Decognize: Prescription Digitization Using Knowledge Graphs

Group Members:
Muhammad Sherjeel Akhtar (P20-0101)
Mahad Ashraf (P20-0563)

Supervisor:
Mr. Muhammad Shoaib Khan

Table of contents

1. Literature Review

LiteratureReview

Sr. no	Year	Basic Idea	Methodologies	Results	Limitations
[1]	2023	OCR with Open CV and tesseract	Implemented Tesseract OCR with Open CV in python. Focusing on image pre-processing for optimal results integrated text detection and recognition components	Achieved satisfactory OCR accuracy with well preprocessed images. However, Tesseract struggled with complex backgrounds and artifacts, yielding suboptimal outputs.	Tesseracts accuracy is hindered by poor image quality. Requiring meticulous preprocessing. Challenges arise in handling artifacts handwriting and diverse languages.
[2]	2021	Optical Character Recognition Using TensorFlow	Implemented OCR with TensorFlow Enhanced model robustness with data augmentation technique. Implemented a custom ResNet architecture for OCR	These results showcase the effectiveness of the OCR model, particularly in accurately recognizing characters within the test set, demonstrating its robustness and suitability for the specified task.	Our Model can fail if the image is complex . E.g cursive writing images or images with Continous Characters Currently our model is trained only on digits and English language
[3]	2021	Construct a Bio Medical Knowledge Graph with NLP	Extracted text from biomedical document using OCR and applied BERN and utilized zerorelation extractor.	Successfully established a Neo4j knowledge graph, showcasing versatility through demonstrated applications such as search engine, co-occurrence analysis and author expertise inspection. While emphasizing its utility for diverse biomedical machine learning applications.	Limitations include persistent NER challenges with BERN, potential inaccuracies in the zero shot relation extractor and the need for exper validation with external database enrichment reliant ondata consistency
[4]	2018	Build a Handwritten Text Recognition System using TensorFlow	Implemented HTR using TensorFlow, with NN trained on IAM word-images, including CNN, RNN and CTC layers. Preprocessed data with resizing normalization and potential augmentation. Utilized RMSProp for training and explored enhancements like data augmentation, input size adjustments and decoding strategies.	Implemented sucessful HTR on IAM word-images, enabling flexible NN customization and identifying areas for accuracy improvements.	Limited Diversity due to reliance on IAM dataset Potential recognition errors especially for non-dictionary words CPU based training may be slower: GPU recommended
[5]	2022	Doctor Handwritten Prescription recognition system in multilanguage using deep learning	Implemented a system employing machine learning techniques such as CNNs, RNNs, LSTMs for recognizing and translating handwritten prescription notes i diverse language	Successful recognition and translation of handwritten prescriptions in various languages. Demonstrated the efficiency of CNNs, RNNs, and LSTMS in multilingual handwritten text processing.	Sensitivity to variations in handwriting styles. Reliance on quality and diversity of training data for optimal performance
[6]	2022	A Comparison of various Machine learning Algorithms for recognizing Text on Medical prescriptions	Proposed approach involves image scanning pre-processing and CNN-based feature extraction for	Successful implementation of CNN-based recognition for medical prescription. Need for	Limited Exploration of alternative machine learning algorithms

2. Problem Statement

Problem Statement

- **Problem**: Inefficient healthcare data management for prescriptions.
- **Challenge**: Illegible handwriting , medical jargon and Knowledge Graph
- **Consequence**: Errors in healthcare due to traditional OCR systems.
- **Goal**: Develop NLP-based system for accurate prescription transcription

3. System Diagram

System Diagram

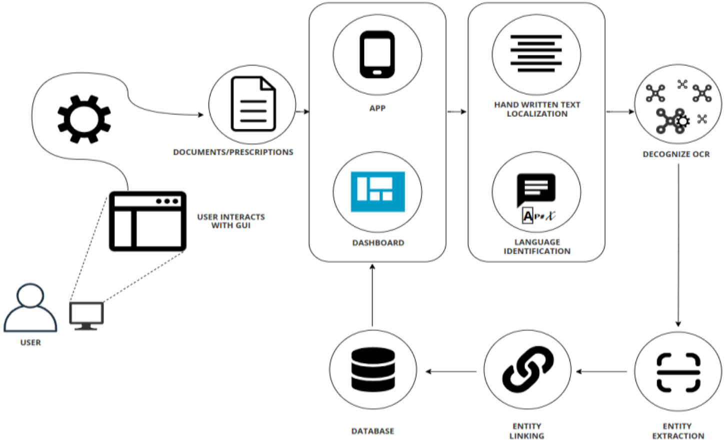


Figure 1: Architecture Diagram of DeCognize

4. UML Diagrams

Use Case Diagram

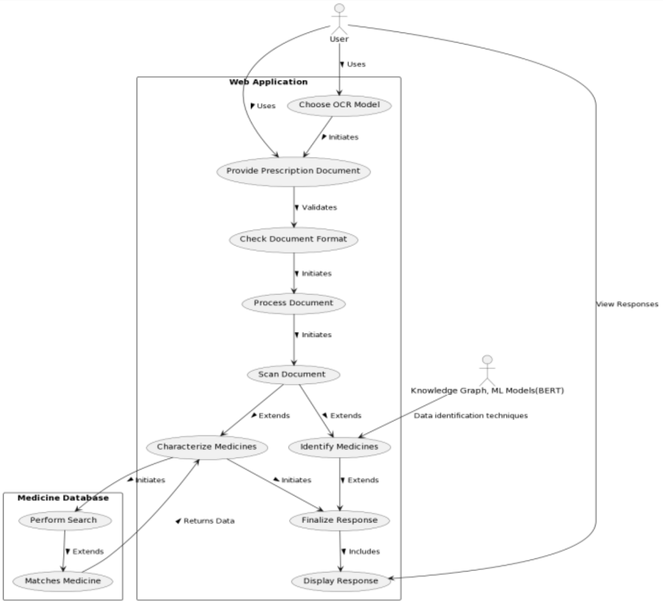


Figure 2: Use Case Diagram of DeCognize

Activity Diagram

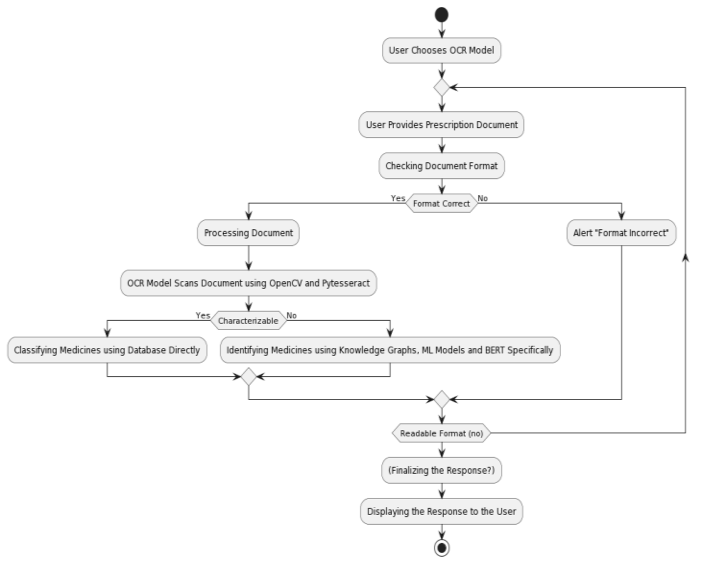


Figure 3: Activity Diagram of DeCognize

Swimline Diagram

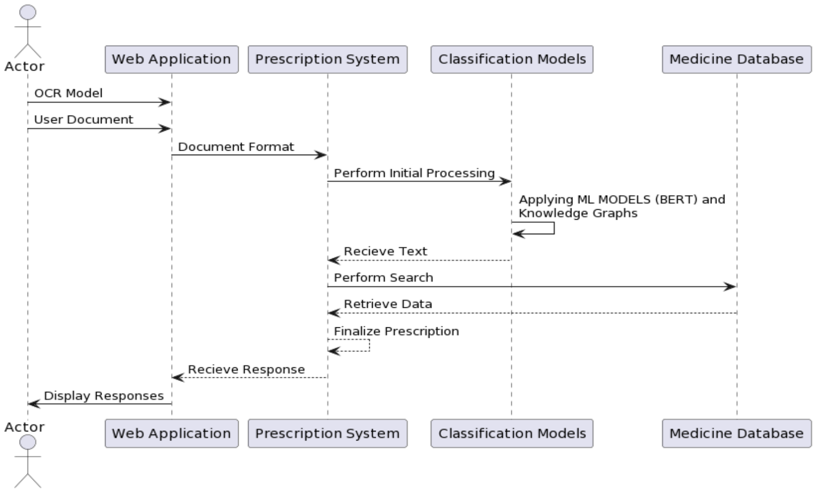


Figure 3: Swimline Diagram of DeCognize

Flow Diagram

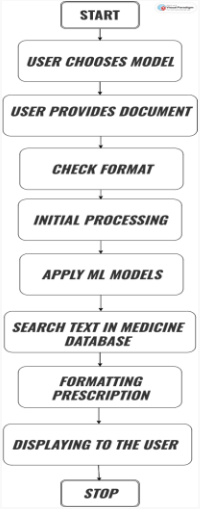


Figure 3: Flow Diagram of DeCognize

5. Objectives

Objectives

- To reduce error percentage in reading prescriptions.
- To create an improved OCR system which could later on deployed on other real-life-domains as well.
- To allow user to save and access their prescription data conveniently.

6. Expected Output

CODE Result

```
import cv2
import pytesseract

pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"

# Reading image
img = cv2.imread("sample.png")

# Convert to RGB
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Use pytesseract to detect and print text
custom_config = r'--oem 3 --psm 6'
texts = pytesseract.image_to_string(img_rgb, config=custom_config)
print("Texts:", texts)

# Save the text to a file
output_file_path = "output.txt"
with open(output_file_path, "w", encoding="utf-8") as text_file:
    text_file.write(texts)

# Use pytesseract to get bounding boxes
boxes = pytesseract.image_to_boxes(img_rgb, config=custom_config)

# Draw bounding boxes on the image
for b in boxes.splitlines():
    b = b.split()
    x, y, w, h = int(b[1]), int(b[2]), int(b[3]), int(b[4])
    img_rgb = cv2.rectangle(img_rgb, (x, img_rgb.shape[0] - y), (w, img_rgb.shape[0] - h), (0, 255, 0), 2)

# Show the image with bounding boxes
cv2.imshow("Output", img_rgb)
cv2.waitKey(0)
cv2.destroyAllWindows()

print(f"Texts saved to {output_file_path}")
```

Figure4: Expected Output

Code output

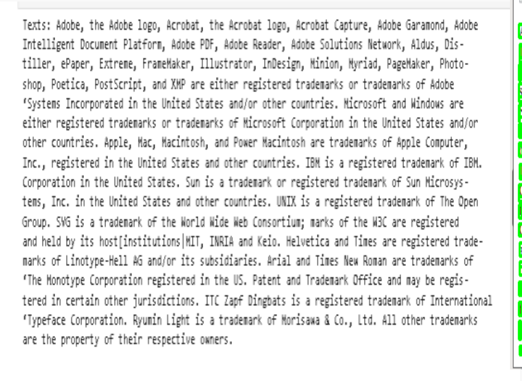
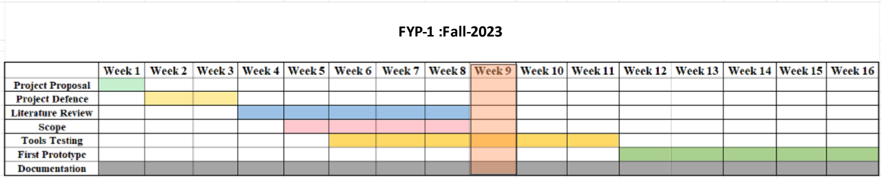

Texts: Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Capture, Adobe Garamond, Adobe Intelligent Document Platform, Adobe PDF, Adobe Reader, Adobe Solutions Network, Aldus, Distiller, ePaper, Extreme, FrameMaker, Illustrator, InDesign, Minion, Myriad, PageMaker, Photoshop, Poetica, PostScript, and XMP are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Apple, Mac, Macintosh, and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. IBM is a registered trademark of IBM. Corporation in the United States. Sun is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of The Open Group. SVG is a trademark of the World Wide Web Consortium; marks of the W3C are registered and held by its host[institutions|MIT, INRIA and Keio. Helvetica and Times are registered trademarks of Linotype-Hell AG and/or its subsidiaries. Arial and Times New Roman are trademarks of The Monotype Corporation registered in the US. Patent and Trademark Office and may be registered in certain other jurisdictions. ITC Zapf Dingbats is a registered trademark of International TypeFace Corporation. Ryumin Light is a trademark of Morisawa & Co., Ltd. All other trademarks are the property of their respective owners.

Figure4: Expected Output

7. Gantt Chart

Gantt Chart



FYP-1 :Fall-2023

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16
Project Proposal																
Project Defence																
Literature Review																
Scope																
Tools Testing																
First Prototype																
Documentation																

Figure5: Gantt Chart

8. References

- Filip Zelic and Anuj Sable. A review on on OCR with Tesseract OpenCV and Python. Nanonets, 2023.
- Kamlesh Solanki. A review on optical character recognition using tensor flow. Medium, 10:39154-39176, 2021.
- Tomaz Bratanic , D. Kim *et al.*, "A Neural Named Entity Recognition and Multi-Type Normalization Tool for Biomedical Text Mining," in *IEEE Access*,Medium vol. 7, pp. 73729–73740, 2019, doi: 10.1109/ACCESS.2019.2920708. , 2021

References [2]

- Harald Scheidl. Article on Build a handwritten text recognition using tensorflow. Medium, 9:87643-87662, 2018.
- Kamalanaban, E., M. Gopinath, and S. Premkumar. "Medicine box: Doctor's prescription recognition using deep machine learning." International Journal of Engineering and Technology (UAE) 7 (2018): 114-117.
- Sandhya, P., and K. P. Rama Prabha. "Comparison Of Various Machine Learning Algorithms For Recognizing Text On The Medical Prescriptions." Journal of Pharmaceutical Negative Results (2022): 2083-2091.

References [3]

- Tabassum, Shaira, Nuren Abedin, Md Mahmudur Rahman, Md Moshiur Rahman, Mostafa Taufiq Ahmed, Rafiqul Islam, and Ashir Ahmed. "An online cursive handwritten medical words recognition system for busy doctors in developing countries for ensuring efficient healthcare service delivery." Scientific reports 12, no. 1 (2022): 1-13
- Hassan, Esraa, Habiba Tarek, Mai Hazem, Shaza Bahnacy, Lobna Shaheen, and Walaa H. Elashmwai. "Medical prescription recognition using machine learning." In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0973-0979. IEEE, 2021.
- Wijewardena, W. R. A. D. "Medical Prescription Identification Solution." PhD diss., 2021