The Application of Deep Learning in Optical Character Recognition

Sildi Shahini
European University of Tirana

Ardiana Topi
European University of Tirana

Forsian Elezi
University "Aleksandër Moisiu", Durrës

Abstract

Optical Character Recognition (OCR) is an essential technology for document digitization, enabling the conversion of scanned paper documents, PDFs and images into editable and searchable data. This paper focuses on the application of deep learning in OCR, particularly in digitizing handwritten medical prescriptions, where accuracy is critical for reducing errors and improving healthcare outcomes. Traditional OCR methods face challenges when dealing with handwritten texts due to the variability in handwriting styles and the quality of scanned documents. These limitations can result in recognition errors, which, in a medical context, may lead to serious consequences such as medication errors.

To address the above issue, the study explores deep learning approaches, especially Convolutional Neural Networks (CNNs), that have shown significant promise in overcoming these challenges by learning from large datasets. The study involves collecting handwritten prescriptions, preprocessing the images, and training a deep learning-based OCR model. Performance evaluation metrics, including accuracy, *precision, recall, and F1-score, indicate that the deep learning model significantly outperforms traditional OCR methods in recognizing handwritten prescriptions.*

The results demonstrate the deep learning model's ability to handle the variability of handwriting more effectively, providing a more reliable solution for digitizing medical documents. This research underlines the transformative potential of deep learning in OCR technology, particularly for critical applications such as healthcare. The findings advocate for the wider adoption of deep learning in the healthcare sector, aiming to improve patient care, reduce human error, and enhance operational efficiency, especially in pharmacy management and medical record-keeping.

Keywords: *optical character recognition, neural network, convolutional neural network, deep learning*

Introduction

Deep Learning (DL) has fundamentally changed the applications of Optical Character Recognition (OCR) technology by improving its accuracy, speed, and overall applicability. As this technology continues to evolve, integrating it with OCR will undoubtedly pave the way for new opportunities in automation and data processing efficiency across various sectors (Fateh et al., 2023).

OCR technology has undergone significant transformations since its inception, driven mainly by advancements in DL. These innovations have enabled OCR systems to achieve high accuracy in recognizing text across different document types and conditions (Nockels et al., 2022). The integration of DL into OCR gained momentum in the early 2010s when traditional methods, such as rule-based systems and basic pattern recognition, proved insufficient for complex text environments. Convolutional Neural Networks (CNNs) emerged as an especially effective model for OCR systems because of their ability to process the spatial hierarchy of images, making them ideal for handling varied layouts and font styles found in documents.

Recent studies highlight the effectiveness of DL in OCR. A standard study by Zhao et al. (2020) compared the performance of CNNs and LSTMs in OCR tasks across different languages and found that DL models consistently outperformed traditional OCR methods, with accuracy improvements of up to 20% in complex documents like invoices.

Purpose

The purpose of this thesis is to study Deep Learning technology and its application in developing a system for digitizing medical prescriptions. Medical prescriptions are critical in healthcare, containing essential information about medications, dosages, and patient instructions. However, manually processing these handwritten documents is prone to errors, time-consuming, and inefficient. This inefficiency underscores the need for an effective OCR solution that can accurately digitize handwritten prescriptions, thereby improving healthcare services, reducing medication errors, and optimizing pharmacy operations.

To address this need, the thesis uses a combination of advanced technologies, including Deep Learning, Python, Tesseract, OpenCV, and the Efficient and Accurate Scene Text (EAST) detector. By integrating these technologies, a comprehensive OCR system capable of handling the complexity of handwritten prescriptions is developed. This system combines traditional OCR techniques with modern DL approaches to achieve superior performance.

This study details the methodology for developing the OCR system, including data collection, preprocessing, model training, and evaluation. A case study on the digitization of medical prescriptions demonstrates the practical applications and benefits of this approach. By comparing the system's performance with traditional OCR methods, the thesis emphasizes significant improvements through DL technologies.

Objectives

The primary objective of this thesis is to explore and demonstrate the effectiveness of DL techniques in improving OCR for digitizing handwritten medical prescriptions. The specific objectives include:

Evaluating Traditional OCR Methods: Assessing the limitations and challenges of traditional DL techniques, especially in recognizing handwritten text in medical prescriptions.

Developing a DL-Based OCR System: Designing and implementing an OCR system using Python, Tesseract, OpenCV, and the EAST text detector to improve the accuracy and reliability of digitizing handwritten prescriptions.

Data Collection and Processing: Gathering a comprehensive dataset of handwritten prescriptions and preprocessing these images to improve text recognition.

Model Training and Optimization: Training DL models, focusing on CNNs and RNNs, to accurately recognize and digitize handwritten text.

Performance Evaluation: Evaluating the developed OCR system's performance using metrics like accuracy, precision, recall, and F1-score, and comparing it with traditional OCR methods.

Demonstrating Practical Applications: Conducting a case study on the digitization of medical prescriptions to demonstrate the practical applications and benefits of the developed OCR system.

Future Integration and Research: Exploring the potential for integrating the developed system with existing electronic health record (EHR) systems and identifying areas for future research and improvement.

Research Questions

To guide this study, the primary research question has been formulated, aiming to address the challenges and opportunities in using DL techniques to improve the accuracy and efficiency of OCR for digitizing handwritten medical prescriptions:

How can DL techniques like CNNs and RNNs be applied to improve the accuracy of OCR in recognizing handwritten text in medical prescriptions?

This question seeks to investigate the potential of advanced DL models in addressing the limitations of traditional OCR methods, particularly in processing handwritten text from medical professionals. By answering this question, the study aims to provide insights into the benefits and limitations of these techniques in the context of accurately recognizing medical text.

By answering the research questions, the thesis aims to contribute to practical and theoretical knowledge on applying DL in OCR and offer recommendations for improving healthcare services through enhanced digitization of critical medical documents.

Hypothesis

The hypothesis driving this thesis is that DL techniques, particularly those using CNNs and RNNs, can significantly improve the accuracy and efficiency of OCR for digitizing handwritten medical prescriptions compared to traditional OCR methods.

Accuracy Improvement Hypothesis: DL-based OCR will significantly improve the accuracy of recognizing handwritten medical prescriptions compared to traditional OCR methods.

The potential benefits of DL in OCR applications, especially for digitizing medical prescriptions, are substantial. The research will include comprehensive data collection, model training, and performance evaluation to determine whether DL can surpass traditional OCR methods in terms of accuracy, efficiency, and error reduction.

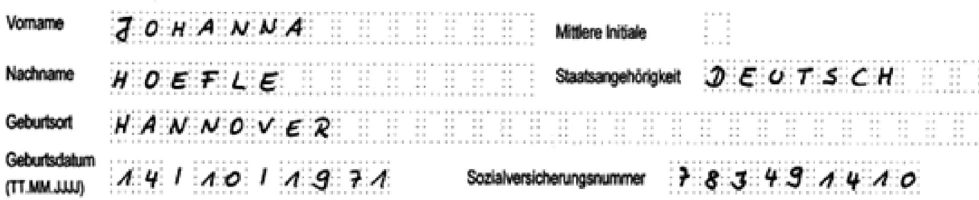
Traditional OCR Methods

Traditional Optical Character Recognition (OCR) methods have been fundamental in the early development of text recognition systems. These methods relied heavily on pattern recognition techniques, feature extraction, and template matching. Despite their initial success, traditional OCR methods encountered significant challenges, especially when dealing with handwritten text and various font styles. This section explores the key techniques used in traditional OCR and their limitations.

Pattern recognition was one of the earliest techniques employed in OCR systems. This approach involves comparing input characters with previously stored templates. Each character in the input image is matched against a database of known character patterns. Techniques like Optical Correlation were used to find the best match between the input character and stored templates. However, pattern recognition methods struggled with variations in handwriting, different fonts, and noise in scanned documents. Relying on exact or near-exact matches made these systems less robust when faced with real-world variability.

Feature extraction techniques aimed to improve the stability of OCR systems by focusing on distinguishing features of characters instead of relying on exact templates. Features like edges and angles were extracted from the input image and used to identify characters. Methods like zoning, where the character image is divided into zones and the pixel density in each zone is analyzed, became popular. The zoning method works as shown in the figure below.

FIGURE 1: Zoning Method (Cem Dilmegani, 2024)



Vorname JOHANNA Mittlere Initiale

Nachname HOEFLE Staatsangehörigkeit DEUTSCH

Geburtsort HANNOVER

Geburtsdatum (TT.MM.JJJJ) 14 / 10 / 1971 Sozialversicherungsnummer 783491410

Using statistical methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) allowed for dimensionality reduction and improved feature representation. These methods enhanced OCR systems' ability to handle variations in character appearance but still faced limitations with complex backgrounds and noise.

Template matching was another fundamental technique in traditional OCR. This approach involved comparing the input image with a set of predefined templates for each character. The character is recognized based on the template that best matches. Techniques like normalized cross-correlation were used to measure the similarity between the input image and templates. While effective for recognizing printed text with stable fonts, template matching was less successful with handwritten text due to the variability in writing styles. Moreover, template matching was computationally expensive, as it required numerous comparisons with all possible templates.

Statistical methods such as Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) were introduced to improve the accuracy of OCR systems. HMMs modeled the sequential nature of text and were particularly useful for handwriting recognition. SVMs, on the other hand, provided powerful classification capabilities by finding the optimal hyperplane that separates different character classes.

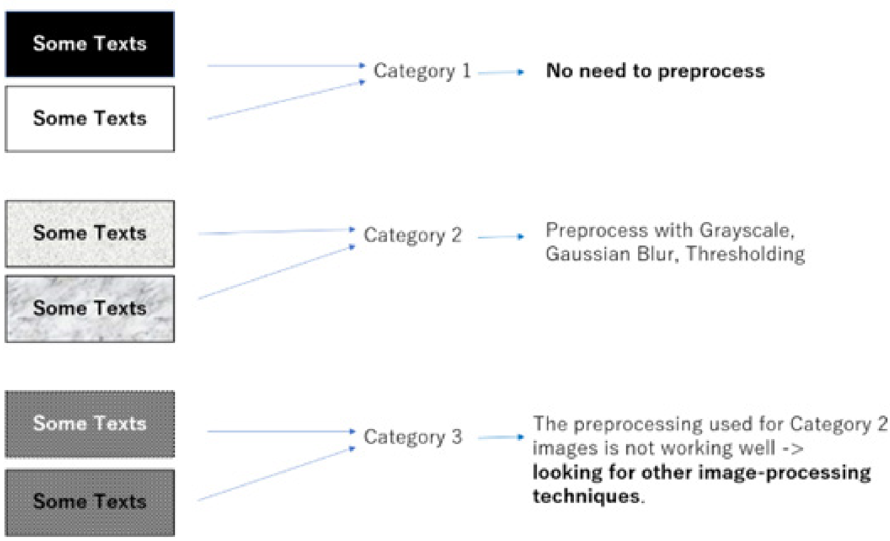
Structural methods focused on the geometric and topological properties of characters. Techniques like graph-based representations and syntactic pattern recognition exploited the structural relationships between different parts of a character. These methods aimed to offer a more holistic understanding of character shapes but required complex preprocessing steps and feature extraction.

Despite their contributions to OCR technology, traditional methods faced several limitations:

Variability in handwriting: Traditional methods struggled with the wide range of handwriting styles, leading to high error rates in recognizing handwritten text.

Sensitivity to noise: Scanned documents often contained noise, such as stains and distortions, which negatively impacted the performance of traditional OCR systems.

FIGURE 2: Noise Handling Approach (NguyenHai, 2022)



Computational complexity: Many traditional methods, particularly template matching and statistical methods, were computationally intensive, making them less suitable for real-time applications.

Limited flexibility: Traditional OCR systems were less adaptable to new fonts and handwriting styles without extensive retraining or redesign.

Traditional OCR methods laid the foundation for modern OCR systems by introducing key concepts in pattern recognition, feature extraction, and statistical modeling. However, their limitations underscored the need for more advanced techniques capable of handling the complexities of text recognition in the real world. The emergence of deep learning has addressed many of these challenges, offering more robust and accurate solutions for OCR tasks.

The Evolution of Deep Learning in OCR

The evolution of deep learning in Optical Character Recognition (OCR) has transformed the field from basic pattern recognition techniques to sophisticated and highly accurate models used today. This section reviews key moments and technological advancements that have propelled OCR forward through deep learning, highlighting recent developments from the last five years.

Early Neural Networks and OCR

The integration of neural networks into OCR began in the late 1980s and early 1990s. LeCun (1998) was one of the pioneers to introduce Convolutional Neural Networks (CNNs) for digit recognition. Their work on the LeNet-5 architecture demonstrated that CNNs could automatically extract features from images, significantly improving character recognition accuracy. This marked a departure from traditional OCR methods, which relied heavily on manually engineered features and template matching.

Convolutional Neural Networks (CNNs)

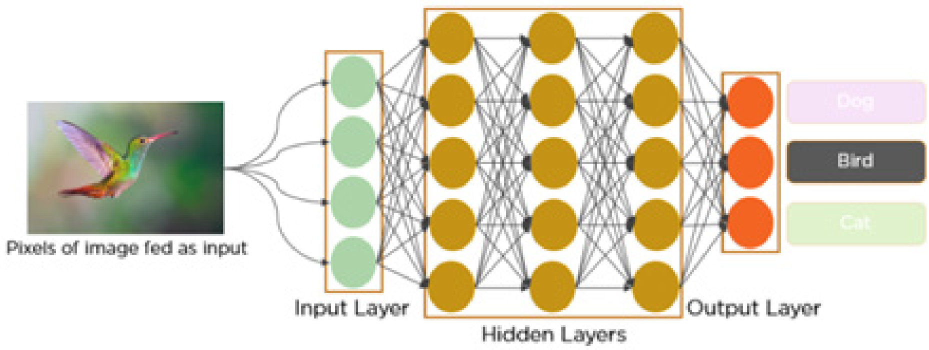
CNNs are a subcategory of machine learning at the heart of deep learning algorithms. They consist of layers of nodes, including an input layer, one or more hidden layers, and an output layer. Each node is connected to another, with an assigned weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed to the next layer.

Several types of neural networks are used for different use cases and data types. For instance, Recurrent Neural Networks (RNNs) are commonly used for natural language processing and speech recognition tasks, while Convolutional Neural Networks (CNNs) are often used for image classification and computer vision tasks. Before the advent of CNNs, manual and laborious feature extraction methods were used to identify objects in images. However, CNNs now offer a more scalable approach to image classification and object recognition tasks by leveraging principles from linear algebra, particularly matrix multiplication, to identify patterns within an image.

CNNs revolutionized OCR by providing a powerful method for feature extraction and pattern recognition. The hierarchical structure of CNNs allows them to capture spatial hierarchies in images, making them particularly effective for text recognition. Advances in deep learning frameworks like TensorFlow and PyTorch have made it easier to develop and train CNNs on large datasets.

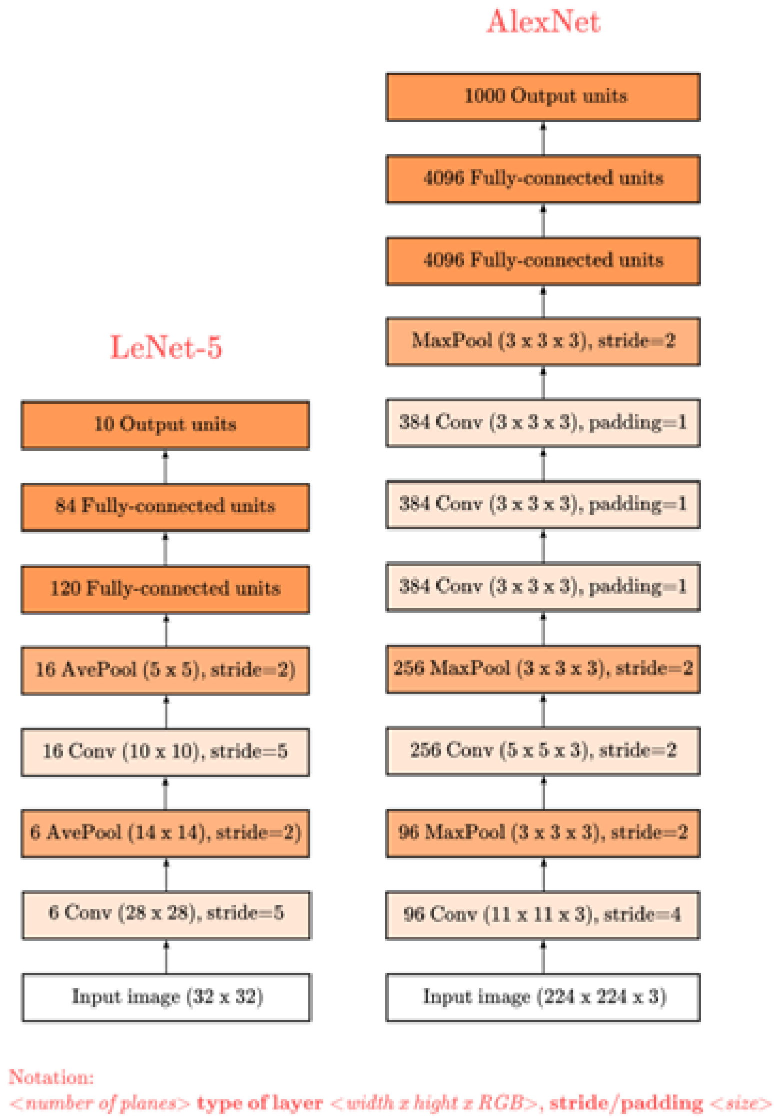
The success of the AlexNet architecture in 2012 highlighted the potential of deep learning in image processing tasks, including OCR. Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet was a breakthrough in deep learning. It used a deeper and wider architecture than its predecessors, with five convolutional layers and three fully connected layers. It also introduced innovations like ReLU activation functions and dropout regularization to prevent overfitting. This architecture achieved unprecedented accuracy on the ImageNet dataset, significantly outperforming traditional image recognition methods and ushering in a new era for CNNs.

FIGURE 3: AlexNet Model Performing Image Recognition Algorithm



The LeNet architecture, developed by Yann LeCun and his colleagues in the late 1980s and early 1990s, was one of the first CNN architectures designed for handwritten digit recognition. It laid the groundwork for modern CNNs with its simple but effective structure consisting of two convolutional layers followed by two subsampling layers and two fully connected layers. LeNet was instrumental in demonstrating the feasibility of using CNNs for image recognition tasks and inspired further research and development in this field.

FIGURE 4: Comparison Between LeNet and AlexNet (Pablo Caceres, 2020)



CNNs excel at handling variations in handwriting by learning from diverse datasets. Techniques like data augmentation, which involves artificially increasing the diversity of training samples, have further improved CNN robustness. This has allowed OCR systems to better generalize across different handwriting styles and text layouts.

OCR in the Field of Medicine

One of the most researched applications of OCR systems in healthcare involves the digitization of blood pressure (BP) records using Optical Character Recognition (OCR) across various forms of healthcare settings (SalahEldin Kasem et al., 2023). OCR in mobile health (m-health), particularly based on smartphone technology, is also an active research area, especially the digitization of BP records using smartphone cameras. For example, Ghoneim et al. adapted Google's TensorFlow library for mobile devices and leveraged convolutional neural networks (CNNs) in the m-health domain to digitize BP readings recorded by BP measurement instruments. This allowed for the automatic digitization of BP meter readings, ensuring a hard copy remained intact throughout the retrieval process and eliminating the need for handwritten transcription.

Another key practice in modernizing healthcare methods involves the digitization of medical charts and related materials through images or screenshots. Sokina et al. introduced a "DeepNeuro TextSpotter" (DN-TS) system that further processes bounding boxes of extracted characters and performs character segmentation, known as the textbox process. However, unlike word-spotting techniques, which require each word to be isolated and processed individually, this OCR method was more densely focused on obtaining accurate bounding boxes from medical images compared to TextSpotter.

The proposed model is reported to be evaluated using open-source projects from DeepNeuro that handle TextSpotter methods for OCR in various medical imaging languages. The evaluations include Hindi, Nepali, and Bangla, which cover most languages of the Indian subcontinent, and Arabic, English, and Malay, supporting OCR in Gulf countries and Australia.

Methodology

In this chapter, we describe the methodology followed in this study, covering the data collection process, the technology used, model training and evaluation, implementation steps, and ethical considerations involved.

The goal is to provide a clear and detailed description of the procedures and techniques used to achieve the research objectives. This chapter serves as a guide to reproduce the study and ensure that the results are reliable and valid.

The research focuses on the application of deep learning techniques in Optical Character Recognition (OCR) for the digitization of handwritten medical prescriptions. Given the critical importance of accurately extracting data in the medical field, the methodology is designed to address the unique challenges presented by handwritten text, which often includes variations in writing styles, inaccuracies, and complex medical terminology.

To achieve the research objectives, a robust and systematic approach is used, including:

Data Collection: Sourcing, anonymizing, and preprocessing a diverse set of handwritten characters and numbers.

Technologies Used: The use of advanced tools and libraries like Python, Tesseract, OpenCV, and EAST for OCR implementation.

Model Training and Evaluation: Developing and evaluating models using rigorous training processes and performance metrics.

Data Collection

We imported two datasets for the training phase, one for numbers (MNIST 0-9 from Keras) and one for letters (Kaggle). These datasets consist of handwritten numbers and letters, with over 70,000 images of numbers and 372,550 letter images. The datasets were imported and installed as follows:

For numbers: Keras – MNIST 0-9

For letters: Kaggle A-Z

Data Preprocessing

The raw data collected from healthcare institutions goes through several preprocessing and annotation steps to prepare it for model training. These steps are crucial for improving the image quality and creating a true dataset for accurate model evaluation:

- *Digitization and Scanning*: All handwritten characters are scanned using high-resolution scanners to convert them into digital images, ensuring clarity and suitability for OCR processing.
- *Image Preprocessing*: Scanned images undergo various preprocessing techniques to enhance their quality and readability, such as:
  - *Binarization*: Converting images to a binary (black and white) format to improve text visibility.
  - *Noise Reduction*: Removing noise and background artifacts to ensure clean text areas.
  - *Image Normalization*: Standardizing image sizes and resolutions to ensure uniformity across the dataset.
- *Manual Annotation*: Each character is manually labeled to create a true dataset, including:

    - *Text Labeling*: Manually transcribing handwritten text into digital format, ensuring accuracy and consistency.
    - *Zone Marking*: Identifying and marking areas of interest within the images.
- *Quality Assurance*: A quality assurance process is implemented to verify the accuracy of the annotations.

Data Augmentation

To increase dataset diversity and improve the model's ability to generalize, data augmentation techniques are applied. These techniques simulate variations that the model might encounter in real-world scenarios. Augmentation methods include:

- *Rotation*: Rotating images to simulate different writing angles.
- *Resizing*: Adjusting image sizes to account for variations in character formats.
- *Image Cropping*: Randomly cropping parts of images to imitate occlusions and partial text.
- *Lighting and Contrast*: Modifying lighting and contrast to reflect different scanning conditions.

Technologies Used

This section details the technologies utilized in the project to digitize handwritten medical prescriptions using **Optical Character Recognition (OCR)**. The chosen technologies provide a powerful, efficient, and accurate system for OCR, combining advanced image processing and deep learning tools.

- *Python*: Python was the primary programming language due to its simplicity, versatility, and extensive ecosystem of libraries for machine learning and image processing. It facilitates fast development and testing, with access to powerful libraries like **TensorFlow, Keras, OpenCV, and Tesseract**.
- *Tesseract*: Tesseract is an open-source OCR engine known for its accuracy in recognizing both printed and handwritten text. Initially developed by HP and now maintained by Google, it supports 166 languages, including Albanian. It integrates easily with Python using pytesseract and allows for customizations to improve recognition accuracy.
- *OpenCV*: OpenCV is a widely used open-source computer vision library for image processing tasks. In this study, it aids in preprocessing images for OCR, handling tasks like **binarization**, **noise reduction**, and **normalization**.

Combined with Tesseract, OpenCV enhances text recognition from scanned images.
- *EAST (Efficient and Accurate Scene Text Detector)*: EAST is a deep learning-based text detector designed for identifying text in complex scenes. It is highly accurate, fast, and suitable for real-time applications. In this project, EAST helps detect text areas in images, allowing Tesseract to accurately recognize and convert them into structured digital formats.

Training and Evaluation

Training and evaluation are critical to ensuring the success of the OCR system in recognizing handwritten medical prescriptions. This section outlines the detailed processes of preparing the dataset, training the model, evaluating performance, and validating the results to ensure accuracy and reliability.

The training process involves several key steps to ensure the models learn effectively from the data and generalize well. The main focus is on optimizing the **EAST text detector** and **Tesseract OCR engine** using a **Convolutional Neural Network (CNN)** architecture.

- *Dataset Preparation*:
  - **Training Set**: Consists of 2,700 images, combining digits (0-9 from MNIST) and English alphabets (a-z from Kaggle). These images are repeated 20 times to enhance training coverage.
  - **Validation Set**: A subset used to adjust hyperparameters and prevent overfitting.
  - **Test Set**: A separate dataset reserved for final evaluation, ensuring unbiased model assessment.

Model Training:

- *EAST Text Detector*:
  - Uses a CNN architecture designed to detect text at the pixel level.
  - **Loss Function**: Combines classification loss (text/non-text regions) and regression loss (bounding box coordinates).
  - **Hyperparameters**: Includes learning rate, batch size, and epochs, optimized through grid search and cross-validation.
- *CNN Layers*:
  - Involves convolutional layers for feature extraction, pooling layers for down-sampling, and fully connected layers for classification.
  - **Activation Functions**: Uses ReLU for non-linearity and softmax for output classification.

- **Optimizer**: Adam is used to fine-tune learning rate and improve convergence.
- *Tesseract OCR*:
  - **Pre-trained** on a large corpus of printed and handwritten text.
  - Further trained on specific datasets to adapt to different handwriting styles.
  - Custom configurations are applied for segmentation and recognition of variable writing styles.

Several evaluation metrics are used to assess the performance of OCR models:

- *Accuracy*: Measures the proportion of correctly recognized text instances out of the total.
- *Precision*: Indicates the proportion of correctly recognized text out of all recognized instances, reflecting the model's ability to avoid false positives.
- *Recall*: Measures the model's ability to detect all true text instances.
- *F1-Score*: Harmonic means of precision and recall, balancing both aspects.
- *Character Error Rate (CER)*: The ratio of incorrect characters to the total number of characters, reflecting detailed recognition accuracy.
- *Word Error Rate (WER)*: The ratio of incorrect words to the total number of words, providing overall text recognition accuracy.

Methods and analysis

Preprocessing Scripts

Preprocessing is an essential step in the image analysis and computer vision tasks pipeline. It involves transforming raw image data into a format more suitable for further processing and analysis. This section covers the basic steps required to preprocess images using OpenCV, a well-known library for computer vision applications. The following steps are critical for effective preprocessing:

- *Loading Images*: Images are loaded into the program from the file system.
- *Converting to Grayscale*: The loaded images are converted to grayscale to reduce complexity and focus on intensity information.
- *Thresholding and Binarization*: Thresholding techniques are applied to convert grayscale images into binary images, making the object of interest and background easily distinguishable.
- *Noise Reduction*: Noise is removed from the binary images to improve the accuracy of further processing steps.

- *Normalization*: Pixel values are normalized to a standard scale to ensure consistency across different images.

Thresholding

Thresholding is an image segmentation method. For example, in an image containing both a dog and a tree, we need to extract the dog. This is done by segmenting the image into areas of interest and non-interest using a point called the threshold.

- **0 - Black:** RGB (0, 0, 0) - black
- **255 - White:** RGB (255, 255, 255) - white

When applying thresholding, we filter the pixels. For instance, with a threshold of 127, pixels with values >= 127 will turn white, while those with values less than 127 will turn black (0). This technique is called binarization, meaning we only have two values: 0 for black and 1 for white. Binarization simplifies the image, making it easier for the algorithm to detect objects, such as text, by reducing complexity.

Gaussian: Uses the mean along with the standard deviation, applying convolution. This technique produces better results by smoothing out noise and making thresholding more adaptive to variations in the image. Gaussian thresholding dynamically adjusts the threshold based on the local region, improving accuracy, especially in images with uneven lighting or noise.

We have chosen **Gaussian Thresholding** because it provides stable and smooth results in noisy images by using the mean and standard deviation for better processing. This method adjusts the threshold dynamically across different regions of the image, making it particularly effective in handling variations in lighting or noise, resulting in more refined segmentation.

Noise Reduction

Morphological operations are techniques used for noise removal, edge detection, and image enhancement. These methods are applied exclusively to binary images (black and white), where 0 represents black and 1 represents white.

Main Operations:

- *Erosion and Dilation*:
  - **Erosion:** Removes pixels, making the image thinner (e.g., thinning text).
  - **Dilation:** Adds pixels, making the image bolder (e.g., emphasizing text).

- *Opening*:
  - This involves applying erosion followed by dilation. First, pixels are removed, then new pixels are added.
  - **Usage:** Mainly used for noise removal. During erosion, unwanted pixels are removed, thinning the text. Dilation then restores the text to its original shape, enlarging the edges.
- *Closing*:
  - This involves applying dilation followed by erosion. First, new pixels are added, and then some are removed.
  - **Usage:** Useful for removing noise within the image (e.g., colored spots inside a letter). When dilation is applied, these spots disappear, and the image edges are enlarged. Erosion is then applied to remove excess pixels.

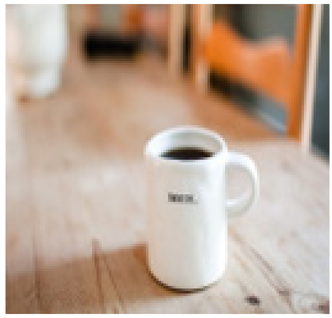
Text Detection Using EAST

In this section, we will explore the process of detecting text within images using the **Efficient and Accurate Scene Text (EAST)** detector. This method is particularly effective for detecting text in complex scenes, as it leverages a deep learning model to accurately identify text areas. The EAST model provides a robust solution for real-time applications due to its efficiency and precision.

The first step in using the EAST detector is loading the pre-trained model. The EAST model is available as a pre-trained neural network in OpenCV, making it easy to integrate into your text detection pipeline.

Once the model is loaded, the next step is processing the images to detect text regions. This involves creating a **blob** from the image, inputting it into the network, and feeding it forward through the network to retrieve the text regions. The images should be resized to dimensions of **320 x 320 (width x height)** for optimal detection. If the images are of a different size, resizing must be performed to ensure consistent input dimensions.

FIGURE 13: Region of interest detection



Pre-processing

Processing

Text Recognition with Tesseract

In this section, we will examine the process of recognizing text from images using **Tesseract**, an open-source OCR (Optical Character Recognition) engine known for its accuracy and flexibility. After detecting text regions with EAST, the next step is extracting and recognizing the textual content within these regions.

Tesseract allows us to configure various parameters to meet the specific needs of our dataset, ensuring more accurate text recognition. This customization is key to achieving high performance, especially when working with different fonts, languages, or noisy images.

FIGURE 15: Detektimi i tekstit

Training the OCR Model

This section describes the process of training an Optical Character Recognition (OCR) model. The training was performed using a **Convolutional Neural Network (CNN)** model and two well-known datasets: **MNIST** (containing digits from 0 to 9) and **Kaggle A-Z** (containing letters from A to Z). A total of **2700 images** from these datasets were used for training, with each image being augmented 20 times to expand the training dataset and improve results.

This section will detail the following steps:

- **Dataset Preparation:** Organizing and preprocessing the MNIST and Kaggle A-Z datasets for training.
- **CNN Model Configuration:** Setting up the CNN architecture to effectively recognize characters.
- **Training Process:** Feeding the model with the training data and running it through multiple epochs.
- **Validation Methods:** Ensuring the model's performance is tested and validated to generalize well to new data.

Through this process, the goal is to achieve an OCR model capable of recognizing characters from the given datasets with high accuracy and reliability.

Trajnimi i rrjetit neural

FIGURE 16: Error Graph

Output:

	precision	recall	f1-score	support
0	0.40	0.52	0.45	1381
1	0.97	0.99	0.98	1575
2	0.90	0.98	0.94	1398
3	0.96	0.99	0.98	1428
4	0.90	0.97	0.93	1365
5	0.78	0.91	0.84	1263
6	0.97	0.96	0.97	1375
7	0.96	0.99	0.97	1459
8	0.96	0.98	0.97	1365
9	0.95	0.99	0.97	1392
A	1.00	0.99	0.99	2774
B	0.99	0.98	0.99	1734
C	0.99	0.98	0.99	4682
D	0.87	0.98	0.92	2027
E	1.00	0.98	0.99	2288
F	0.97	1.00	0.98	233
G	0.91	0.96	0.94	1152
H	0.97	0.96	0.97	1444
I	0.99	0.98	0.99	224
J	0.99	0.97	0.98	1698
K	0.97	0.99	0.98	1121
L	0.97	0.98	0.97	2317
M	0.97	1.00	0.98	2467
N	0.99	0.97	0.98	3802
O	0.94	0.88	0.91	11565
P	1.00	0.98	0.99	3868
Q	0.97	0.97	0.97	1162
R	0.99	0.99	0.99	2313
S	0.99	0.95	0.97	9684
T	1.00	0.98	0.99	4499
U	0.98	0.97	0.98	5801
V	0.86	1.00	0.93	836
W	0.97	0.99	0.98	2157
X	0.98	0.99	0.99	1254
Y	0.98	0.93	0.96	2172
Z	0.97	0.92	0.95	1215
accuracy			0.95	88490
macro avg	0.94	0.96	0.95	88490
weighted avg	0.96	0.95	0.96	88490

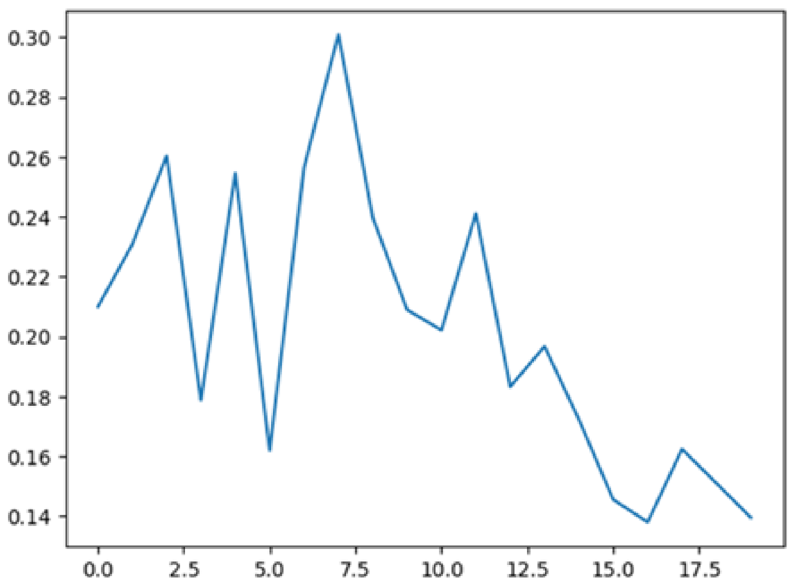
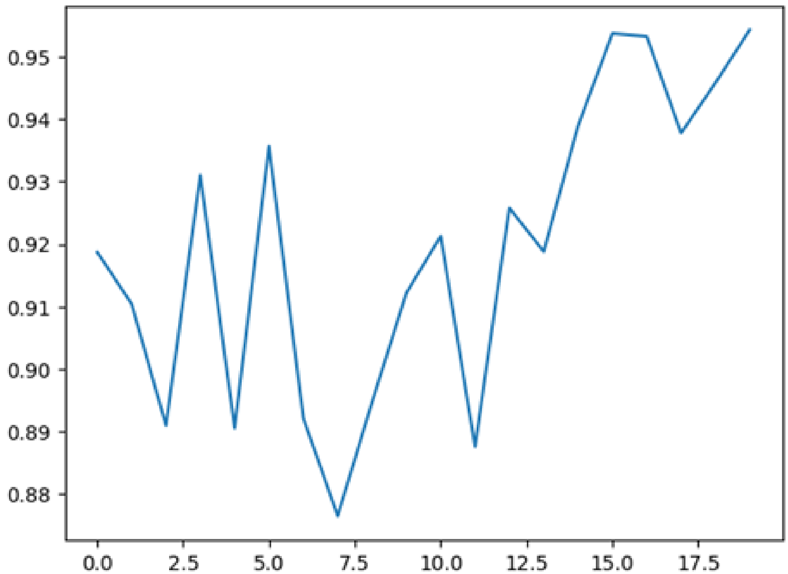


FIGURE 17: Graph of Accuracy



Results

In this chapter, we will explain the application of the trained OCR model to digitize handwritten medical prescriptions from doctors. By utilizing the trained model, we will test its performance on an actual doctor's prescription to evaluate its accuracy and effectiveness in real-world scenarios. This testing will help assess how well the model can interpret and convert handwritten text into digital format, particularly focusing on the unique challenges posed by messy handwriting, abbreviations, and medical terminology typically found in prescriptions.

FIGURE 18: Test image

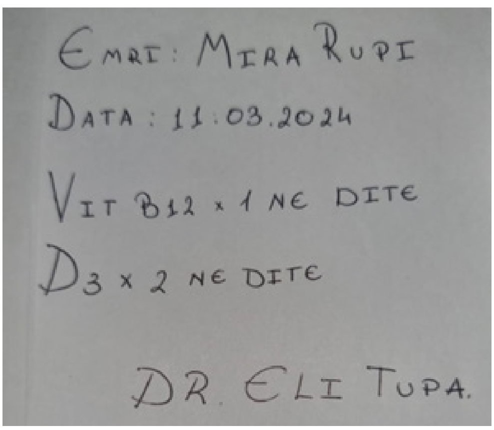
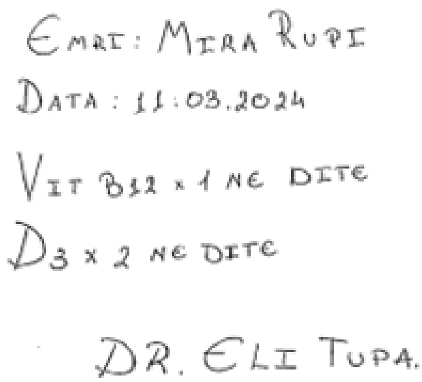
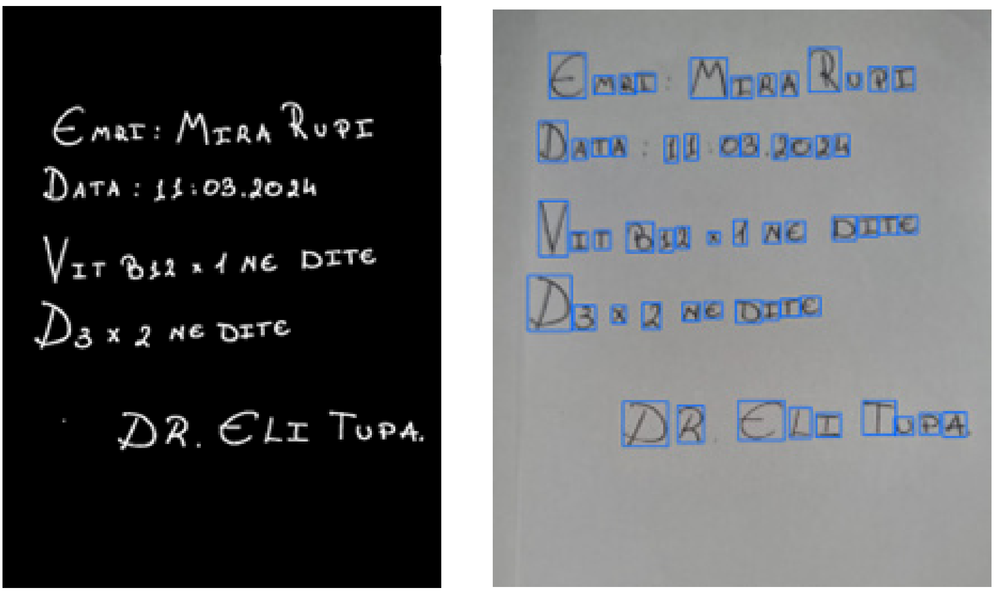


FIGURE 19: Preprocessing of the image



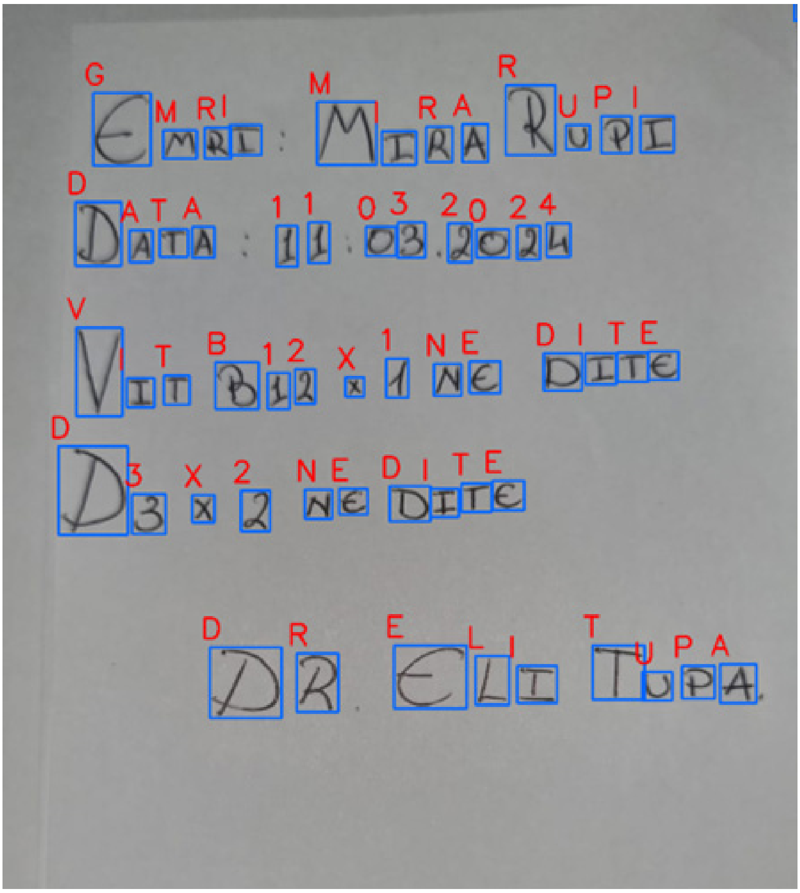
We have separated the text from the background, then we invert the colors so the background is black and text in white because that's how the neural network is trained. After we apply Dialation to make the text more highlighted.

FIGURE 21: Confidence



Letter detection

```
D  -> 100.0
D  -> 100.0
V  -> 99.92490410804749
G  -> 82.33226537704468
I  -> 96.439129114151
A  -> 99.99995231628418
3  -> 99.99985694885254
T  -> 99.99998807907104
M  -> 99.9996423721313
T  -> 99.99971389770508
X  -> 99.99973773956299
A  -> 99.9982237815857
R  -> 99.94874000549316
D  -> 99.99998807907104
B  -> 73.43519330024719
I  -> 99.93749260902405
2  -> 99.98515844345093
1  -> 99.72720742225647
1  -> 75.63432455062866
2  -> 99.46190118789673
R  -> 99.801105260849
N  -> 97.13206887245178
1  -> 97.20686078071594
M  -> 99.99983310699463
E  -> 99.99464750289917
X  -> 99.99959468841553
O  -> 94.72241997718811
I  -> 99.87940788269043
1  -> 99.878913164138
D  -> 99.97654557228088
E  -> 95.24114727973938
3  -> 99.67645406723022
R  -> 97.85822033882141
I  -> 98.98114204406738
N  -> 99.81375932693481
2  -> 99.997079372406
T  -> 99.9996423721313
A  -> 99.99512434005737
E  -> 99.95399713516235
L  -> 99.99998807907104
O  -> 76.2120246887207
E  -> 98.24574589729309
R  -> 99.87449645996094
I  -> 98.05420637130737
2  -> 99.99631643295288
D  -> 99.99985694885254
4  -> 83.85692238807678
U  -> 99.70290064811707
I  -> 99.968206882476
T  -> 99.99980926513672
P  -> 88.06697130203247
T  -> 99.98717308044434
I  -> 99.99995231628418
U  -> 96.91241979598999
E  -> 99.71449971199036
P  -> 99.75659847259521
A  -> 77.09456682205
1  -> 99.97900128364563
```



Conclusion

This study explores the application of Deep Learning technologies, with a focus on Convolutional Neural Networks (CNNs), to enhance the accuracy and efficiency of Optical Character Recognition (OCR) for handwritten medical prescriptions. The research demonstrates that combining CNNs, which extract complex features from images, with Recurrent Neural Networks (RNNs), which manage long-term dependencies in character sequences, significantly improves OCR performance. By training models on diverse datasets and using data augmentation techniques, the study achieved a CNN model accuracy of 95.28%, with only one character error, marking a substantial improvement over traditional OCR method.

The results indicate that deep learning models not only handle variations in handwriting more effectively but also automate the OCR process faster and more reliably. This research confirms the transformative potential of deep learning technologies for digitizing handwritten documents, especially in the medical field.

Furthermore, the study suggests that such technologies can be applied to improve the accuracy and efficiency of OCR in other areas beyond medical prescriptions, offering widespread practical applications.

Bibliography

1. Caceres, P. (2020). The Convolutional Neural Network - Theory and Implementation of LeNet-5 and AlexNet. Available at: pabloinsente.github.io
2. Dilmegani, C. (2024). State of OCR in 2024: Is it dead or a solved problem?
Available at: https://research.aimultiple.com/ocr-technology/
3. Fateh, A., Fateh, M., & Abolghasemi, V. (2023). Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. Engineering Reports. https://doi.org/10.1002/eng2.12832
4. Hai, N. (2022). How to verify if the image contains noise in background before 'OCR' ing.
Available at:
https://stackoverflow.com/questions/70604177/how-to-verify-if-the-image-contains-noise-in-background-before-ocr-ing
5. Kasem, S.E. (2023). Advancements and Challenges in Arabic Optical Character Recognition: A Comprehensive Survey. Available at: https://arxiv.org/pdf/2312.11812
6. Nockels, J., Gooding, P., Ames, S., & Terras, M. (2022). Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research. Archival Science, 22(3), 367–392. https://doi.org/10.1007/s10502-022-09397-0.
7. Zhao, Z., Jiang, M., Guo, S., Wang, Z., Chao, F., & Tan, K. C. (2020). Improving Deep Learning based Optical Character Recognition via Neural Architecture Search. In 2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings (pp. 1-7). Article 9185798 (2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CEC48606.2020.9185798