me and my dreams.

Acknowledgements

I would like to express my sincere gratitude to Georgios Papaioannou and John Pavlopoulos for their continuous support and invaluable guidance throughout writing this thesis. They provided me with insightful feedback and they were always enthusiastic to assist in any way they could throughout my thesis. I would also like to thank Maria Konstantinidou (Assistant Professor at Democritus University of Thrace) for her comments and suggestions regarding aspects of the manuscript studies. In addition, I am thankful to Georgios Parlavantzas (Digitization and Microphotography Department Director at National Library of Greece) for the fruitful discussions we had in the Digitization and Microphotography department and the chance he offered me to gain a deep insight of the manuscript digitization process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Problem statement

This thesis presents OCR applications to old Greek manuscripts. Today there is a trend for OCR software system development and the community engaged aims to provide the users with the highest accuracy rates possible. OCR targeting printed text material can indeed offer efficient results. However, there is a great difficulty when switching to Handwritten Text Recognition (HTR). Common OCR software systems, which contribute high accuracy rate results to the community, are not capable of recognizing most - if not at all - handwritten characters. An illustrative example of the difficulty level can be seen in the two figures provided below. Figure 1 shows one of the manuscripts of interest in our study. It demonstrates handwritten characters of various length and width, either organized as groups in lines or located outside their group, as well as decoration consisting of a pattern of lines and dots.

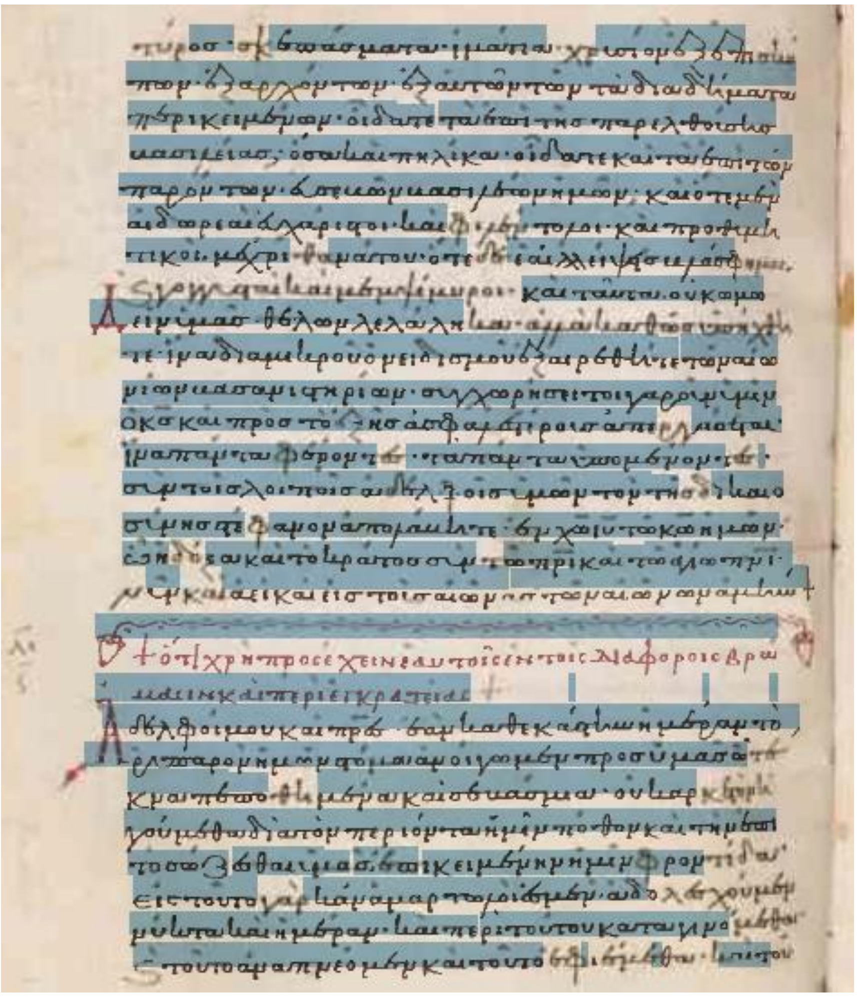


Fig.1: OCR attempt; 11th century AD; MS. Barocci 130 fol.64v recognized by ABBY FineReader 15 OCR Editor. © 2013 Bodleian Libraries, used under a Creative Commons Attribution- Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

This manuscript image (Figure 1) has undergone character recognition and the highlighted areas in blue show exactly what can be machine-read. What the recognized text image can tell us is that programmed machines are not able to distinguish between letters, decorative lines or image noise. It also tells us that OCR systems developed for printed text face extreme difficulty in reading the whole handwritten text. Nevertheless, we should not be deceived by Figure 1 because it only shows us what the OCR system reads. It is Figure 2 that shows us what the OCR system recognizes while reading.

MS. Barocci 130 fol.64v recognized *by ABBY FineReader 15 OCR Editor* Text

' I /· ' * * / '

'ᵀ∧ᴾ ∘ ντιομ΄Υ)^^

^^μ-'ό^ρ/^μτοομ-^'ΐιτ^

'/: “*? * K/ᴵ/ˣ^7-'=°Λ' · οι$\χΓΓΐ Τ^^^Λτησστ^ρ Ι^-^υτ^υτ ^^^Ι^οσ^ί^τΤΗ^ιμ,^ · ο"·<Λα^ΊΓΕ

-7ρχ^Γ

αχό ωρί ολ. <£/^χρ ι £ρ> ι·1«_αχ^) ^ΑΑν τo/λι· χοίχ·7τρο^χμ1^_ ΤΙ^οΐ. Μ-^ζΡ « Ρ «λτ> υ ·ο ΤΕ </^ ίοίι^τ', ■ψζ'ο- ιχιχύχΆ

Ι^ΜΑΧΤα* ^ι^μ^/μ.»ροι- κ αυ^^ου.ₒ όχα» Η»

^,·., £1^ ^ΧΧΧΧΤ'4^.«Ο//z\εΛ<Χ7^Η ί*ΧΜ· Υ„ ο/$««τν^^ίχ

-Τι · | μ αυ^Ι Ο^α Ιι_ρ ο υ ο μ [ί'&'Ί σρ_ο υ ου ρ £-0-ύυτί Τ*μ <ΧΛ ΟΟ

ρΐ°°Ρ “ασηρ Ι ητ-Η ρ ι οα μ · ΓΓΑΛ^» ρ η στ Ιτοί Ζμΐ μ

Ὼ Κ σ"Κ“-* Ή ροσ ·{ ο//4 Ησαω ■ α^_ί^^τ ρ ο νο-αστττ \μαυ7Γα-μ·νσ> (£> έρ ο μ•^ταστπ-ά-μ·-ι-Ου^ί:σΌ^1ί μυμ.ι≤{1 · σ~'-ρ'·το ι σχ\°«·7το«
<Τ<Χ7</Ι^Λ^^ ο ισ^μ^οα μ ·ττα μ «ττίο-^Ι Ιί-Οΐ.0 Γ5Αίγ> κΓΤερ ^<χ^ο^<χζ7πγ_αχ£_ίΛ.·τε · (ίρρ^»(υτοομα> Η ±Λ_α>μ·'

(ίζ> Η^ν^ ^ι^αχ'τη (χρ (χζ^ο-ετ,^ <ιΓ^,7ι^

^~ν Κ*3^ °-ί * κ.0-* ί 1 σ'ΤΟΙΖΓ <χχοο μ α^σ ·ΤΧΛΟ μ ου. οο μ οομ αμΐ-1ΛΜΐι]* ^^ΐθ'τίχρ Μ*7ΓΡθε« χΐΐΝ («νν'τοΓεεΜΤτιίοΛΛά^σροχοβ,ΐΜχι

-Λ -χΛχίχιχχ«^·π-ϊρ»εΓχ^ζ_ΐΐΐ<ί-Τ

}-ΧΧ> V |<αχ ·7τρ <έτ ■ &α~μ (χχχτ^Έ Κ “-Ί ^Α*" Η Η-^/21/7 'ΤΤ> /

.^· * &Λ^ατα^>ο μ ηρί-οομ Εμόμ-αυ αι^/ο »^»ο Ρ-^ί7 *7Τρ °σ^ ρι_αυτ α>'Γ^~' ι

Κ_μ <χ> ~~π Ζ χυ~~ ο -~θ-ίι μ>-ά-μ <υ ;<αχ σ-6-χχζ*-σ-|χχ ασ · ο υ Ιι αρ |<_ίρ Ιλ ί^υμΛβ-ου^αστ-ομ στε ρ >■ ° ρ*ΤΓα-< Η|-Μ μ ·7ΓΟ^&1° μ
1^οΜ.*ΓΗ μ

ΤΌ σχηθ <6-0-ου. νρ-χιχτ/^τίΓί^ει^χ^ρ^Ρ πμ-ΐμ^ί ρομ'τΐ ^α-ί

&ι στοσχο ^•-ρ Ιι ημ η π-ρ τοα^ο ι ^γι^μ·<υί>Ό ^ίτ^Λ υμ^ρ μ ν Ιυτου (χχχχ Η μ-6~ρα.μ· Ιί-ο-ιστΐ ρ ι^τουτον ^αυ-τουγι μο ΜΧ·^ <^το ντο αμ
αστтμε ο μ~^τμ «σα ·το ντο 6τ^) ι ^τμ-^> 0 ' Ιίχυτ'3 ν

Fig.2: Recognized text of Fig.1

It is crystal clear that the system’s capacity for reading demonstrated in Figure 1 is completely different from the one of recognition demonstrated in Figure 2, where the OCR editor fails to recognize the characters appropriately. The main problem seems to be the recognition of the characters as distinct, isolated units. The editor tries to detect symmetries across the text and follow the lines. While trying to detect such patterns the editor makes some guesses by using printed characters with which it has been trained. This is why the result is a sequence of a variety of symbols along with a few alphabet characters. It is also worth noting that slash characters are frequent in the recognized text, which shows the difficulty in understanding the very nature of handwritten characters, that is, not fixed shapes.

The OCR application to handwritten characters has to deal with the several limitations handwriting imposes. In our case, that is, the case of Greek handwritten characters, limitations such as the ones presented below apply.

An interesting fact about handwriting in old Greek manuscripts is that different representations for the same character do manifest themselves. This means that although the machine learns that the source character corresponds to the target character, it is possible that another source character corresponds to the previous target character. For instance, Figure 3 can show clearly this complicated relation between input character and output character. The Greek alphabet character “ν” is the output of both input character images. This fact may have a detrimental effect on accuracy scores since the OCR system is fed with complex information.

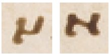
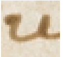
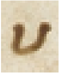
INPUT	**OUTPUT**
	ν

Fig.3: Different representations of the character “ν”. Source; Bodleian-Library-MS-Barocci-102_00157_fol-75r © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

Character visual similarity cannot but be part of the discussion here. Visual similarities between different characters are very common and this fact controls the machine's reading success rate. Figure 4 provides an example of visually similar Greek characters found in the very same manuscript. In the first case the characters “β” and “υ” share to a great extent the same semi-rounded shape, while in the second case the characters “μ” and “ν” do not only share the above mentioned pattern but the vertical line on the left side of the pattern as well.

VISUAL SIMILARITY CASE 1

INPUT	OUTPUT
	β
	υ

VISUAL SIMILARITY CASE 2

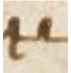
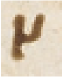
INPUT	OUTPUT
	μ
	ν

Fig.4: Character visual similarity. Source; Bodleian-Library-MS-Barocci-102_00157_fol-75r
© 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

Another major problem is the fact that the way the characters were written was not the same across the centuries. Different writing styles appear through various periods of time. Scholars decided to categorize them and provide them with names reflecting their main characteristics. That said, there are a variety of identifiable minuscule styles. For example, the script which demonstrates characters stringing together because of the presence of a great number of ligatures (mid. 10 th c. AD - 12th c. AD) is called Perl script because this stringing reminds of a necklace made of pearls [1]. The Beta Gamma script (mid.13th c. AD - mid. 14th c. AD) owes its name to the fact that those two letters are prominent [1]. The Bouletée script (10th. c. AD) is large and rounded [1]. These scripts constitute only a few examples of the script categories available.

However, the most significant problem of all is the very nature of the handwritten character which is completely different from the one of the printed character. Their difference is based upon the fact that the printed character is required to meet certain expectations, which are conventions made by the community as a result of how the machine is able to output information. Contrastingly, the handwritten character is neither standard nor fixed. It consists of irregular lines and it reflects again conventions made by the community of practice, which dictate that one is free to draw the character shape in the way one wishes as long as it is closer to the target character than to other glyphs. In consequence, this means that characters can be different even when it comes to the very same writing style. In other words, the scribe of a manuscript presenting Perl script is highly probable to create characters which are slightly different from the scribe of another manuscript presenting the same writing style. This is because writing styles are human conventions based on average characteristics without examining the particularities demonstrated on each and every manuscript. Each manuscript is unique and this fact complicates character recognition for the machine.

Different handwriting across the old Greek manuscripts is also due to the writing tool used. This adds to the machine's difficulty of recognizing characters since in this day and age the writing tools in use do not present the common characteristic of the writing tools of those times, that is, ink welling from the writing tool and spreading all over the folio area of use. The common writing tool scribes of the middle ages used when writing on parchment or papyrus is named as *calamus* or *canna* (*κάλαμος*, *γραφίς*) [2]. *Calamus* was a sharpened reed used for writing in ink. Apart from *calamus*, *penna* (*κονδύλιον*, *κόνδυλο*), which is made from a sharpened and split feather, was also popular among scribes for writing in ink from the fourth century onwards [2]. As a result of the fact that the scribe dipped the writing tool in the ink, the characters written would receive a great amount of ink and probably be smudged. Even if this is not the case, there is no doubt that the scribe would struggle to write as many characters as possible before the ink is gone and this means continuous writing. This is completely different to today’s handwriting because ink is stored in the pen and if one is not in a hurry, handwriting can be much more clear and coherent. However, it is not just the quantity of the ink one should take into account but also the quality. The ink was made from metallic infusions and other substances, among which was vitriol [3]. The presence of certain amounts of each and every chemical substance plays a decisive role in handwriting. Should the amounts be different than what is expected, the ink may either fade over time or seep through the parchment and sometimes result in parchment piercing.

Understanding handwriting of earlier periods of time is challenging itself but it can be even more challenging if we take into account the practice of handwriting of those times. Figure 5 illustrates the common practice of these times. According to these times’ trend, scribes would sit on a bench or a stool, with the manuscript laid across their knees [4]. The habit of writing on a table takes place at a later time and there are thus several cases in which the exemplar is open on the desk before the scribe who copies the text on the manuscript he holds [5]. A number of factors related to this practice such as fatigue may exert considerable influence on handwriting.

Fig.5: Illustration of the Evangelist Mark as a scribe; Walters Ms. W.531, Trebizond Gospels fol. 60r. Mid 12th century AD.
Source: www.thedigitalwalters.org

1.2 Thesis goals

This thesis focuses on the exploration of the various possibilities offered by Optical Character Recognition. We are interested in text recognition and perception of the human written language by the non-human eye. Our goal goes further beyond the text recognition by the emphasis shift on the handwritten text form instead of the printed one.

State-of-the-art tools have been already designed in order to provide their users with decent OCR results. In this thesis indicative use of such a state-of-the-art tool takes place. As we proceed with an analysis of the results of the tool, the following question arises Q does a state-of-the-art OCR system have the potential to adequately meet the expectations of a state-of-the-art OCR system and in consequence, could we go beyond the state of the art? It is widely accepted that these models achieve good, although not excellent, success rates, even when it comes to handwritten text recognition.

We are interested in evaluating state-of-the-art model performance and investigating the resources needed for calibrating. Given the fact that such a model is provided with the necessary resources it would be interesting to explore the degree to which state-of-the-art models can compete against machine learning methods. For this reason, we create a new dataset, test state-of-the-art model performance on this dataset, evaluate results per century and then, decide on the best data structure possible taking into account the state-of-the-art performance evaluation.

The thesis is organized as follows; first, we present recognition results using state-of-the-art OCR software. Next, we provide details for our approach and then we evaluate our method against the reference OCR software.

Our study aims to show that there are alternatives to the already developed OCR models and that it is possible to work even on data which may seem challenging. There are a variety of factors contributing to the problematic aspect of the data and what we aim to show is that taking these factors into consideration will enable efficient working.

1.3 Literature review

When it comes to the decision on which OCR approach to use, there are usually two methodologies to which scholars tend to resort; the segmentation-based approach and the segmentation-free one.

1.3.1 Segmentation-based OCR

The first approach involves segmentation at line level, word level and character level. The use of Convolutional Neural Network (CNN) seems to be a popular solution. Nirmalasari et al. [6] experimented with the NIST Special Database 19 as a training dataset and a handwritten text on screen as a testing dataset. In the first place, a lexicon CNN model was fed with words frequently appearing in the text and then was used to recognize them. In this system, when the model failed to recognize a word, two other models, the Character Count CNN and the Character Prediction FCN (Fully Convolutional Network) would be activated. The former consists of classes of characters to be recognized in the text as well as images of these classes. The FCN model is fed with character images from NIST Special Database 19 in order to recognize the word by reading character by character with the aid of a sliding window. The highest accuracy rates of the three models used is 99.98%, 98.56%, and 83.52%, respectively.

Balci et al. [7] adopted a different approach which enabled them to make use of both word classification and character segmentation. The IAM Handwriting Dataset was used as training and testing dataset. Preprocessing involved padding and rotating images and zero-centering data with the aid of the dataset mean pixel values. A word classifier was built and was trained with the following CNN architectures for deep learning; VGG-19, RESNET-18, and RESNET-34. As far as character segmentation is concerned, the Tesseract 4.0 neural network-based CNN/LSTM engine was used after the appropriate adjustment involving the creation of a character dictionary. However, the final errors cannot be attributed to either of the models, the segmentation one and the character classification one, because they were trained separately.

Vamvakas et al. [8] used two handwritten character databases (CEDAR and CIL) and two handwritten digit databases (MNIST and CEDAR) for character classification based on subdivisions of the character image. Firstly, image binarization and resize take place. Image subdivision is to follow and once feature vector extraction is completed, each and every of the features is scaled to [0,1]. In the recognition phase, classification was performed with the use of Support Vector Machine (SVM) in conjunction with the Radial Basis Function (RBF) kernel. Accuracy rates were among the highest at the time of publication.

Haviluddin et al. [9] analyze image segmentation by referring to their approach to Buginese Lontara script from Makassar; the Vector Quantization Technique. According to their method, image segmentation takes place and results in nine segments and IoC is to count the number of black pixels included in each segment. The use of the Learning Vector Quantization (LVQ) Method is similar to the Back Propagation Neural Network (BPNN) one, consisting of the input layer, the hidden layers and the output one. The highest accuracy rate reaches 66.66 %.

1.3.2 Segmentation-free OCR

The segmentation–free approach is popular as well and significant experiments have been carried out based on this approach, which does not involve segmentation at character level but rather focuses on word detection. Ntzios et al. [11] applied this method to a collection of Old Greek handwritten manuscripts, which the St. Catherine's Mount Sinai Monastery hosts. The reason why they adopted the aforementioned method is because of the very nature of the script used for recognition. The great number of ligatures and the continuity in writing led them to work on word level and detect characters depending on the demonstrated cavities. In the first place, image binarization, enhancement and skeletonization take place. Skeletonization enables the use of the appropriate algorithm which can in turn detect the open and closed cavities of the characters. Cavity detection is of interest in the case that the width of the cavity is greater than the 1/3 of the mean width of all the cavities and in the case that it does not include open and closed cavities sharing common boundaries. Each cavity is then part of a bounding box with top – left and bottom – right coordinates. The feature estimation is realized through a vertical and a horizontal mode, that is, a kind of top – bottom and left – right side, respectively, scanning of the protrusible segments of each feature – character. These segments tend to belong to more than one character and thus, the upper and lower left most and right most bounding boxes of the open cavities are not estimated. The pixels are marked and are not considered in the next estimation phase. Cavity merging is then expected to take place in the case of at least two closed cavities sharing a common boundary and the result may be a character or a character ligature. It is then determined whether it belongs to the category of horizontal characters or vertical characters on the basis of the minimum and maximum y–coordinates it bears. Open cavity merging takes place once there are no protrusible segments on the top and the bottom of the features. The character classification is all the more important to the character recognition and thus, a dictionary of open and closed cavity patterns is compiled, that is, 3886 characters and character ligatures in total. Once the algorithm is applied, average precision for each of the characters is 88.85% and recall

90.74%. The described method indeed gives accurate results to a great extent but it cannot still solve the problem of character recognition in the various old Greek writing styles.

Except for the Greek characters, Chinese character recognition has been of interest to scholars as well. Messina and Louradour [12] use Multi-Dimensional Long-Short Term Memory Recurrent Neural Networks (MDLSTM-RNN) in order to recognize lines of handwritten Chinese text and show no interest in character segmentation. The training and testing data can be found in the CASIA Off-line Chinese Handwriting Databases. The MDLSTM-RNN model receives as input an image of a text line which is multiply scanned including different directions. Connectionist Temporal Classification is deemed as a necessary method to match the sequence of network outputs for each image with the sequence of characters in the transcribed text and therefore, to skip character segmentation.

More recently, Yousef and Bishop [13] proposed the OrigamiNet, a NN module that can convert a single line recognizer trained with CTC into a multi – line recognizer and can unfold 2D input signals into 1D without information loss. This conversion is realized through up-scaling vertically and at the same time down-scaling horizontally, in two stages. The method achieved state–of-the–art results for full page recognition at the time of publication.

Another interesting approach, though not being extremely relevant to our work but still sharing useful ideas for OCR improvement, is the one focusing on OCR post-hoc correction. Lyu et al. [14] use a recurrent (RNN) and a deep convolutional network (ConvNet) in order to correct errors found in the already recognized text. Input text consists of transcribed texts of historical books in German Language from the 16th–18th centuries provided by the *Austrian National Library* (OeNB). The *Deutsches Textarchiv* (DTA) collection's transcriptions serve as ground-truth transcriptions, while the Google Books project provides automated and consequently erroneous transcriptions. Errors are categorised into three types, namely, over-segmentation, under-segmentation and word error. The neural approach is based on an encoder-decoder architecture according to which the encoder achieves representation of non-corrected input text at character level with the aid of RNN and deep ConvNets while the decoder achieves character error correction thanks to an RNN model. WER rates reach 82% and 89%.

What is worth mentioning here is the fact that although there has been a growing tendency towards Greek NLP, which includes OCR techniques, in research, a survey shows that scholars prefer working with modern Greek than ancient and old Greek [15]. Table 1 provides information on the number of research papers published over the last decades. Papers addressing OCR techniques for old Greek manuscripts are only a few according to our own online research and for this reason there is a need for further studies related to the ancient and old Greek handwritten text recognition.

Period	**Modern Greek**	**Ancient Greek**	**Dialects**	**Total**
[1990-2000]	7	-	-	7
[2000-2010]	15	1	-	16
[2010-2015]	9	2	-	11
[2015-2020]	50	11	4	61
TOTALS	79	15	4	99

Table 1: Greek NLP papers from 1990 to 2020. Source: taken from [15].

Chapter 2: Dataset Description

2.1 Source

The images of folios used in our study can be freely browsed from the website of a wide digitization project generously supported by The Polonsky Foundation. This Polonsky Foundation Digitization Project was carried out between 2012 and 2017. This major project is a collaboration between the University of Oxford Bodleian Libraries and the Biblioteca Apostolica Vaticana. The digitized collections include early printed books, Greek manuscripts, Hebrew manuscripts and Latin manuscripts. The digitized collection used in our study is the Greek manuscripts one. The Barocci collection consists of 244 volumes and it is the largest acquisition of the Bodleian collection. It owes its name to Francesco Barocci (1537-1604) whose grandson inherited and later sold the collection to the library in 1629.[1]

The dates of these manuscripts range from the 8th century AD to the 17th century AD. In our study we decided to categorize manuscripts into seven groups based on the century to which they date. This means that we are not interested in manuscripts belonging to more than one group. Table 2 enumerates the members of each group providing the identification number the respective digitized items bear. The total number of manuscripts grouped is one hundred and ninety.

CENTURY	MANUSCRIPT (MS) BAROCCI ID NUMBER
10 AD	50.1, 50.2, 184, 199, 238, 242
11 AD	77, 102, 128, 130, 163, 185, 186, 196, 210, 229, 230, 237
12 AD	15, 123, 132, 138, 143, 144, 182, 190, 222, 225, 228
13 AD	11, 16, 17, 18, 23, 24, 30, 31, 99, 118, 122, 131, 157, 177, 188, 215, 220, 234
14 AD	4, 5, 20, 27, 28, 56, 69, 73, 79, 89, 91, 100, 101, 103, 110, 120, 127, 129, 136, 137, 139, 141, 156, 172, 193, 195, 197, 219, 227, 241
15 AD	1, 6, 7, 9, 13, 19, 32, 34, 35, 38, 39, 41, 43, 45, 46, 48, 51, 52, 53, 54, 58, 59, 60, 61, 62, 64, 68, 70, 71, 72, 75, 76, 78, 80, 81, 82, 83, 84, 85, 87, 88, 90, 94, 95, 97, 98, 104, 105, 106, 109, 111, 112, 113, 114, 115, 119, 124, 135, 140, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 158, 159, 160, 161, 162, 165, 166, 168, 169, 171, 175, 179, 194, 211, 223, 224, 226, 231, 232
16 AD	8, 33, 36, 37, 40, 42, 44, 47, 49, 65, 66, 67, 74, 92, 93, 108, 117, 125, 170 (Latin and Greek), 176, 178, 187, 189, 200, 212, 213

Table 2: Oxford University Bodleian Libraries Greek Manuscripts Periodization

---

[1] Retrieved November 30, 2021, from OX Bodleian Library statement on Greek collection acquisition,

To our knowledge, there is not an online available OCR contribution concerning the Barocci collection. OCR works with other Bodleian collections involve handwritten text recognition in historical manuscripts[2], the Bodleian Library's Book of Curiosities Project[3] and Letter identification in tremulous medieval handwriting with the aid of an ensemble of evolutionary algorithms[4].

2.2 Our dataset

Our dataset consists of three incrementally built editions of data. The first edition involves no segmentation whereas the next two editions are the result of careful segmentation. The second edition involves line segmentation while the third edition involves character segmentation. That said, the three editions comprise folio images processed and thus, of different size. Image size also varies among the images of the last two editions.

V 1.0 (page/image)

Our first edition consists of an image dataset of 100 items. Each image displays the folio and hence text, while it may also display other items necessary for manuscript digitization. These items may include a ruler as well as tools enabling the manuscripts to stay open. Apart from external items, characteristics other than the text itself may also be demonstrated on the image. These characteristics concern the manuscript and may be associated with either its production or its condition. Page numbers, dust and dirt are some of the usual characteristics.

The text includes Greek characters and can be divided into two categories which are the following ones; easy to read and difficult to read text. The latter is a number of pages written in a writing style that enables continuous writing and this is why writing style in some pages demonstrated in the images is not easily comprehensible. This is prominent in the manuscript pages dating to the last two centuries of the time span set in our study.

The 100 images are grouped into seven categories according to the date of the manuscripts. The 10th century group comprises ten images, the 11th century group ten images, the 12th century group ten images, the 13th century group ten images, the 14th century group ten images, the 15th century group ten images and the 16th century group forty images. The forty images belong to two different manuscripts. That said, images of eight digitized manuscripts in total are used in our study.

---

[2] Retrieved November 30, 2021, from Easily Adaptable Handwriting Recognition in Historical Manuscripts

[3] Retrieved November 30, 2021, from The Bodleian Library's Book of Curiosities Project

[4] Retrieved November 30, 2021, from Investigating the use of an ensemble of evolutionary algorithms for letter identification in tremulous medieval handwriting

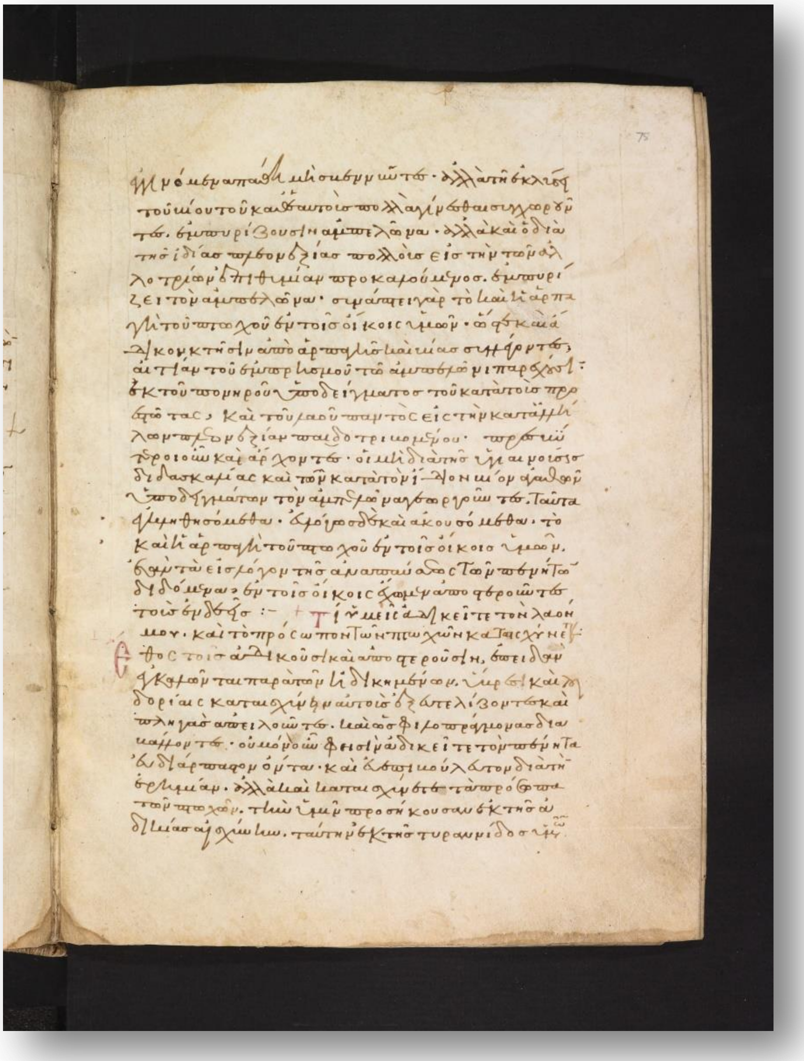


Fig.6: V 1.0 sample. © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

V 2.0 (line/image)

The second edition is based on our first edition. We examine carefully the first edition's images and we select the images that follow specific criteria. We are in particular interested in images which demonstrate dirt and dust-free pages as well as characters set in horizontal upright position. After image selection we segment five text lines in each image. Two images from each century group are used for this task. Before we were to embark on our project, we established a list of guidelines for segmenting our lines.

Line Segmentation Guidelines:

1. Segment manually using the screenshot option.
2. Crop and keep the area of interest only in line shape, that is, let aside characters violating the rule.
3. Bear in mind that the line length depends on the position of the first and last character in each line.
4. Breaths, accents, stresses and punctuation marks are not of interest. Unless they form part of our line shape, we ignore them.
5. The upper and lower points of the segmented line result from the estimation of the closeness of the line to its neighboring ones. No parts of above and below lines should be included in the segmented line

Having taken the above mentioned line segmentation guidelines into consideration, we succeeded to segment 1,906 lines in total. The new edition consists of images of continuous writing style as well. In this way, the major difference between the first and second edition is the fact that while we proceed with the OCR task, segmentation will have already taken place in the second edition's data, whereas automated text layout analysis is expected to take place in the first edition's data.

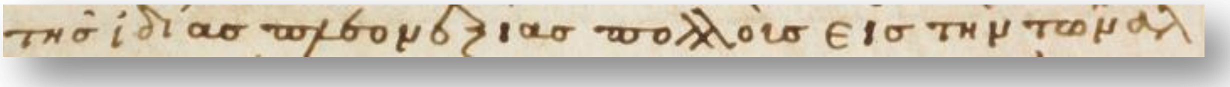


Fig. 7: V 2.0 sample. © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

V 3.0 (character/image)

The third edition is an image dataset of 2,291 characters. It is actually a new edition based on the previous one. The already segmented lines serve as the basis of the segmentation process this time. The characters forming the text lines are considered separately and the result is a total of 2,291 images of various sizes. The segmentation guidelines established for editing the images are the following ones.

Character Image Segmentation Guidelines:

1. Use the V 2.0 data.
2. Segment manually using the screenshot option.
3. Crop and keep the area of interest only in the shape provided, that is, let aside parts of characters located out of the area of interest and consequently, out of the already segmented line.
4. Breaths, accents, stresses and punctuation marks are not of interest. Unless they form part of the character's rectangular shape, we ignore them.
5. The upper and lower parts of the segmented shape result from the estimation of the closeness of the depicted character to its neighboring ones. No parts of other characters should be included in each one of the segmented character images.

Apart from letters, other characters were also included during the segmentation process. These characters may be punctuation related characters such as the comma or the question mark or even empty characters.

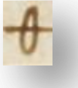


Fig.8: V 3.0 sample. © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

2.3 Criteria for data selection

The selection of the particular image dataset results from the need for controlling groups of data and minimizing other factors’ effect on the results of our study. This is the reason why we established specific criteria satisfying our goals and promising encouraging results. In this section we explain briefly our decisions on language, script and digitized manuscript collection selection.

In the first place, we decided to work with Greek characters because as one can see from the previous research in the field there is a preference in working with Latin characters, which are common in most European languages, than Greek characters. This adds to the difficulty of our work because there is not sufficient guidance on creating successful machine or deep learning OCR models suitable for handwritten Greek characters available. Nevertheless, we believe that such an endeavor can prove to be useful for future projects.

The specific image dataset is used due to the fact that the respective folios serve as characteristic examples of the writing style they represent. They include both Greek minuscule script and the cursive style of the minuscule script. In this way, our work involves examination of different styles and is not limited to one style of script. This enables us to draw conclusions on machine reading ability demonstrated in different styles.

Another extremely important aspect of the dataset selection aspect is readability. The images selected display text information in a clear way. In most cases the text can be read with no previous extensive knowledge of Greek paleography. In this way we provide the machine with easy to read data and we will be able to investigate whether the difficulty in reading on behalf of the machine is because of handwriting of the times concerned itself or other factors. Readability is not only helpful for the machine but also for the human annotation task. In other words, the particular images have also been selected because they are easy to read and thus, easy to transcribe. Apart from the fact that transcription of these folios seems as an easy to do task there is the chance of validating the transcription as well and this is one more reason for data selection. Transcription validation is possible through the various online databases on condition that the transcriber can read sufficient text information.

However, we have to admit that digitization itself plays a major role in data selection. Unfortunately, one does not have the opportunity to get access to a vast corpus of digitized manuscripts from different collections. This is due to the fact that there are not a great number of digitized collections freely available online. This sets a limit on the choices we have. Collection watermarks or other digital signs on the digitized images are also to account for the ineffectiveness of the respective images being recognized. The dataset we selected provides us with the opportunity to study and use the material for personal and non-commercial purposes under the terms of the UK Creative Commons ‘Attribution-NonCommercial- ShareAlike 3.0’ Licence (CC-BY-NC-SA).

2.4 Data characteristics

The digitized manuscript texts share a significant number of common characteristics. Table 3 provides examples of these texts from the second edition created for our study. The images appear next to the century they date.

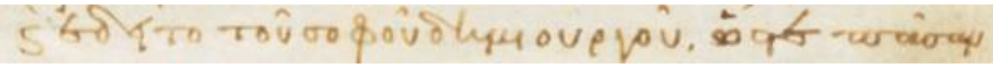
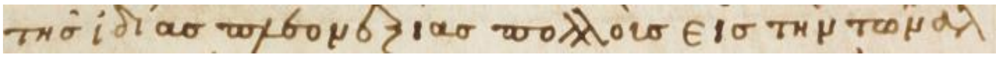
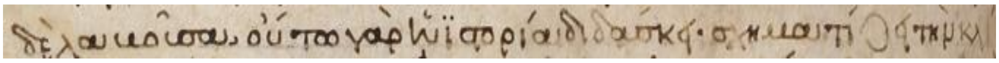
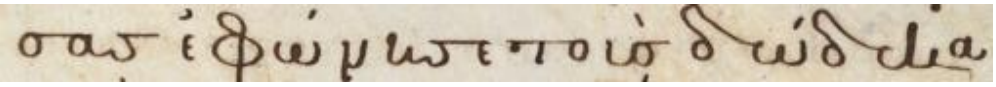
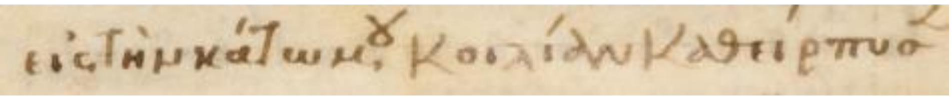
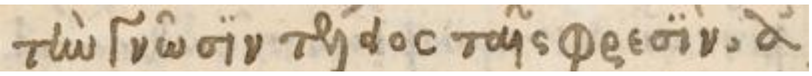
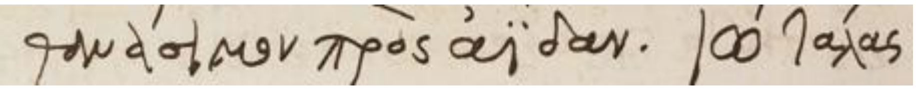
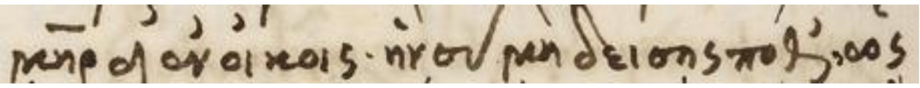
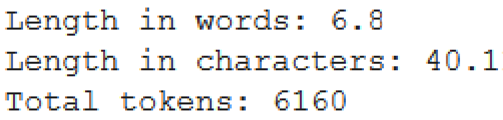
CENTURY	**MANUSCRIPT (MS) BAROCCI**
10 AD	
11 AD	
12 AD	
13 AD	
14 AD	
15 AD	
16 AD	 

Table 3: Script examples used in our study © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

One can easily tell from Figure 9 what the characteristics of our data are. At first, there is the distinction between minuscule clear writing style and cursive style which one can see in the sixteenth century's example. The characters are of various sizes while it is possible that there is size inconsistency in the same character group. One can also notice that the characters are not always grouped together following one another in the line but they can be placed above or below their group which is the case of the fourteenth century's example in Figure 9. When in a group, they may join each other and form ligatures. Ligatures are more or less frequent and appear in manuscripts dating to all centuries concerned as one can tell from Figure 9. Another characteristic of the text is that, although the script is lowercase, in many cases the text includes both uppercase and lowercase characters. This fact confuses machine learning models because they need to be trained for both uppercase and lowercase characters found in the train text and afterwards produce the same output for each character, that is, a lowercase character because we need to take into consideration that the input script is lowercase and this is what the output script is desired to be. Moreover, since we undertake the task of working with Greek scripts we are to encounter other characters apart from letters. In other words, it is also characteristic of these texts to include breaths and accents. These marks may be placed over the associated syllable or even further in the text. The fact that they are not often aligned with the corresponding letter renders the marks difficult to read for the machine, while handwriting is a contributing factor in machine reading inability. For these reasons we decided that we will not transcribe them although we admit that their inevitable presence in our many cases of our editions may come at a cost.

2.5 Statistical analysis

Once we have prepared our data and the respective files including transcriptions we can experiment with a statistical analysis of our data.



```
Length in words: 6.8
Length in characters: 40.1
Total tokens: 6160
```

Fig.9: Data statistics

First, we look at quantitative data. We examine the numbers that best describe our data. Figure 10 shows the results of this task. What we see in Figure 9 is the average length in words and characters in every line. This means that each text line is approximately seven words long and approximately forty characters long. In this way we can describe the length of our text lines. Searching through the very text we get the number 6.160 which is the total number of the tokens found in our text.

After we have described our data at shape level, we proceed with questions regarding the content of our data. This time we need to preprocess the text. This involves lowercasing our characters and removing breaths, accents and punctuation marks in order to take accurate results. Should we not preprocess the text, the word *καί*, for example, would appear two times in our results because it is also written as *καὶ* and therefore, we would get different numbers for the same word and this is not our goal. Part of the result of preprocessing can be seen in the word cloud in Figure 10. Once preprocessing is complete, we can look at word frequency and find words which appear several times in the text. We limit the most frequent words to the first forty ones. Figure 10 shows this top word frequency.

.

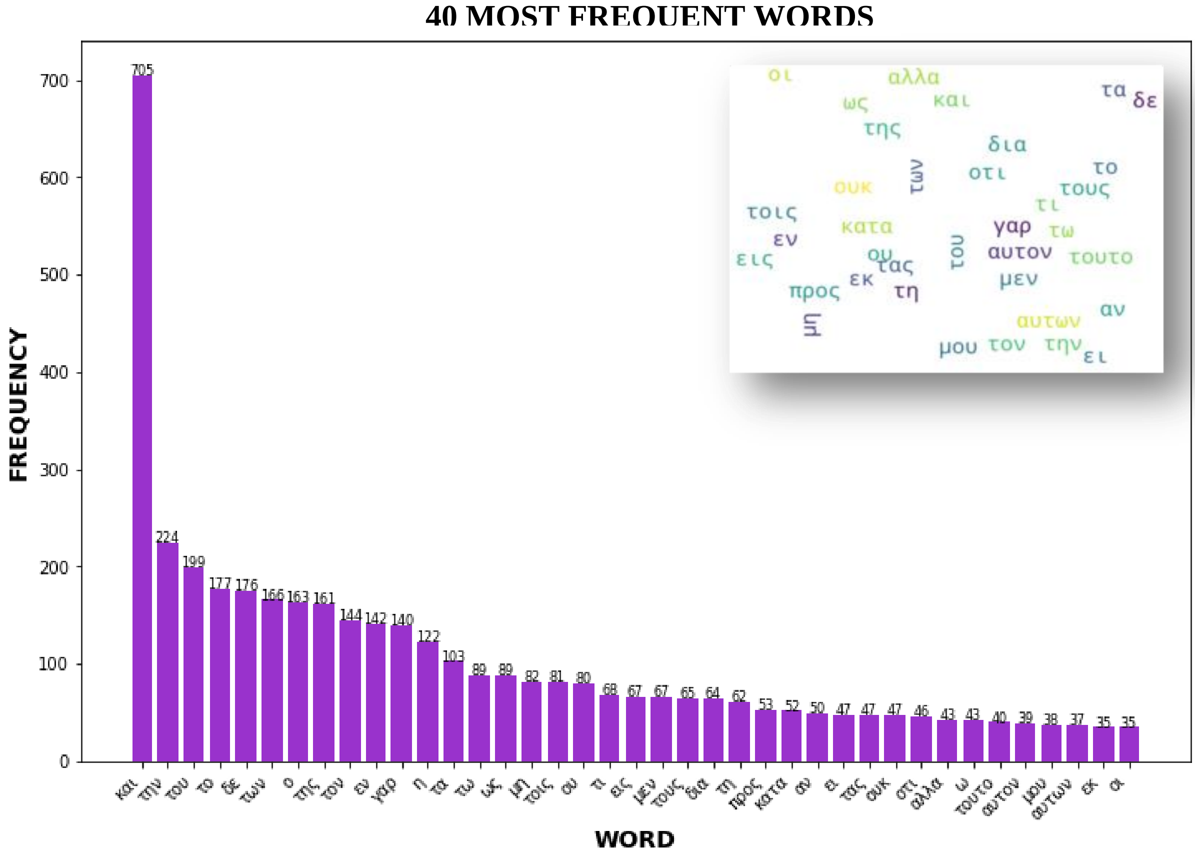


Fig.10: Word frequency in our preprocessed dataset. In the upper right corner Wordcloud of preprocessed data is provided.

As we can see in Figure 10, most frequent words are monosyllabic words. The conjunction *και* is the most frequent one with seven hundred and five examples. Other frequent words include articles, other conjunctions, pronouns, negation and prepositions.

However, we are still not able to tell anything about the content of our text. Thus we go back to preprocessing and edit the word list. This time we create a list of stopwords. Based on the previous results demonstrated in Figure 12 we remove as many conjunctions, prepositions, articles and negation as we can from our final word list. Part of the results can be seen in the word cloud in Figure 11. After editing our data, we can look again at word frequency and aspects of the text content. Figure 11 provides us with the forty most frequent words in the clean text.

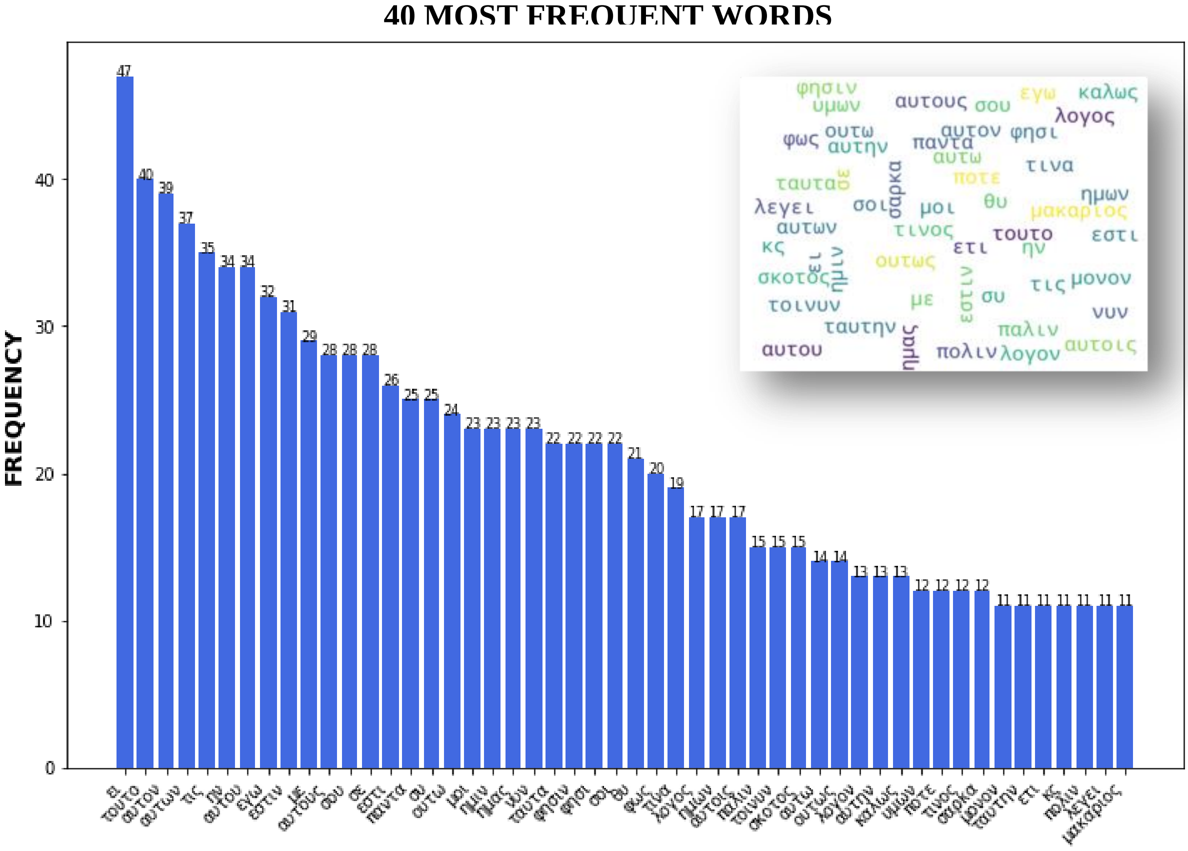


Fig.11: Word frequency in our edited data. In the upper right corner Wordcloud of edited data is provided.

What we can see in Figure 11 is that the most frequent word is *ει*. However, we should take into consideration that except for the verb ‘to be’ it can serve as a conjunction as well. Even if this be the case, there is still popularity in personal pronouns and particularly first and second person. This fact shows some tendency towards addressing somebody in the text. Words like *θ(εο)ῦ*, *λόγος*, *σάρκα*, *κ(ύριο)ς* and *μακάριος* are among the frequent words which reveal that the texts in their majority concern religious issues. The use of nomina sacra affirms the fact that these texts include many non-words in the sense that parts of the word are hidden and implied. This is one more problem for the machine for word detection.

After having investigated our data at word level, we can look at character level and draw some interesting conclusions regarding the extent to which our images can be machine-readable. Figure 12 provides character frequency statistics in the whole text.

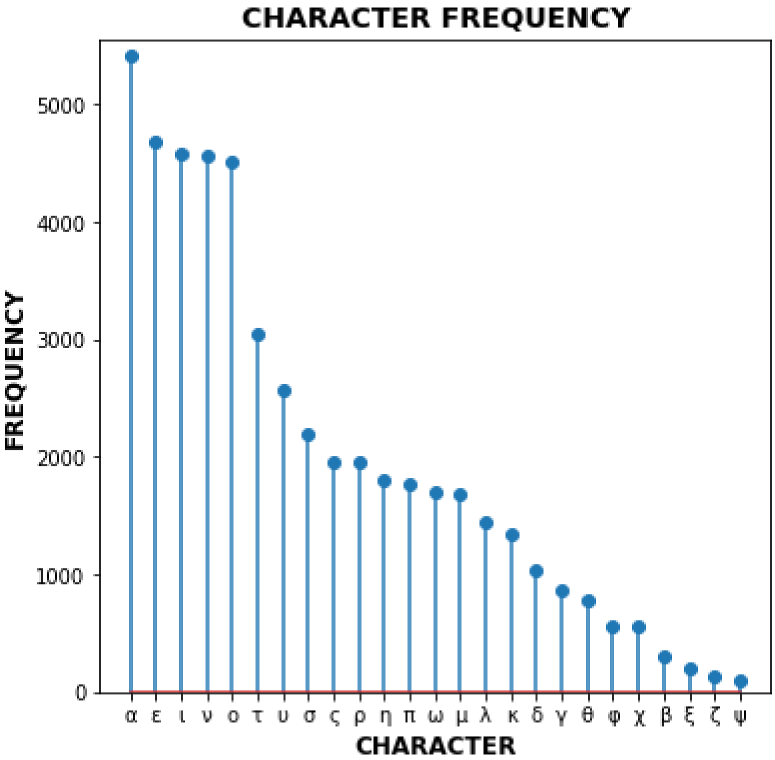


Fig.12: Character frequency

The character α appears to be the most frequent character in the text with 5.421 examples. The next four characters with high frequency are ε, ι, ν and ο with 4.682, 4.579, 4.556 and 4.503 examples respectively, while the rest of the characters appear less than four thousand times in the whole text. These numbers are extremely important because they can account for correct or false model training. In other words, the more frequent the character is the easier the machine can recognize it since it will have been trained several times on recognizing the particular character. This is exactly why we should now shift our focus to the characters with the lowest frequencies. Figure 12 shows that these characters are the following ones; β, ξ, ζ and ψ, with less than half thousand examples. This means that it will not be surprising to see that the model will make bad guesses with respect to these characters.

It is also interesting to look at character frequency for each century separately. Figure 13 enables us to get a deep understanding of character frequencies in all centuries concerned at the same time.

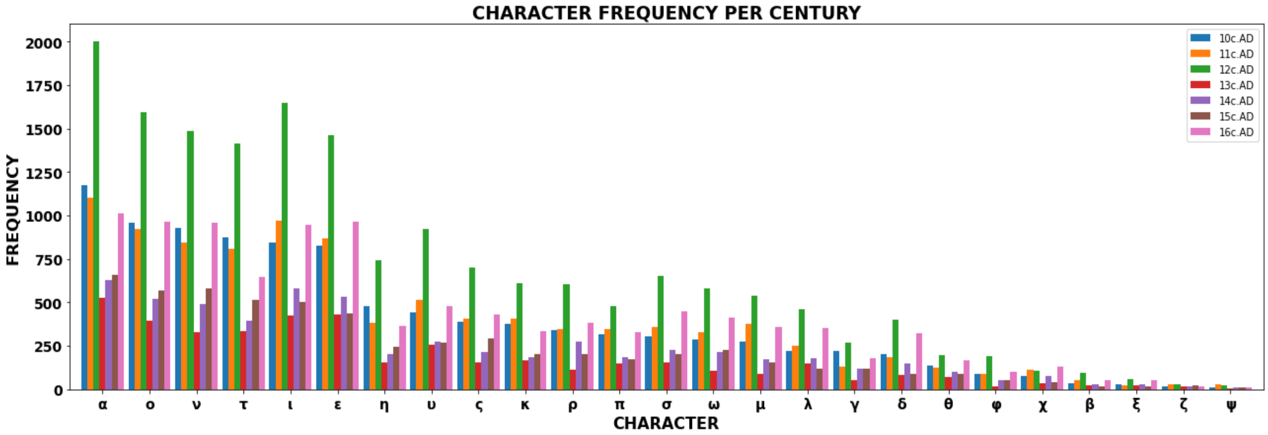


Fig.13: Character frequency per century

The character α is the most frequent character in all centuries with 1.175 examples in the tenth century data, 1.100 examples in the eleventh century data, 2.001examples in the twelfth century data, 524 examples in the thirteenth century data, 630 examples in the fourteenth century data, 661 examples in the fifteenth century data and 1.015 examples in the sixteenth century data.

Figure 13 also shows that the 12th century data share the highest character frequencies. Thus, we can presume that the 12th century manuscript is the most densely written of all the manuscripts used. Meanwhile, we can see that the 13th century data share the lowest character frequencies and thus, we can say that this data group consists of only a few text lines.

The four most frequent characters after the character α may be ο, ν, ε, ι or τ, depending on the century data group. The characters ο, ι are among the four most frequent ones in all century data groups, the character ε is among the four most frequent ones in the 11th, 12th, 13th and 16th century data groups, the character ν is among the four most frequent ones in the 10th, 11th, 12th, 14th, 15th and 16th century data groups and the character τ is among the four most frequent ones in the 10th, 13th and 15th century data groups.

Statistical analysis of the dataset can provide us with useful information about the dataset. Not only will we be able to get to know our text and draw conclusions on its use and particularities but by familiarizing ourselves with the text we will also be able to predict difficulties probable to arise during model training. At this point we will continue with the experimental phase of our study.

Chapter 3: A State-of-the-art Tool

3.1 Introduction to the tool

The tool we use for the OCR task is *Transkribus* 1.15.1. It is designed to accommodate AI-powered text recognition and transcription of historical documents[5]. The tool works efficiently thanks to the assistance of neural networks. Since we deal with old manuscripts this tool will enable us to train our own AI text recognition model, recognize layout and transcribe even in Greek. The training of such a model in *Transkribus* is to be in line with the rules established by the platform. It requires, namely, digitized image text in uniform writing style and accurate transcriptions. The model can be trained only in the case that 5,000-15,000-word-long transcription text is provided and this applies mostly to the case of handwritten text recognition rather than the printed text one[6]. A whole collection of images of text can be uploaded through the "Import document(s)" option and then manual transcription of each one of the uploaded images is to take place. Early before transcription layout analysis is necessary in order for the transcription to be aligned with the appropriate text region. The layout analysis can be automatic and the tool can find text regions alone but in the case of our study this will not suffice due to the fact that a great number of the images in use present additional information. Thus, once the tool finds the text regions of interest, we are to edit and delete regions captured that do not refer to the text. At this point we are ready to transcribe the final text lines. After transcription we can train our HTR model. In order to do this task we have to identify our train and validation data in the "HTR Training" window. In addition, we are asked to provide details concerning the model training. These details include language of image text, number of epochs and the use of a base model if desired, yet not applicable to the case of our study. Our trained model is ready to be used for text recognition. The next step is to upload a test dataset. The writing style demonstrated in the test dataset should not be extremely different from the one the training dataset demonstrates in order to be effectively machine-readable by the trained model. Once the test dataset is recognized by the trained HTR model, transcription of the text is automatically generated. Accuracy of the model is computed after transcription of the test text. The previous recognized transcription is saved and the new manual transcription is compared to the previous one. It is because of this system's function that the model's accuracy can be computed. The evaluation results involve character error rates (CER) as well as word error rates (WER).

3.2 Train and validation data

The first edition of our data serves as our train and validation data. A hundred images of text pages have been selected for training and validation of our model. Ninety three out of one

---

[5] Retrieved November 30, 2021, from https://readcoop.eu/transkribus/

[6] Retrieved November 30, 2021, from https://readcoop.eu/transkribus/howto/how-to-train-a-handwritten-text-recognition-model-in-transkribus/

hundred images are used for training while seven images are used for validation. The training image dataset consists of ten images from the tenth century data group, ten images from the eleventh century data group, ten images from the twelfth century data group, ten images from the thirteenth century data group, ten images from the fourteenth century data group, ten images from the fifteenth century group and forty images from the sixteenth century data group. Validation data needs to be representative of the training data used and thus we should choose the appropriate pages for validation. Having taken into consideration the fact that our training data are categorized into different century groups we decided to select a representative image of each century data group. In this way, seven images from the seven century data groups have been selected for the validation.

The total training time was 54’. Parameters such as epochs were defined. The training set was divided into 25 epochs. When the training and validation of our model were completed, model accuracy was provided. Figure 14 shows how accurate our model is. It demonstrates, namely, two lines, each of which represents error rates for every epoch. While training time passes, which means that epoch number increases, the error rates all but plateau. While the model is learning characters, we see that character error rates decrease. Once training time is over, character error rates on the train set reach 14.96% and character error rates on the validation set reach 17.16%. These numbers are extremely good because it means that our model is well trained and thus not prone to make many false predictions during the test phase.

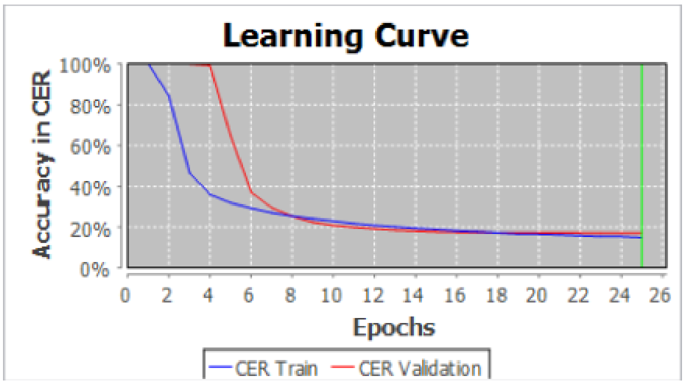


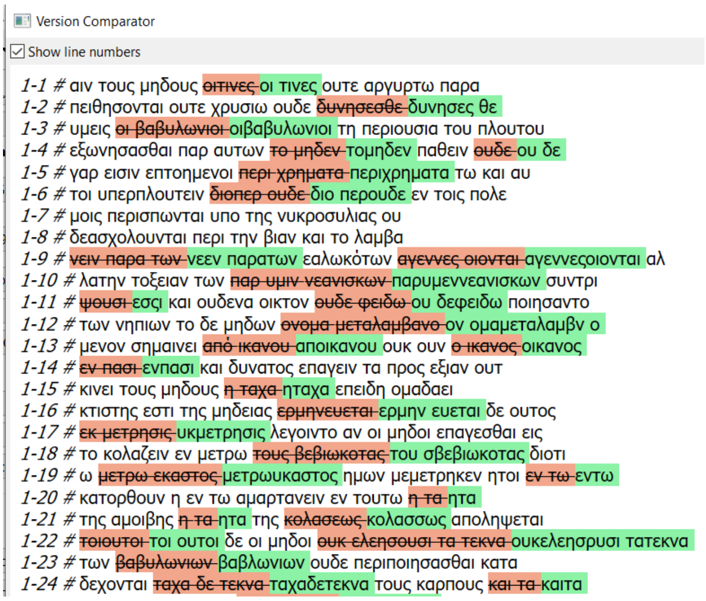
Fig. 14: Training and Validation Accuracy in CER.

3.3 Test data

The test data used are to be representative of the training dataset as well. For this reason we dealt with the case of the test dataset in the same way we did with the case of the validation dataset. In other words, we selected seven images from the collection’s manuscripts dating to the seven respective centuries which characterize our seven century data groups. *Transkribus* then provides us with the automated transcription which appears in the text editor field. Accuracy can be computed once human transcription is provided. We transcribed manually the seven test text pages and the accuracy score results from the comparison between the two text versions, the automatically transcribed one and the manually transcribed one.

3.4 OCR results

Transkribus provides a visual text comparison window for each test page. Figure 15 is an example of such a visual text comparison. What *Transkribus* shows in these windows is the text in the way it was recognized by our model. Apart from this, it also shows the false predictions the model made. These predictions are highlighted in green while the manually transcribed correct text is highlighted in red.



1-1 # αιν τους μηδους ~~οιτινες~~ οι τινες ουτε αργυρτω παρα
1-2 # πειθησονται ουτε χρυσιω ουδε ~~δυνησεσθε~~ δυνησες θε
1-3 # υμεις ~~οι βαβυλωνιοι~~ οιβαβυλωνιοι τη περιουσια του πλουτου
1-4 # εξωνησασθαι παρ αυτων ~~το μηδεν~~ τομηδεν παθειν ~~ουδε~~ ου δε
1-5 # γαρ εισιν επτοημενοι ~~περι χρηματα~~ περιχρηματα τω και αυ
1-6 # τοι υπερπλουτειν ~~διοπερ ουδε~~ διο περουδε εν τοις πολε
1-7 # μοις περισπωνται υπο της νυκροσυλιας ου
1-8 # δεασχολουνται περι την βιαν και το λαμβα
1-9 # ~~νειν παρα των~~ νεεν παρατων εαλωκότων ~~αγεννες οιονται~~ αγεννεςοιονται αλ
1-10 # λατην τοξειαν των ~~παρ υμιν νεανισκων~~ παρυμεννεανισκων συντρι
1-11 # ~~ψουσι~~ εσςι και ουδενα οικτον ~~ουδε φειδω~~ ου δεφειδω ποιησαντο
1-12 # των νηπιων το δε μηδων ~~ονομα μεταλαμβανο~~ ον ομαμεταλαμβν ο
1-13 # μενον σημαινει ~~από ικανου~~ αποικανου ουκ ουν ~~ο ικανος~~ οικανος
1-14 # ~~εν πασι~~ ενπασι και δυνατος επαγειν τα προς εξιαν ουτ
1-15 # κινει τους μηδους ~~η ταχα~~ ηταχα επειδη ομαδαει
1-16 # κτιστης εστι της μηδειας ~~ερμηνευεται~~ ερμην ευεται δε ουτος
1-17 # ~~εκ μετρησις~~ υκμετρησις λεγοιντο αν οι μηδοι επαγεσθαι εις
1-18 # το κολαζειν εν μετρω ~~τους βεβιωκοτας~~ του σβεβιωκοτας διοτι
1-19 # ω ~~μετρω εκαστος~~ μετρωυκαστος ημων μεμετρηκεν ητοι ~~εν τω~~ εντω
1-20 # κατορθουν η εν τω αμαρτανειν εν τουτω ~~η τα~~ ητα
1-21 # της αμοιβης ~~η τα~~ ητα της ~~κολασεως~~ κολασσως αποληψεται
1-22 # ~~τοιουτοι~~ τοι ουτοι δε οι μηδοι ~~ουκ ελεησουσι τα τεκνα~~ ουκελεησρυσι τατεκνα
1-23 # των ~~βαβυλωνιων~~ βαβλωνιων ουδε περιποιησασθαι κατα
1-24 # δεχονται ~~ταχα δε τεκνα~~ ταχαδετεκνα τους καρπους ~~και τα~~ καιτα

Fig. 15: From pp.2 (11th c. AD). Visual Text Version Comparison sample.
Red areas indicate manually corrected text.

In Figure 15, for instance, we see that our model hardly makes false predictions at character level. Contrarily, when it comes to the word level, false predictions increase. This means that our model cannot easily recognize empty spaces in the lines and since it has not been fed with words, it can only determine what the character and it achieves to do so to a great extent.

However, character error rates are not the same across the different century data groups. Figure 16 shows this imbalance in character recognition. What we see is that, although the model performs well in the first four century data groups, poor character recognition performance characterizes the century data groups after the thirteenth one.

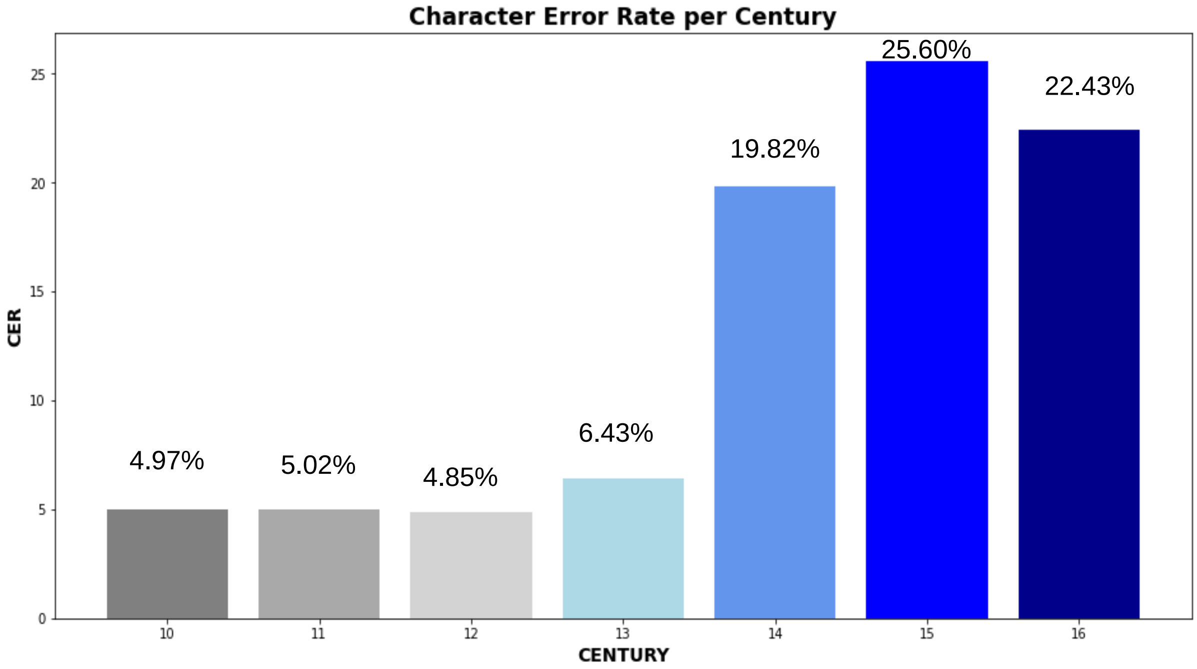


Fig.16: CER across our seven century data groups.

Word error rates are even higher than the character error ones and according to Figure 17, vary across the century data groups as well. The model can read characters more efficiently than words. In the last three century data groups, that is, the fourteenth, fifteenth and sixteenth century groups, the model reaches the highest word error rates. This may be due to the fact that the chosen manuscript pages present continuous writing, taking into consideration that the model cannot predict spaces where they are not clearly seen.

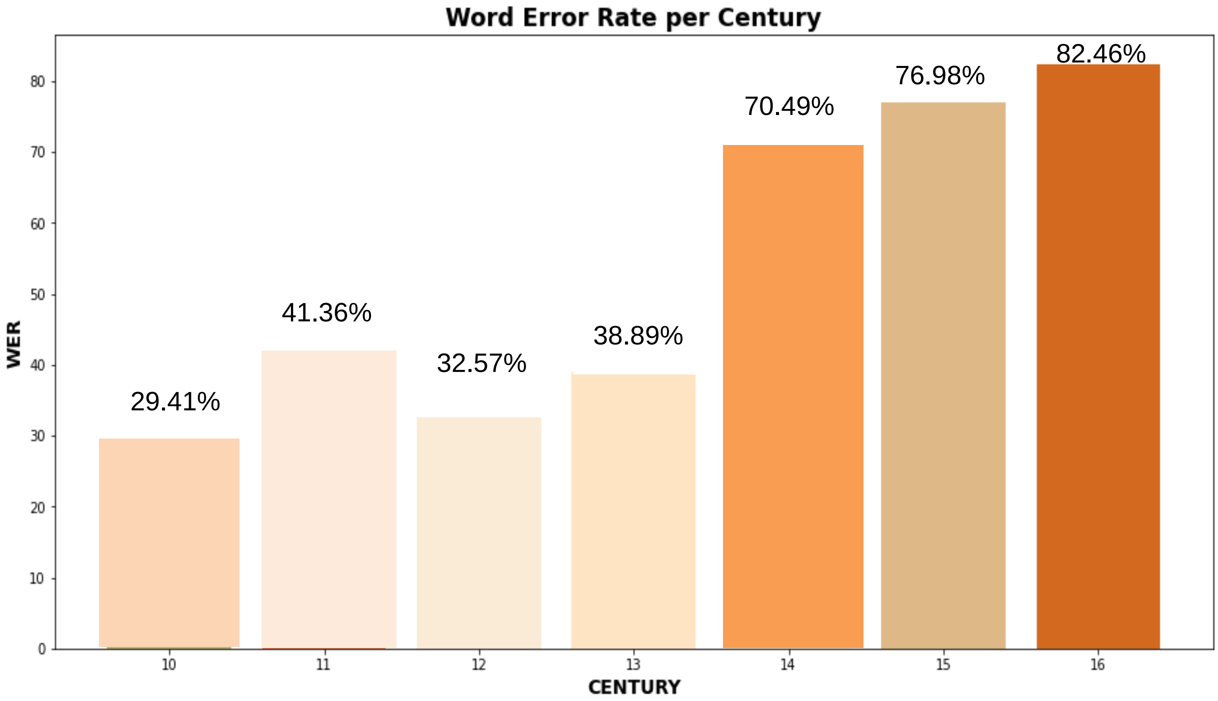


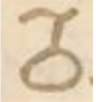
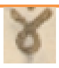
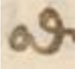
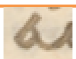
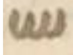
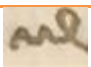
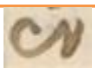
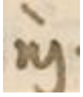
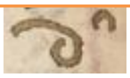
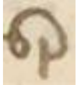
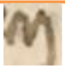
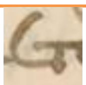
Fig.17: WER across our seven century data groups.

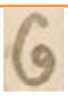
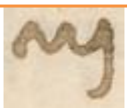
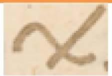
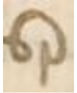
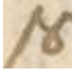
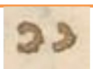
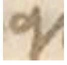
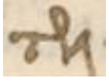
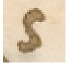
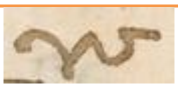
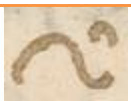
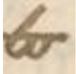
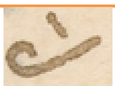
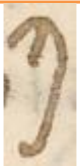
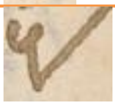
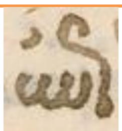
3.5 Analysis

In this *Transkribus* OCR results analysis we will focus on character misrecognition. Words are not easy for the model to read as for the reasons we have already discussed. Whereas there are reasons which can account for false word recognition, we cannot easily infer the answer to the problem when it comes to false character recognition, given that the text used is clear enough for the machine to read. However, if we examine in detail the text and take a better look at the characters, we will be able to understand what seems to be wrong in the case of the fourteenth, fifteenth and sixteenth century data groups. We consider a great number of characters in these data groups as problematic because they present one or more than one of the difficulties discussed below.

In the following table (Table 4) we provide a list of the problematic characters and their respective transcriptions. We present representative images of characters found in the fifteenth century data group. Most of these characters appear in the fourteenth and sixteenth century data groups as well. However, they are very frequent in the particular century data group we examine.

These characters are the result of a particular procedure which is usually merging, grouping or symbolization. There is a tendency towards continuous writing in the particular pages which leads to character union. The scribe does not lift up the hand when the next character is to be written and this makes a character union. Nevertheless, according to Table 4, it seems that such union does not take place everywhere in the text but, contrariwise, there are specific characters which tend to form groups. Another way of drawing characters which is quite interesting is matching characters with symbols. In this case, lines and curved lines stand for specific character combinations and they usually appear at the end of the word or the line.

MANUSCRIPT ID	CHARACTER	INTENDED MEANING	FREQUENCY
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Το	6
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Ει	9
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Στ	19
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Ου	14
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Σθ	5
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Ευ	8
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Υν	5
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Με	21
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Εν	7
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Κ	5
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Οις	3
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Ερ	7
Bodleian-Library-MS-Barocci-59_00076_fol-42v		Λλ	4
Bodleian-Library-MS-Barocci-		Ετ	4

59_00076_fol-42v			
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Σο	4
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Με	4
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Nov	8
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ερ	8
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Λο	4
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Αις	8
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Αγ	5
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Τελ	11
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ης	3
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ους	5
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ων	4
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ες	7
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ας	4
Bodleian-Library-MS-Barocci-59_00077_fol-43r		Ται	9
Bodleian-Library-MS-Barocci-59_00078_fol-43v		Ους	3
Bodleian-Library-MS-Barocci-59_00078_fol-43v		ἐπι	2

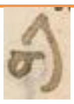
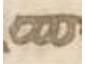
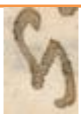
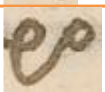
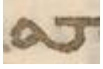
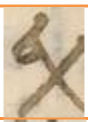
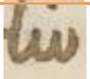
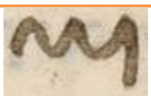
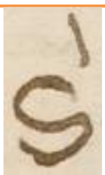
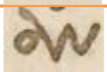
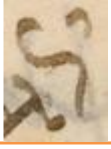
Bodleian-Library-MS-Barocci-59_00079_fol-44r		Σην	2
Bodleian-Library-MS-Barocci-59_00079_fol-44r		Σπ	7
Bodleian-Library-MS-Barocci-59_00080_fol-44v		Ελ	6
Bodleian-Library-MS-Barocci-59_00080_fol-44v		Ρο	8
Bodleian-Library-MS-Barocci-59_00080_fol-44v		Σς	5
Bodleian-Library-MS-Barocci-59_00080_fol-44v		Εχ	6
Bodleian-Library-MS-Barocci-59_00081_fol-45r		Ην	12
Bodleian-Library-MS-Barocci-59_00082_fol-45v		Μι	3
Bodleian-Library-MS-Barocci-59_00082_fol-45v		Κ	4
Bodleian-Library-MS-Barocci-59_00083_fol-46r		Αν	5
Bodleian-Library-MS-Barocci-59_00085_fol-47r		Λως	3

Table 4: Problematic 15th c. AD characters. © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

What we wish to show here is the fact that when the model is to read such complicated characters, it is extremely possible that it will make false predictions since transcription does not correspond to what it reads. Thus, character error rates increase and character recognition is not an easy task for the model to do. Our hypothesis is verified by the frequency numbers which show that problematic characters in total can account for misclassification.

Apart from the character list provided above, there are other difficulties as well. There is a trend in the sixteenth century data group for writing characters inside other characters. This is usually the case with the character “o” which is often magnified and specific characters are written inside this magnified character. Table 5 includes representative images of this character combination.

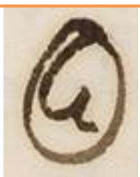
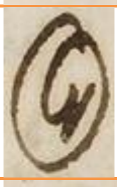
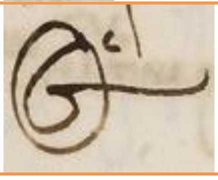
MANUSCRIPT ID	CHARACTER	INTENDED MEANING	FREQUENCY
Bodleian-Library-MS-Barocci-66_00322_fol-155v		Οι	13
Bodleian-Library-MS-Barocci-66_00322_fol-155v		Ον	3
Bodleian-Library-MS-Barocci-66_00323_fol-156r		ὅσ	3

Table 5: Problematic 16th c. AD characters. © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

Character shape and legibility go hand in hand but other difficulties arise in the data groups of interest as well. The text images of the fourteenth and sixteenth century data groups do not demonstrate only the main text but also characters between the lines of interest which are either “glosses” or text analysis. This may also account to some extent for high character error rates in these data groups.

Chapter 4: Our Approach and Experimental Validation

After some experimentation with *Transkribus*, we proceed with proposing a new model. *Transkribus* served as a great platform for experimentation with the data. What we learnt during that stage is extremely significant to the decisions to be taken in this stage. In this section we discuss the decisions we have taken in order to address the challenges of handwritten character recognition. We, namely, discuss our experimentation with CNNs for image processing and describe the results of our experiments. In the first place, we provide a step-by-step report of the data preprocessing stage. Algorithm description and results follow the preprocessing section.

4.1 Beyond the State of the art

Based on *Transkribus* OCR results analysis we were able to understand that the handwritten text recognition problem begins with the false character predictions. For this reason we have decided to focus on character level reading and thus, we use the third edition of our dataset, that is, the dataset comprising images of characters. This seems to be a promising approach to the character recognition problem because it gives us the opportunity to segment the character unions listed in the previous chapter and provide transcriptions for each character separately. In order to tackle the problem, we make use of one of the deep learning methods. We have already seen in related work that such methods are considerably popular and effective. We have adopted popular solutions to the problem in order to maximize accuracy results. We, namely, use CNNs, or ConvNets, or simply Convolutional Neural Networks, which are types of artificial neural networks. CNN architecture is discussed extensively in [16] and [17]. In our algorithm we benefit from the *Dropout*, because it aims at improving CNN performance while reducing model overfitting [18] and the *Modelcheckpoint* callback, because it saves the best model observed during the training period [19].

4.2 Data preparation

In order to be able to work with our data efficiently, we are to consider a variety of factors. First of all, we need to take into account that our goal is to make the feature demonstrated on the image machine-readable. For this reason we are required to follow a set of specific steps. This procedure targets customizing image data and can be briefly seen in Figure 18.

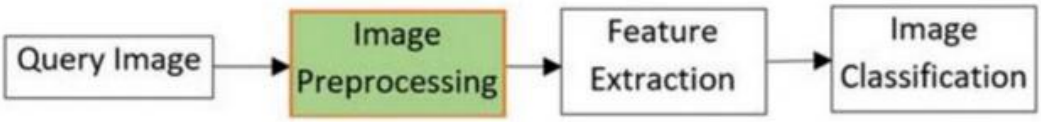
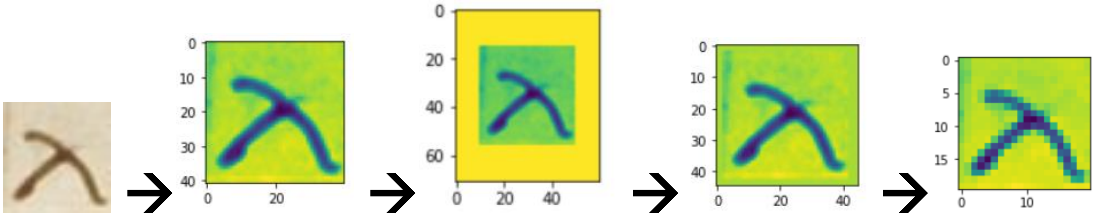


Fig.18: The image preprocessing step. Source: taken from [20].

The first step in this procedure is an image preprocessing stage and involves the cleanup of image data [20]. Our dataset consists of RGBA images. However, information can be best carried by the image intensity (i.e. grayscale value) and this is why we clean image data and convert the RGBA image into grayscale image, that is, 2D image [21]. We convert our images into 2D arrays. Afterwards, the original image needs to be zero padded in order for the length of the output to be the same as the one of the input. In particular, we pad the image boundaries with zeroes and fill the rest of the image [21]. Zero padding is of the utmost importance in our work because the images of our dataset are not cropped in the same way and hence, they bear different sizes. Zero padding enables us to extend image borders and read information near borders as well. The next step involves resizing the image and setting the dimensions to the ones wished. This is essential because in this way the same dimensions will apply to the total of our image dataset. The last step is the median value calculation. Each pixel value is replaced with the median value of the neighboring pixels. What is captured is actually the intensity of the central pixel. In this way the image is resized. The result of the procedure described here can be seen in Figure 19.



Step 1
Grayscale

Step 2
Zero padding

Step 3
Resize

Step 4
Median value

Fig.19: Image preprocessing steps © 2013 Bodleian Libraries, used under a Creative Commons Attribution-Noncommercial-ShareAlike licence: http://creativecommons.org/licenses/by-nc-sa/3.0/

At this point we need to take into account the fact that pixel values are in the range [0, 255]. What we should do is to normalize pixel values to the range [0, 1]. This may not be necessarily adequate but it can prove to be helpful because we will work with neural networks, the performance of which depends on the weights of the inputs received. Non-scaled images may lead to considerable delay or even disruption of the learning process.

After having discussed image preprocessing, we will now continue with the other task that is to be performed. We are to prepare the data for the algorithm. Machine learning algorithms usually cannot operate on label data but require numeric values as inputs and outputs instead. Our data are letters and hence, categorical values. What we have to do is to convert the categorical values into numeric. We will resort to the approach to encoding categorical features which is called label encoding. What we actually do is assign a number in range [1,24] to the twenty four letters of the Greek alphabet [α,ω]. Empty characters and punctuation marks are encoded as zeroes [0]. Later, we need to perform a second encoding task. The reason behind the new task is the multi-classification problem we face. To put it simply, we try to make the model able to recognize that the “α” character is the “α” character and that it is different from the “β” character. The model is expected to classify the input into one of the many possible classes. For this reason we import the corresponding library from Keras. The function to_categorical is the one used to convert an array of labeled data (from 0 to nb_classes - 1) to a one-hot vector. The one-hot vector is a binary vector where all values are set to zero [0] except for the index of the integer value, that is, the corresponding to the categorical feature value, which is marked with one [1].

After the encoding procedure, we decide that the training data used for our model will consist of the 90% of our data in total and the test data will form the 10% of the data. We will now examine the data on which the model will be trained. We are interested in the frequency of the characters in the images used for model training. Character frequency plays a crucial role to our model’s performance because the more instances of a character are used for training, the more accurate the character prediction made by the model will be. We proceed with statistical analysis of character distribution in the training data. Figure 20 shows the number of appearances in the training dataset for each character.

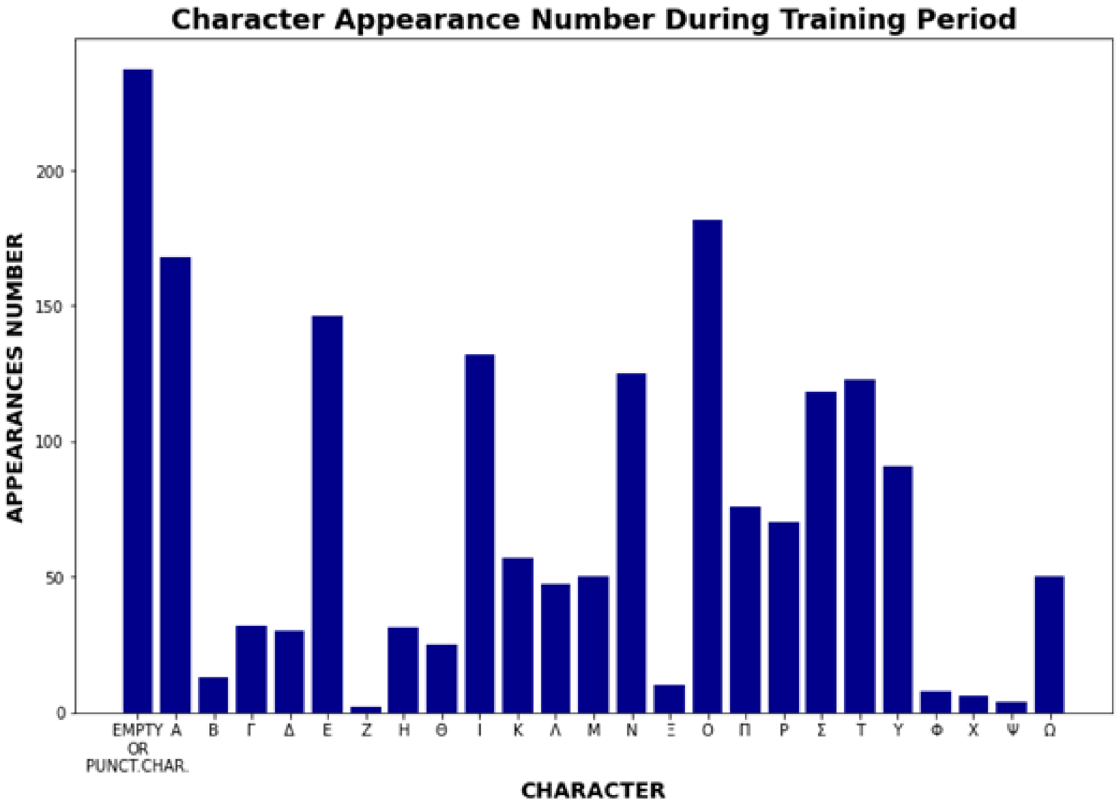


Fig.20: Character frequency in the training dataset

The figure above informs us about what the model will be able to see. In other words, during its training the model will come across all characters whether they are alphabetic, empty or punctuation. In some cases such as, for example, in the case of the characters “α” and “ο”, the model will be trained on several instances. However, the emphasis should be on character infrequency. There are particular classes of characters which appear only a few times in the training dataset. Highly underrepresented character classes are considered to be the “ζ”, “φ”, “χ” and “ψ” ones. In order to prevent poor model performance as a result of deficient training, we will use the SMOTE strategy, which stands for Synthetic Minority Oversampling TEchnique. This strategy is an effective solution to the problem of class imbalance, which applies to our case as well. It involves five steps, that is, detecting an instance of the minority class in the data, finding its k-closest neighbors, choosing one of the k-closest neighbors at random, creating a synthetic example at a randomly selected point between the chosen neighbor and the instance and then, repeating the procedure until class balance is achieved [22]. Figure 21 shows the instances of our data classes before and after upsampling with SMOTE.

2061 → 6825

Fig.21: Training data shape before and after upsampling with SMOTE.

4.2 CNN

We import the necessary libraries and packages including Keras and TensorFlow. Figure 22 shows a list of parameters applied to our model. Our CNN model consists of a Convolution 2D layer with a total of 100 filters in 3x3 kernel size, the ReLU activation function, a MaxPooling 2D layer and a Flatten layer, while two Dense layers feed the outputs from the preceding layers to their own neurons which provide the next layer with the new output. We use the stochastic gradient descent optimizer provided by Keras with a small positive value [0.01] as learning rate. The loss function used is the categorical crossentropy one. Finally, we drop out, that is, ignore, neurons during training with a given probability [0.1 or 10%] in order to avoid model overfitting. When developing our model we use 75% for training and 15% for validation.

PARAMETERS	RANGE
Batch size	20
Epochs	200
Output activation function	ReLU, Sigmoid, Softmax
Loss function	Categorical Cross Entropy
Optimizer	SGD
Dropout	0.1

Fig.22: CNN Parameters

4.3 Results

After model training, we plotted training and validation accuracy rates per epoch. The result of this task is Figure 23. What we see is that training accuracy is high while validation accuracy is promising because there is a tendency towards higher rates. The more the epochs we train the model, the better the validation accuracy is.

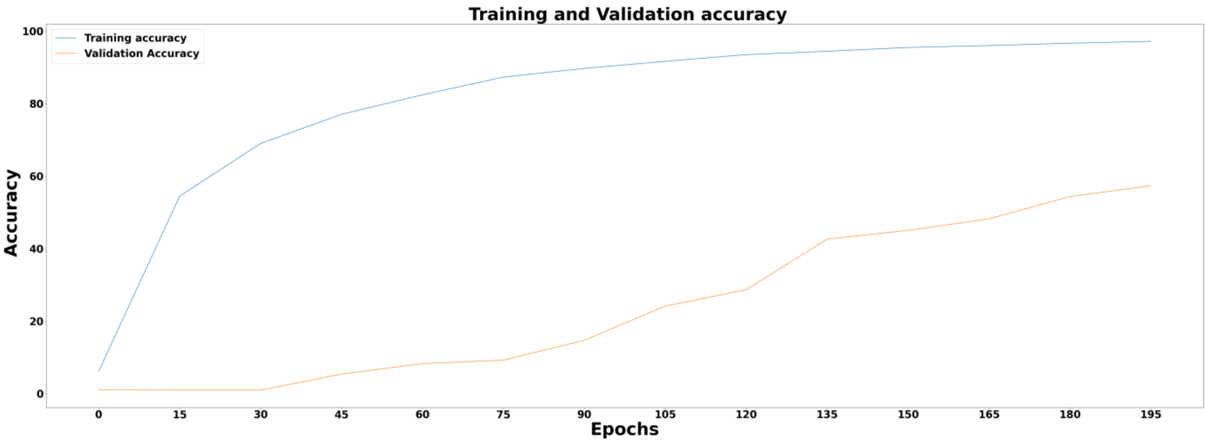
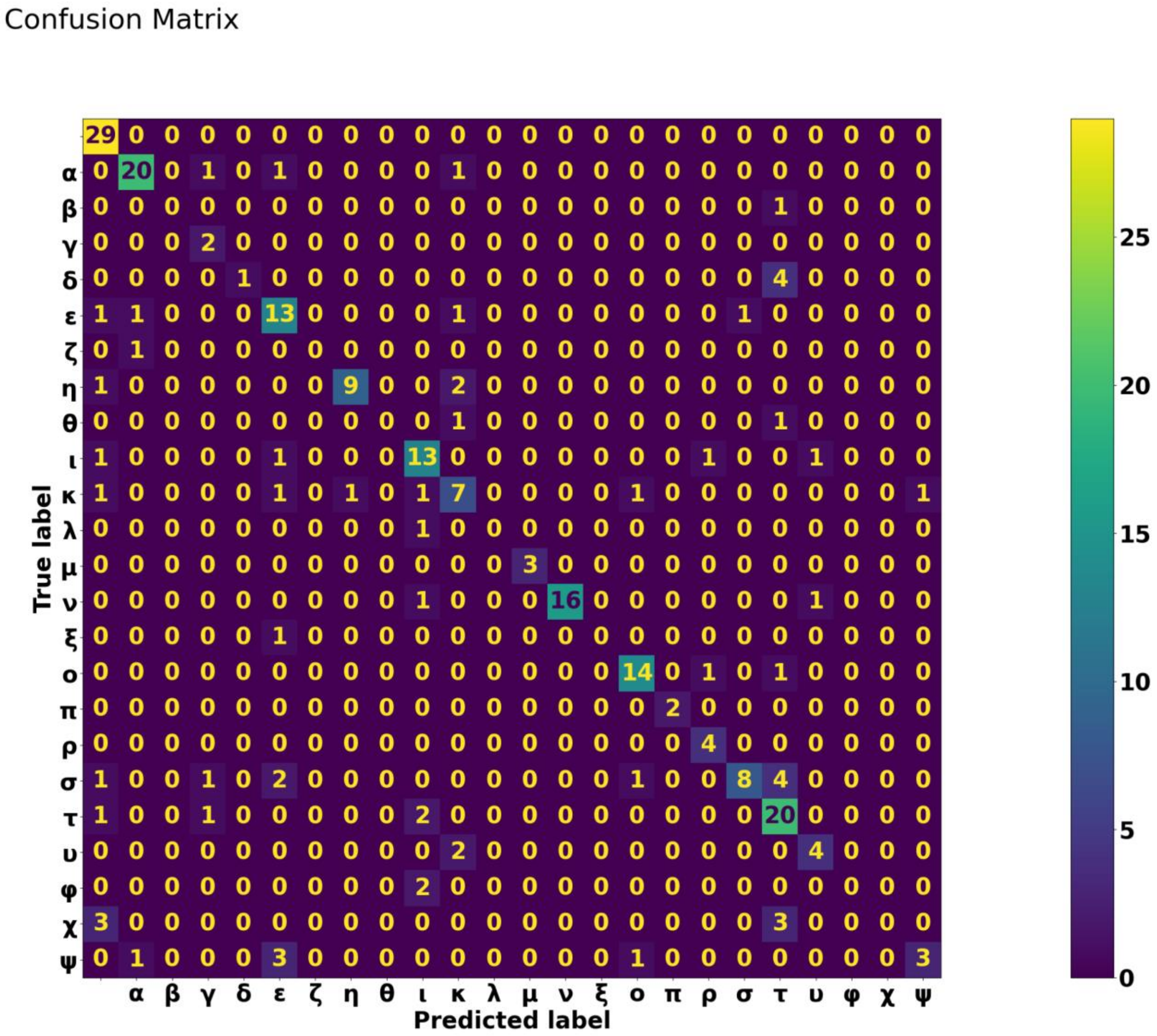


Fig.23: Training and Validation Accuracy rates given in [0, 200] epochs

We then test our trained model on our test data which we have already defined as the last 10% of the total data in use. We compute the test accuracy and the result is approximately 73%, which is remarkable given the fact that top accuracy is 100%.

We are interested in model performance on character recognition. For this reason we will plot a confusion matrix which shows the model predictions for each one of the characters.

In Figure 24, diagonal numbers of the matrix denote the correct character predictions the model has made. Light colored cells denote a large number of image character samples. Off-diagonal numbers of the matrix show incorrect model predictions. We see that these numbers are small and this fact verifies our model’s accuracy score. We will inspect the columns with many numbers, that is, many incorrect predictions. We see that there are two instances in which ε is misclassified as “σ”, three instances in which it is misclassified as “ψ”, one instance in which it is misclassified as “ξ”, one instance in which it is misclassified as “κ”, one instance in which it is misclassified as “ι” and one instance in which it is misclassified as “α”. Furthermore, we see that there are two instances in which ι is misclassified as “φ”, two instances in which it is misclassified as “τ”, one instance in which it is misclassified as “ν”, one instance in which it is misclassified as “λ” and one instance in which it is misclassified as “κ”. Another frequently misclassified character is κ with two instances in which it is misclassified as “υ”, two instances in which it is misclassified as “η”, one instance in which it is misclassified as “ε” and one instance in which it is misclassified as “α”. The character τ appears to be the most often misclassified character, with three instances in which it is misclassified as “χ”, four instances in which it is misclassified as “σ”, one instance in which it is misclassified as “ο”, one instance in which it is misclassified as “θ”, four instances in which it is misclassified as “δ” and one instance in which it is misclassified as “β”.



		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ
	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
α	0	20	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
γ	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
δ	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
ε	1	1	0	0	0	13	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
ζ	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
η	1	0	0	0	0	0	0	9	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
θ	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
ι	1	0	0	0	0	1	0	0	0	13	0	0	0	0	0	0	0	1	0	0	1	0	0	0
κ	1	0	0	0	0	1	0	1	0	1	7	0	0	0	0	1	0	0	0	0	0	0	0	1
λ	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
μ	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
ν	0	0	0	0	0	0	0	0	0	1	0	0	0	16	0	0	0	0	0	0	1	0	0	0
ξ	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ο	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	1	0	1	0	0	0	0
π	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
ρ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
σ	1	0	0	1	0	2	0	0	0	0	0	0	0	0	0	1	0	0	8	4	0	0	0	0
τ	1	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	20	0	0	0	0
υ	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	4	0	0	0
φ	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
χ	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
ψ	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3

Fig.24: Confusion Matrix showing our model’s performance on handwritten character recognition

Table 6 shows Precision, Recall and F1-score for each one of the characters. The label “other” indicates empty characters and punctuation related characters. The *Support* section provides information on the respective number of character samples the model comes across during testing.

CHARACTER	PRECISION	RECALL	F1-SCORE	SUPPORT
OTHER	71%	100%	83%	29
Α	95%	87%	91%	23
Β	0%	0%	0%	1
Γ	40%	100%	57%	2
Δ	100%	40%	57%	5
Ε	69%	65%	67%	17
Ζ	0%	0%	0%	1
Η	89%	67%	76%	12
Θ	0%	0%	0%	2
Ι	70%	82%	76%	17
Κ	64%	54%	58%	13
Λ	0%	0%	0%	1
Μ	75%	100%	86%	3
Ν	100%	83%	91%	18
Ξ	0%	0%	0%	1
Ο	65%	69%	67%	16
Π	25%	50%	33%	2
Ρ	57%	100%	73%	4

Σ	59%	59%	59%	17
Τ	61%	83%	70%	24
Υ	50%	50%	50%	6
Φ	0%	0%	0%	2
Χ	0%	0%	0%	6
Ψ	-	-	-	-
Ω	100%	0%	22%	8

Table 6: Precision, Recall, F1-score

Figure 25 provides us with visualizations of Precision, Recall and F1-score results. We see that the less easy to recognize characters are the following ones; “β”, “ζ”, “η”, “θ”, “λ” and “φ”. The model has a difficulty in identifying these characters. On the other hand, the model achieves high scores. The characters “μ”, “ν” and “π” seem to be extremely easy to classify and hence, recognize.

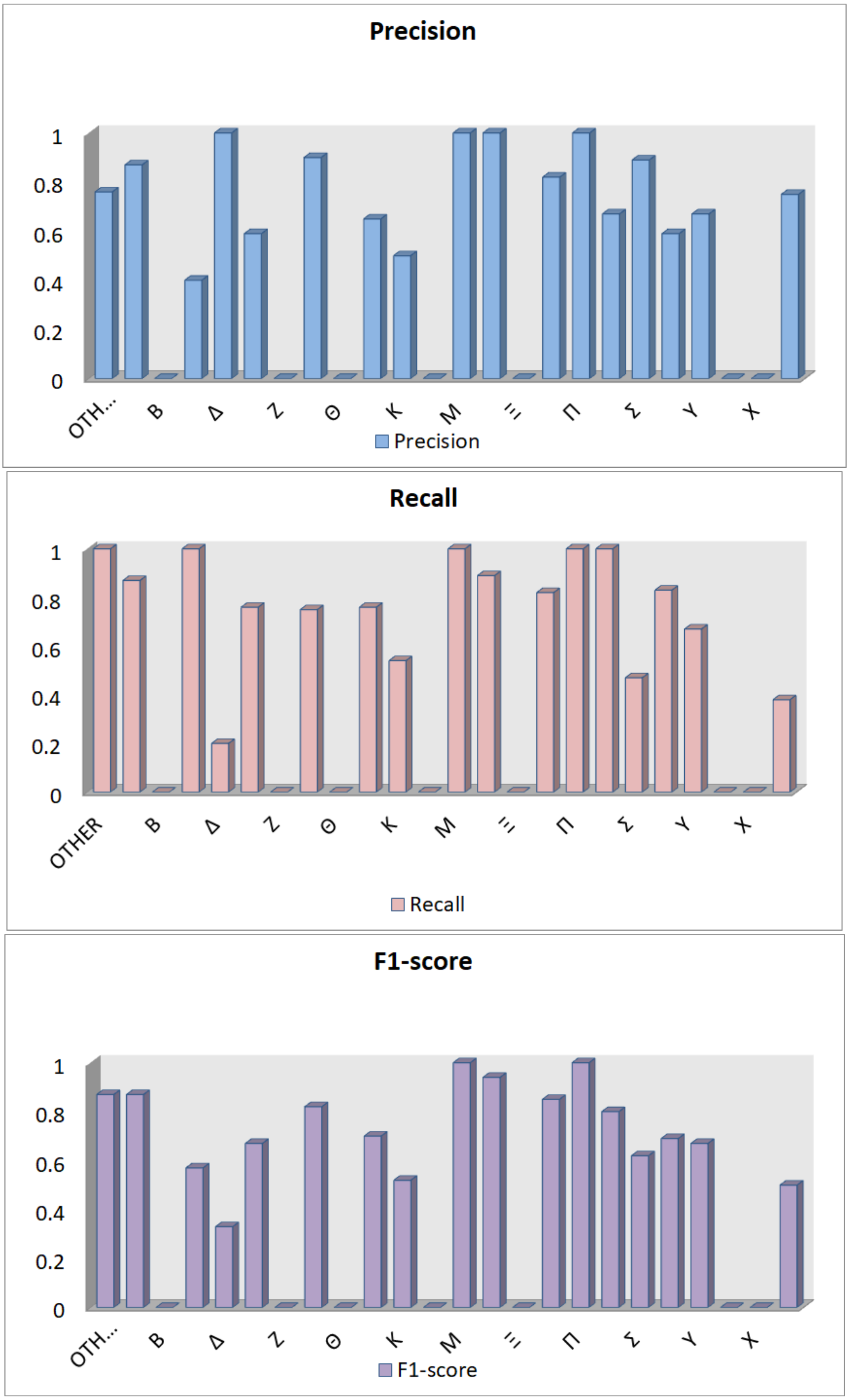


Figure 25: Model performance evaluation with the use of Precision, Recall and F1-score

4.4 Supplementary Experiment: Model Use Cases

In this section we will describe two use cases of our model. Our model has been trained on data from the Barocci collection of the Oxford University Bodleian Library. At first, we will use our model on data from the National Library of Greece manuscript collections. We, namely, use images of a Greek manuscript belonging to the manuscript collection which provides its manuscripts with the following label; *Codex Atheniensis*. The writing style in this manuscript is similar to the writing style represented by our first five century data groups. It is not cursive and it is easy to read.

For the image recognition task we follow the same procedure with the previous dataset which was used for training and testing of our OCR model. We follow the same segmentation guidelines and create a small dataset of 100 images of characters. We, then, use our OCR model to read the images. The accuracy is 67.9% which is close to the previous accuracy score. This means that our model can recognize to a great extent handwritten texts which share common characteristics with our data. The confusion matrix below (Figure 26) provides us with information on model performance.

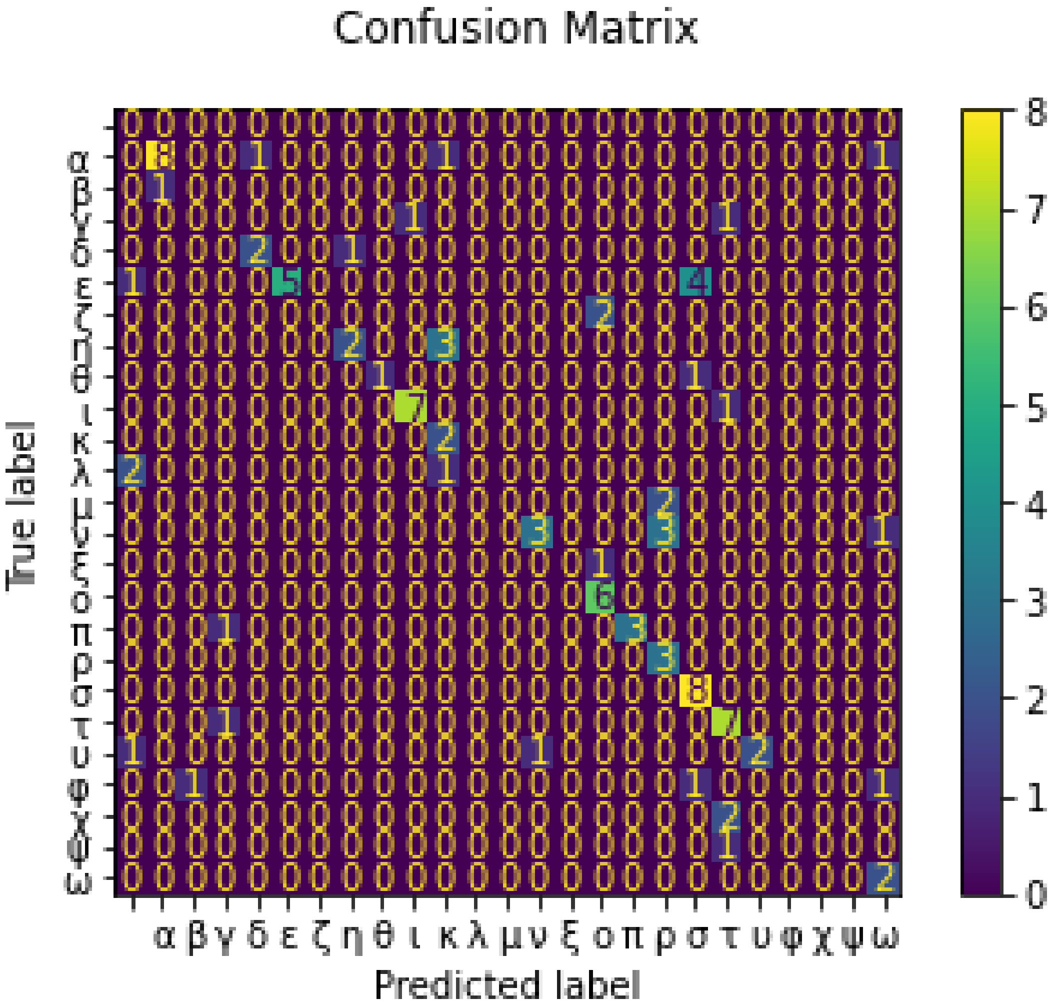


Fig. 26: CNN model performance on National Library of Greece manuscript collection data

According to the confusion matrix above, the characters **ω**, **σ**, **ρ**, **ο**, **ξ**, **μ**, **κ**, ζ and **β** are the ones which are predicted correctly. Furthermore, the confusion matrix can show us that there are certain characters which seem to be difficult to read. In other words, certain character misclassification takes place in this dataset as well as the previous one. We see that there are four instances in which **σ** is misclassified as "ε", two instances in which **τ** is misclassified as "χ" and one instance in which it is misclassified as "ι" as well as three instances in which **κ** is misclassified as "η". These actual characters and their predicted values are part of the previous misclassification problem. Apart from these characters, **ν and ρ** are now misclassified as "υ" and "ν", respectively.

At the second stage of our experiment, we compare our model's performance with *Transkribus* tool's performance on character recognition. We use a certain page of the fourteenth century data group and we evaluate the two models' performance at character level. The total number of character images used is 407. Figure 27 shows the CER each of the models produces.

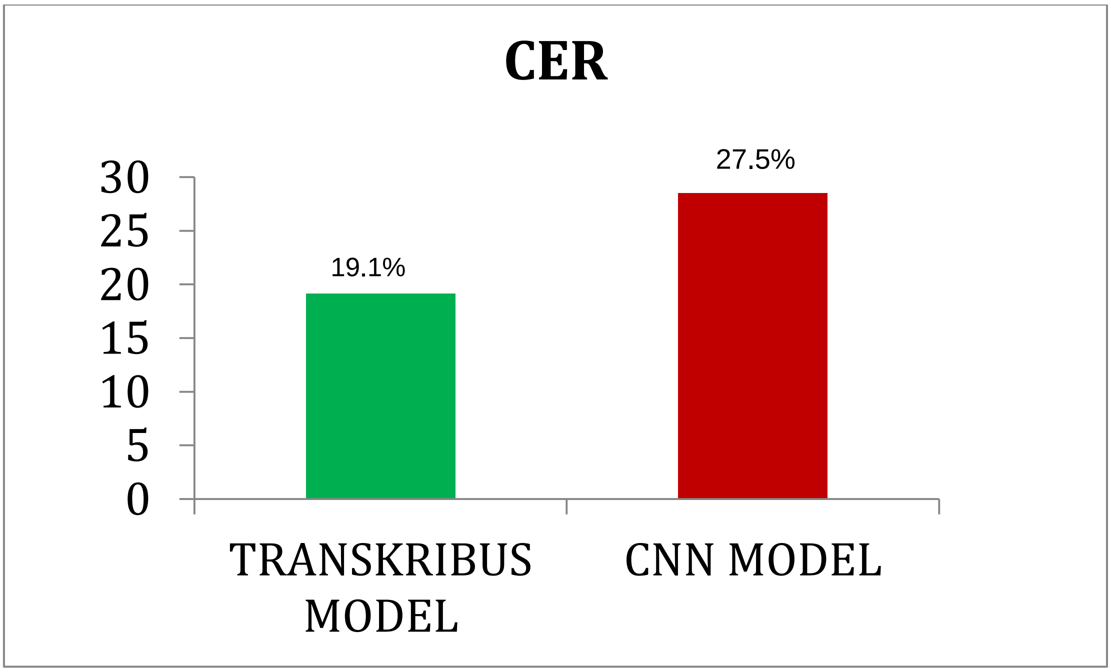


Fig. 27: CNN and *Transkribus* model CER comparison

We see that the *Transkribus* model achieves higher accuracy rates than our own model. However, our model still provides accurate results which seem to be close to the ones of the other model. We will now examine the misclassified cases in each model. In the *Transkribus* model results there are instances in which "ο" is misclassified as "ι" and "α", "ε" is misclassified as "ο" and "ι", "η" is misclassified as "ν" and "α", "υ" is misclassified as "ν", "α" is misclassified as "σ" and "π" is misclassified as "τ". Most of these characters share same characteristics in terms of shape, hence misclassification. The confusion matrix above (Figure 28) shows character misclassification in our CNN model. The most misclassified character appears to be the character “ξ” which has been misclassified in ten cases as “ε” and “η”.

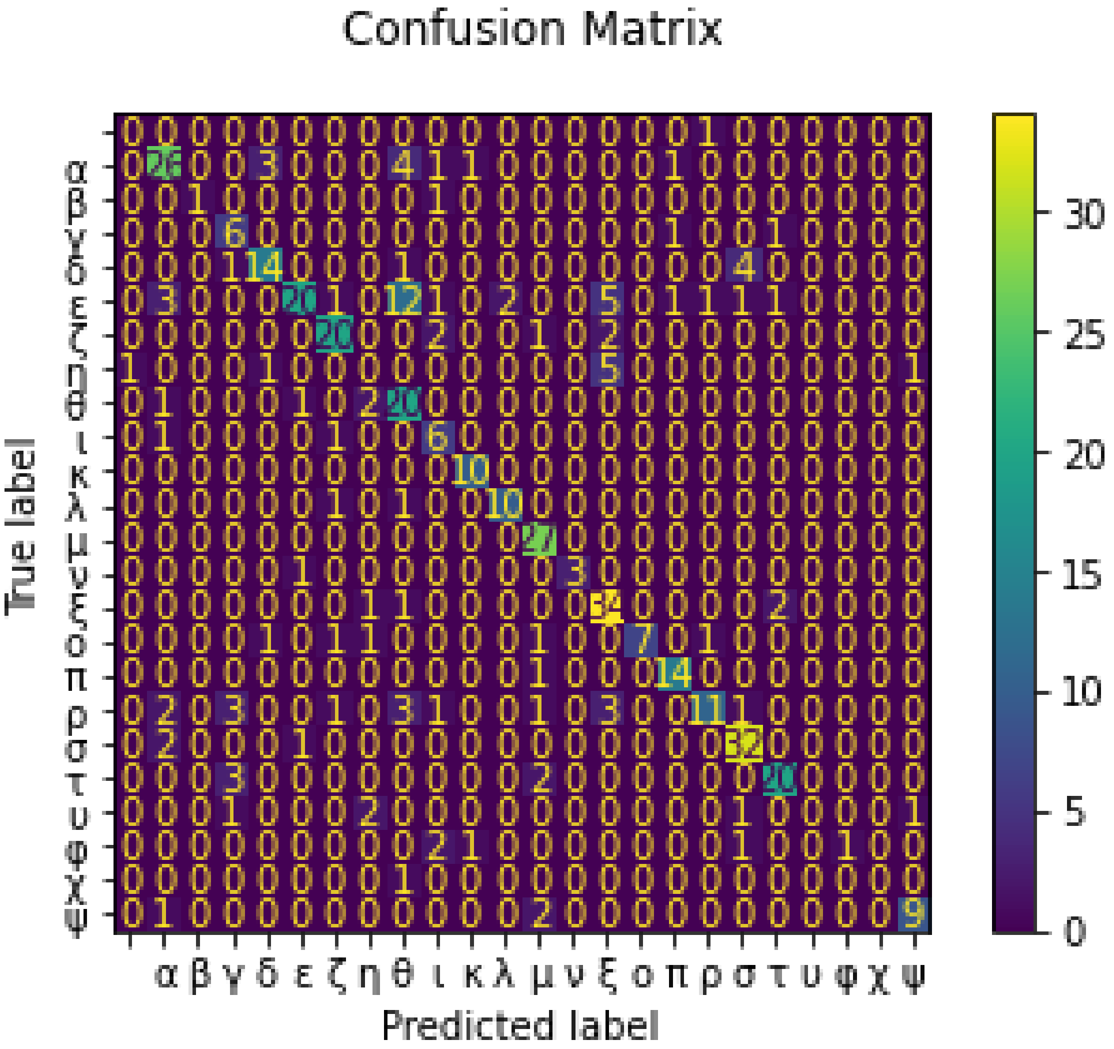


Fig. 28: CNN model performance on the same with *Transkribus* data

Chapter 5: Conclusion and Future Work

This thesis serves as a characteristic example of tasks associated with the field of Digital Paleography which is a field that stands at the intersection of Computer Science and Paleography Studies. We employed methods from the field of Computer Science in order to enable experts from the Manuscript Studies field and owners of digitized manuscript collections to transcribe, edit and search digitized manuscripts. We focused on old Greek manuscripts dating from the tenth to the sixteenth century. What we aimed to do was to automate transcription of the text demonstrated on the manuscript image. We created three data versions and after reflection on the task difficulty as well as some experimentation on our dataset with the aid of an AI-powered handwritten text recognition tool we decided that we were to proceed with the third data version, which involves character segmentation. We addressed the handwritten character recognition problem by training a deep learning model. In particular, we adopted a complex procedure which involves image preprocessing, manual transcription, human annotation, data preprocessing, statistical analysis, model training and evaluation. Our model achieves worthy results. Model accuracy is higher than 73%.

In this work, we make the following contributions:

- We create two data collections with a parallel corpus of 1,906 transcribed lines and 2,291 transcribed characters respectively from 8 manuscripts (10th–16th century) in Greek;
- We provide an error analysis, stressing the difficulty of handwritten character recognition;
- We provide two tables of difficult to identify characters appearing in our dataset
- We propose an approach that provides reduced character error rates (20-30% CER) for Greek paleographic manuscripts.

In future work, we plan to test more systems in order to reduce current character error rates. The current model performance is satisfactory and we believe that the appropriate system combination can bring remarkable results. Our future goals also include bigger data, better benchmarking and public release of the dataset. We further plan to work on full-image recognition in order to make the manuscripts as such machine-readable by providing as input images of multi-line text.

REFERENCES

[1] Aune, D. E. (2003) *The Westminster Dictionary of New Testament and Early Christian Literature and Rhetoric*. Louisville: Westminster John Knox Press. 305

[2] Mioni, E. (2009) *Εισαγωγή στην Ελληνική Παλαιογραφία*, (P. M. Nikolaos, Trans.). Athens: MIET. 42-43

[3] Thompson, E. M. (1912) *An Introduction to Greek and Latin Palaeography*. Oxford: Clarendon Press. 41

[4] Lightfoot, N. R. (2003) *How We Got the Bible*. Michigan: Baker Books. 30

[5] Royse, J. R. (2008) *Scribal Habits in Early Greek: New Testament Papyri*. Leiden: BRILL. 98

[6] Nirmalasari, D. A., Suciati, N. and Navastara, D. A. (2021) "Handwritten Text Recognition using Fully Convolutional Network". Paper represented at the IOP Conference Series: *Materials Science and Engineering (ICITDA) in Florida*. Orlando. Retrieved August 18, 2021, from https://iopscience.iop.org/article/10.1088/1757-899X/1077/1/012030

[7] Balci, B., Saadati, D. and Shiferaw, D. (2017) "Handwritten Text Recognition using Deep Learning". Journal or Conference missing. Retrieved August 18, 2021, from https://www.semanticscholar.org/paper/Recognition-using-Deep-Learning-Balc%C4%B1-Saadati/3339237110cd5fd3fa8206623e9b740be1d72c9e

[8] Vamvakas, G., Gatos, B. and Perantonis, S. J. (2010) "Handwritten character recognition through two-stage foreground sub-sampling". *Pattern Recognition* 43(8): 2807-2816. Retrieved August 19, 2021, from https://www.researchgate.net/publication/223519735_Handwritten_character_recognition_through_two-stage_foreground_sub-sampling

[9] Haviluddin, Rayner Alfred R., Moham, N., Pakpahan, H. S., Islamiyah and Setyadi, H. J. (2019) "Handwriting Character Recognition using Vector Quantization Technique". *Knowledge Engineering and Data Science (KEDS)* 2: 82-89. Retrieved August 19, 2021, from https://www.researchgate.net/publication/338128881_Handwriting_Character_Recognition_using_Vector_Quantization_Technique

[10] Daimary, D., Bora, M. B., Amitab, K., Kandar, D. (2020) "Brain Tumor Segmentation from MRI Images using Hybrid Convolutional Neural Networks". *Procedia Computer Science* 167: 2419-2428. Retrieved August 18, 2021, from https://www.sciencedirect.com/science/article/pii/S1877050920307614

[11] Ntzios, K., Gatos, B., Pratikakis, I., Konidaris, T. and Perantonis, S. (2005). "An old Greek handwritten OCR system". Paper represented at the 9th International Conference on Document Analysis and Recognition (ICDAR) in South Korea. Seoul. Retrieved August 5, 2021, from https://www.researchgate.net/publication/4214782_An_old_Greek_handwritten_OCR_system

[12] Messina, R. and Louradour, J. (2015) "Segmentation-free handwritten Chinese Text Recognition with LSTM-RNN". Paper represented at the 13th International Conference of Document Analysis and Recognition (ICDAR) in France. Paris. Retrieved August 10, 2021, from https://www.researchgate.net/publication/278022925_Segmentation-free_Handwritten_Chinese_Text_Recognition_with_LSTM-RNN

[13] Yousef, M. and Bishop, T. E. (2020) "OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by learning to unfold". Paper represented at the Conference on Computer Vision and Pattern Recognition (CVPR) virtually. Retrieved August 10, 2021, from

[14] Lyu, L., Koutraki, M., Krickl, M., Fetahu, B. (2021) "Neural OCR Post-Hoc Correction of Historical Corpora". *Transactions of the Association for Computational Linguistics*. 9: 479–493

[15] Papantoniou, K. and Tzitzikas, Y. (2020) "NLP for the Greek Language: A Brief Survey". Paper represented at the11th Conference on Artificial Intelligence (SETN) in Athens. Greece.

[16] Karim, M. R., Sewak, M. and Pujari, P. (2018) *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python.* Birmingham: Packt Publishing

[17] Hemanth, D. J. and Estrel, V. V. (2017) *Deep Learning for Image Processing Applications.* Amsterdam: IOS Press

[18] Liu, D., Xie, S., Li, Y., Zhao, D. and El-Sayed El-Alfy, M. (eds) (2017) *Neural Information Processing*: *24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I.* New York: Springer

[19] Brownlee, J. (2020) *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions.* Vermont, Victoria: Machine Learning Mastery

[20] Chaki, J., Dey, N. (2018) *A Beginner's Guide to Image Preprocessing Techniques.* New York: CRC Press, Taylor and Francis Group- Series: Intelligent Signal Processing and Data Analysis

[21] Pattanayak, S. (2017) *Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python.* Berkeley, CA: Apress

[22] Jain, V., Juneja, S., Juneja, A., Kannan, R. (eds) (2020) *Handbook of Machine Learning for Computational Optimization: Applications and Case Studies.* U.K.: CRC Press, Taylor and Francis Group.