Decognize: Prescription Digitization Using Knowledge Graphs

Group Members:

Muhammad Sharjeel Akhtar (P20-0101)

Mahad Ashraf (P20-0563)

Supervisor:

Mr. Muhammad Shoaib Khan

Table of contents

1.Project Objective

Project Objective

- To reduce error percentage in prescriptions readability.
- 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.
- Generate a user-friendly output that provides a clear and organized list of recognized medications with recommended dosages.
- Utilize deep learning techniques, including TensorFlow, Keras, and OpenCV, to process and detect characters in illegible handwritten texts.

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. Architecture Design

System Diagram

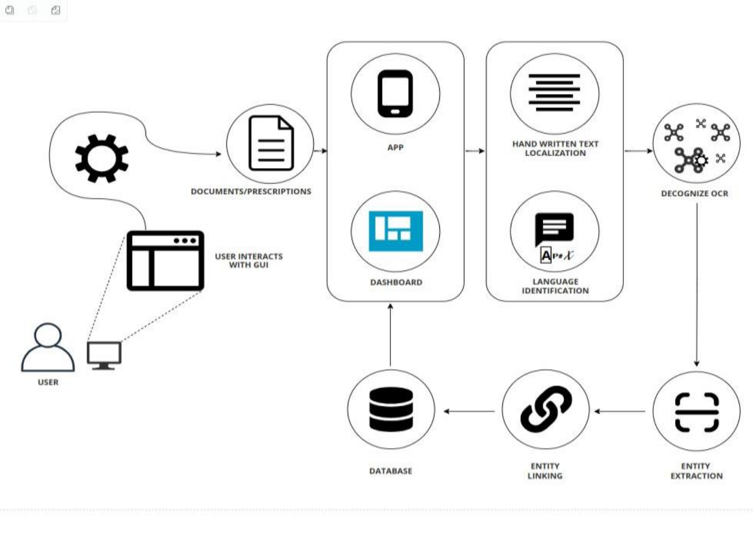


Figure1: Architecture Diagram of DeCognize

4. Activity Diagram

Activity Diagram

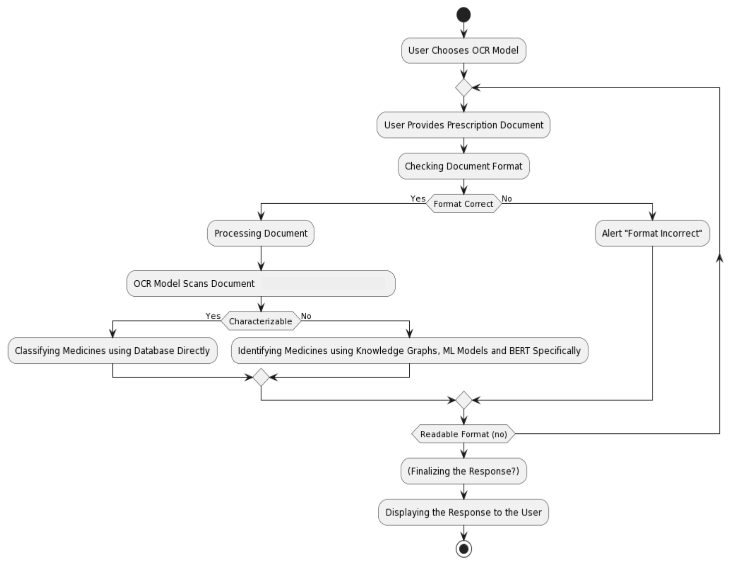

User Chooses OCR Model
User Provides Prescription Document
Checking Document Format
Yes
Format Correct
No
Processing Document
Alert "Format Incorrect"
OCR Model Scans Document
Yes
Characterizable
No
Classifying Medicines using Database Directly
Identifying Medicines using Knowledge Graphs, ML Models and BERT Specifically
Readable Format (no)
(Finalizing the Response?)
Displaying the Response to the User

Figure 3: Activity Diagram of DeCognize

5. Sequence Diagram

Sequence Diagram

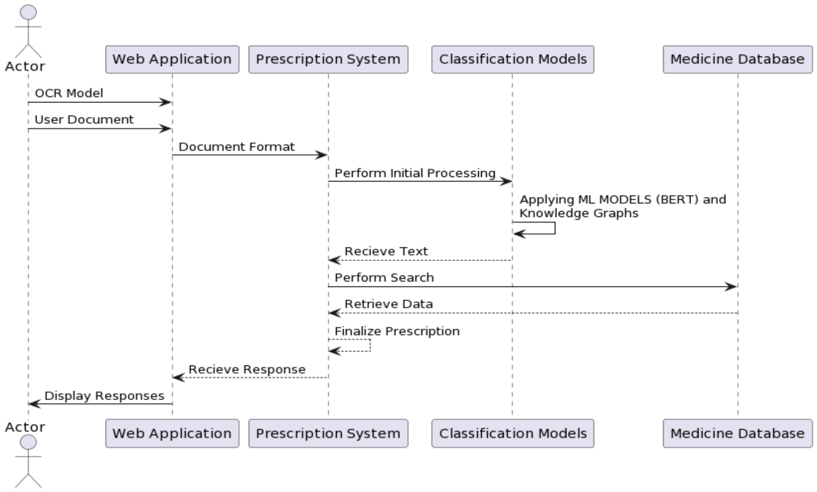


Figure3: Swimlane Diagram of DeCognize

6. Layered Diagram

Layered Diagram

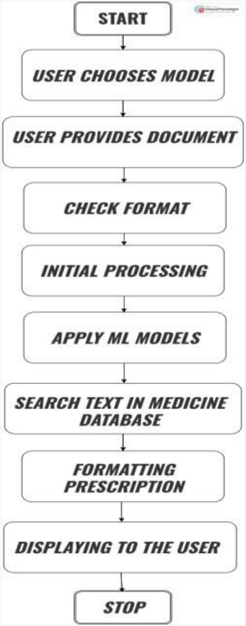


Figure 3: Layered Diagram of DeCognize

7. Flow Diagram

Flow Diagram

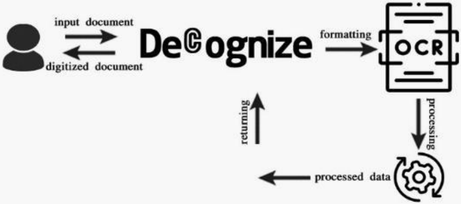


Figure3: Flow Diagram of DeCognize

8. Use Case Diagram

Structured Use Case Diagram

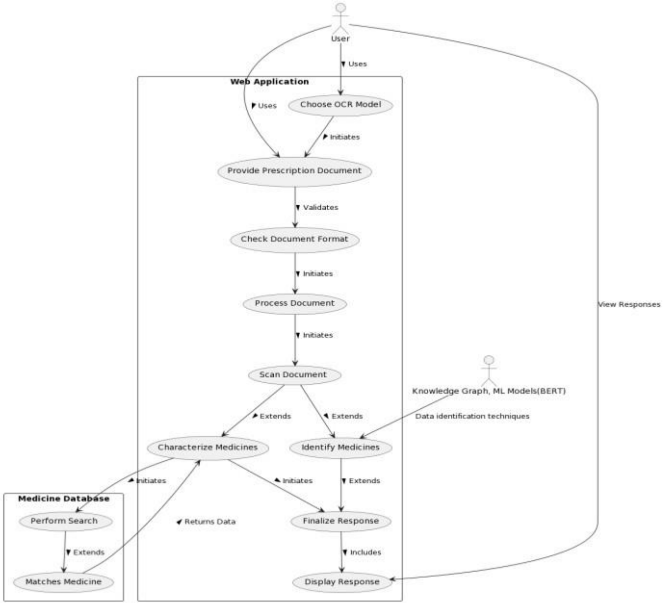


Figure: Structured Use Case Diagram of DeCognize

9. State Machine Diagram

State Machine Diagram

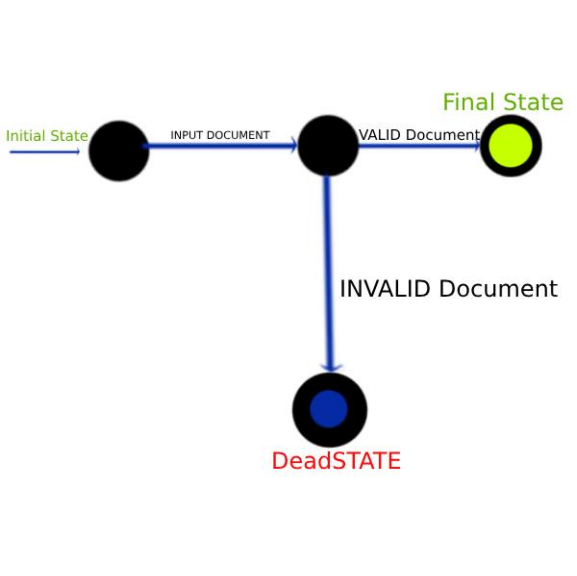


Figure: State Machine Diagram of DeCognize

10. Data Flow Diagram

Data Flow Diagram

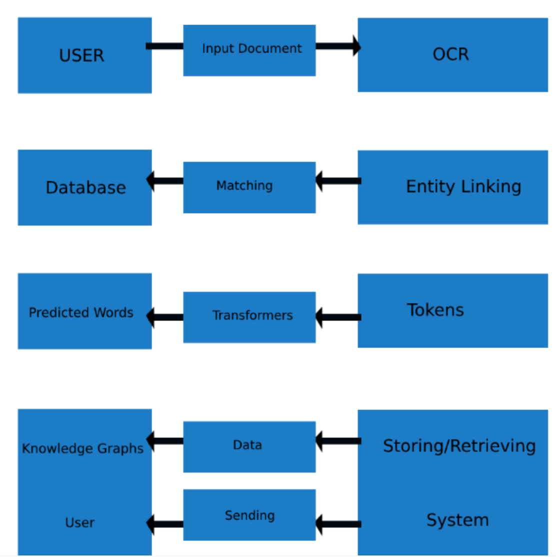


Figure: Data Flow Diagram of DeCognize

11. Component Diagram

Component Diagram

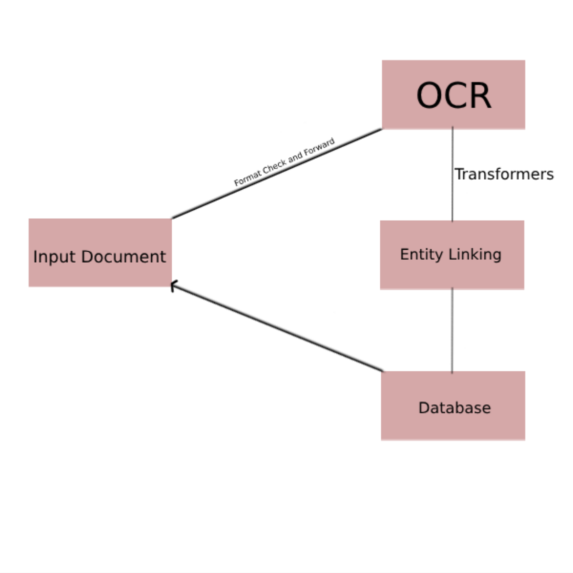
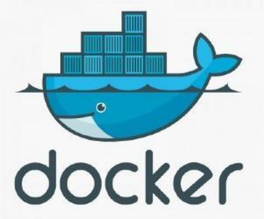


Figure: Component Diagram of DeCognize

Technologies Used



12.Business Opportunity

Business Opportunity

1. **Efficient Data Digitization:** Streamline conversion of handwritten medical notes to digital records for time savings and reduced errors.

2. **Enhanced Decision-Making:** Improve data accuracy to facilitate quicker, informed healthcare decisions, ultimately enhancing patient care.

3. **Research and Efficiency:** Utilize biomedical knowledge graphs to enable advanced research and analytics, driving innovation in the medical field.

4. **Cost Savings Automate:** Manual data entry, leading to significant operational cost reductions for healthcare institutions.

13.Project Scope

Project Scope

- **Global Market Growth:** The global OCR market is experiencing rapid growth, showcasing its significance in addressing diverse industry needs. OCR market was valued at USD 8.93 billion A Compound Annual Growth Rate (CAGR) of 15.4% is anticipated between 2022 and 2030

- **Recent Projects:** Automated doctor prescription by Nano Net Technologies Inc and Neurodata Group.

- **OCR in Healthcare in Pakistan:** Active research by Seerat Rani, Abd Ur Rehman, Beenish Yousaf, Hafiz Tayyab Rauf, Emad Abouel Nasr, and Seifedine Kadry.

- **Summary:** OCR enhancing healthcare in Pakistan through innovation and integration.

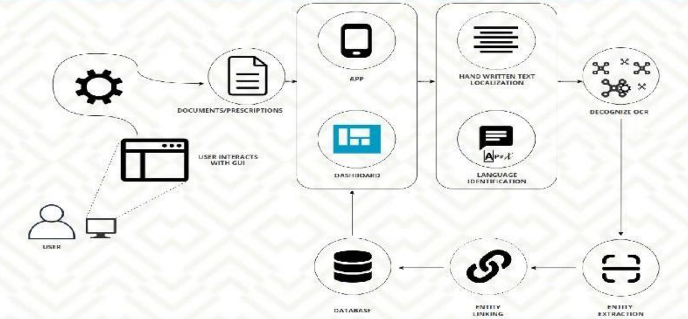
14. Poster

DeCognize

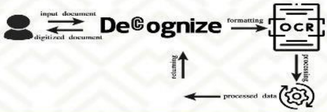
Prescription Digitization
Using Knowledge Graphs

OUR METHODOLOGY INCLUDES EXTRACTING TEXT FROM DOCUMENTS AND THEN PARSING CRUCIAL ENTITIES TO ORGANIZE INFORMATION IN A STRUCTURED MANNER. THIS STRUCTURED FORMAT EMPOWERS ORGANIZATIONS TO ANALYZE THE DATA EFFECTIVELY, FACILITATING DATA-DRIVEN DECISIONS BASED ON THE DOCUMENT INFORMATION.

ARCHITECTURE



FLOW

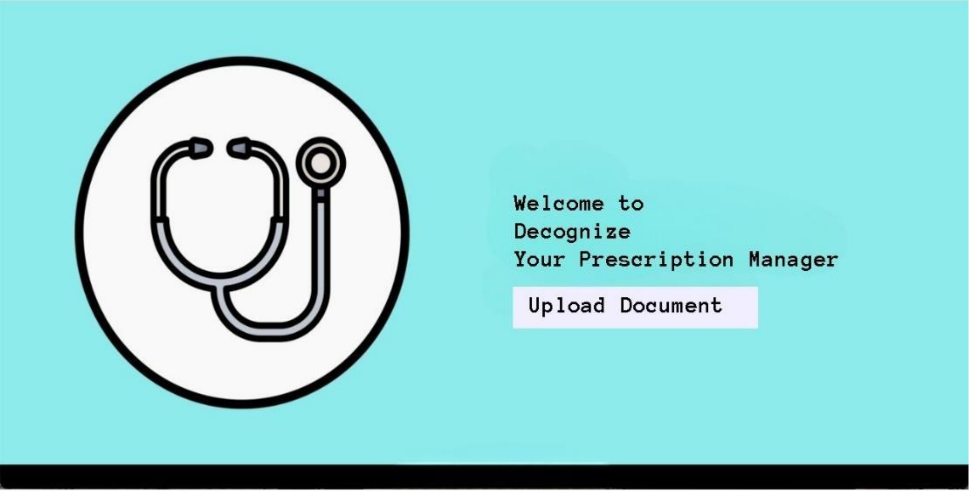


GOAL

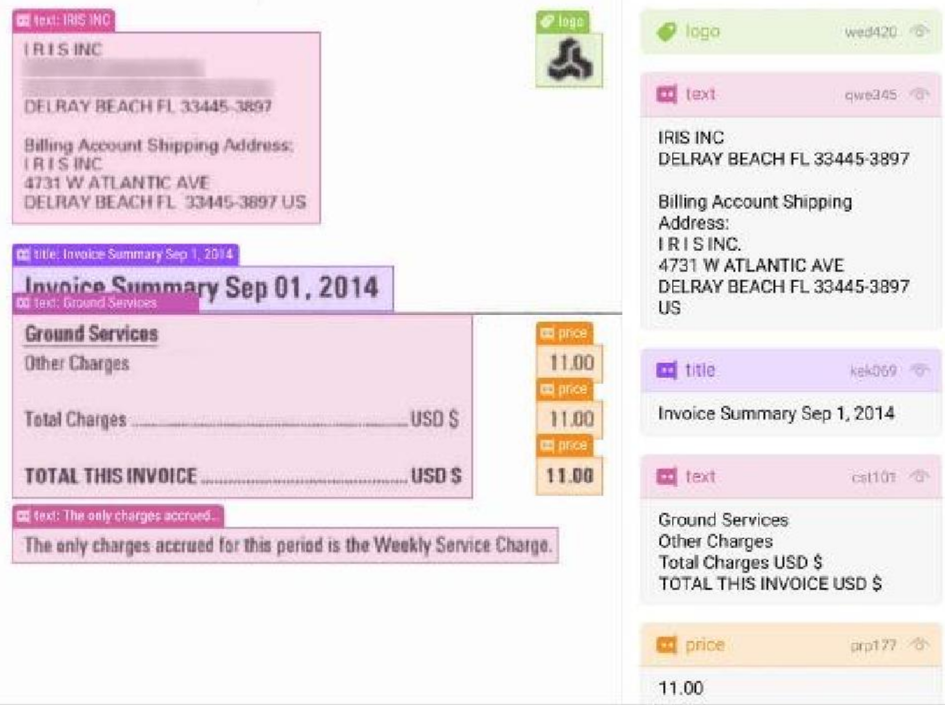
"DECOGNIZE" AIMS TO SIMPLIFY HEALTHCARE DATA MANAGEMENT BY USING ADVANCED TECHNOLOGIES TO STREAMLINE THE DIGITIZATION OF PRESCRIPTIONS, ENSURING ENHANCED EFFICIENCY AND IMPROVED PATIENT SAFETY.

GROUP MEMBERS:	SUPERVISOR	TOOLS
MUHAMMAD SHARJEEL AKHTAR (20P 0101) MAHAD ASHRAF (20P 0563)	SIR.SOHAIB MUHAMMAD KHAN	neo4j docker TensorFlow K OpenCV

15. UI Design



Welcome to
Decognize
Your Prescription Manager


text: IRIS INC
logo
title: Invoice Summary Sep 1, 2014
text: Ground Services
price
price
price
text: The only charges accrued...

I R I S INC
[redacted]
DELRAY BEACH FL 33445-3897

Billing Account Shipping Address:
I R I S INC
4731 W ATLANTIC AVE
DELRAY BEACH FL 33445-3897 US

Invoice Summary Sep 01, 2014

Ground Services
Other Charges 11.00

Total Charges .... USD $ 11.00

TOTAL THIS INVOICE .... USD $ 11.00

The only charges accrued for this period is the Weekly Service Charge.

IRIS INC
DELRAY BEACH FL 33445-3897

Billing Account Shipping Address:
I R I S INC.
4731 W ATLANTIC AVE
DELRAY BEACH FL 33445-3897 US

Invoice Summary Sep 1, 2014

Ground Services
Other Charges
Total Charges USD $
TOTAL THIS INVOICE USD $

11.00

16. Comparison with Tesseract

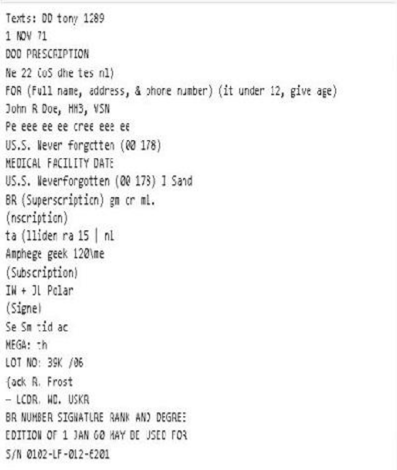
Implementation Code

```
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)
```

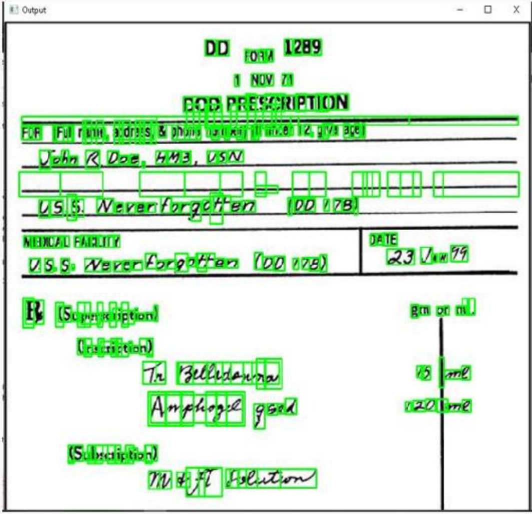
```
# Use pytesseract to get bounding boxes
boxes = pytesseract.image_to_boxes(img_rgb,
config=custom_config)

# 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: Sample Code

Sample Output



```
Texts: DD tony 1289
1 NOV 71
DOD PRESCRIPTION
Ne 22 CoS dhe tes nl)
FOR (Full name, address, & ɔhore nυmber) (it under 12, give age)
John R Doe, HH3, VSN
Pe eee ee ee cree eee ee
US.S. Never forgctten (00 178)
MEDICAL FACILITY DATE
US.S. Neverforgotten (00 173) I Sand
BR (Superscripticn) gm cr ml.
(nscripticn)
ta (1liden ra 15 | nl
Amphege geek 120\me
(Subscription)
IW + Jl Polar
(Signe)
Se Sm :id ac
MEGA: :h
LOT NO: 39K /06
{ack R. Frost
— LCDR. WD. USKR
BR NUMBER SIGNATLRE RANK AN) DEGRE:
EDITION OF 1 JAN 60 HAY DE JSED FOR
S/N 0102-LF-012-E201
```



DD FORM 1289
1 NOV 71
DOD PRESCRIPTION

FOR (Full name, address, & phone number) (If under 12, give age)
John R Doe, HM3, USN
U.S.S. Neverforgotten (DD 178)

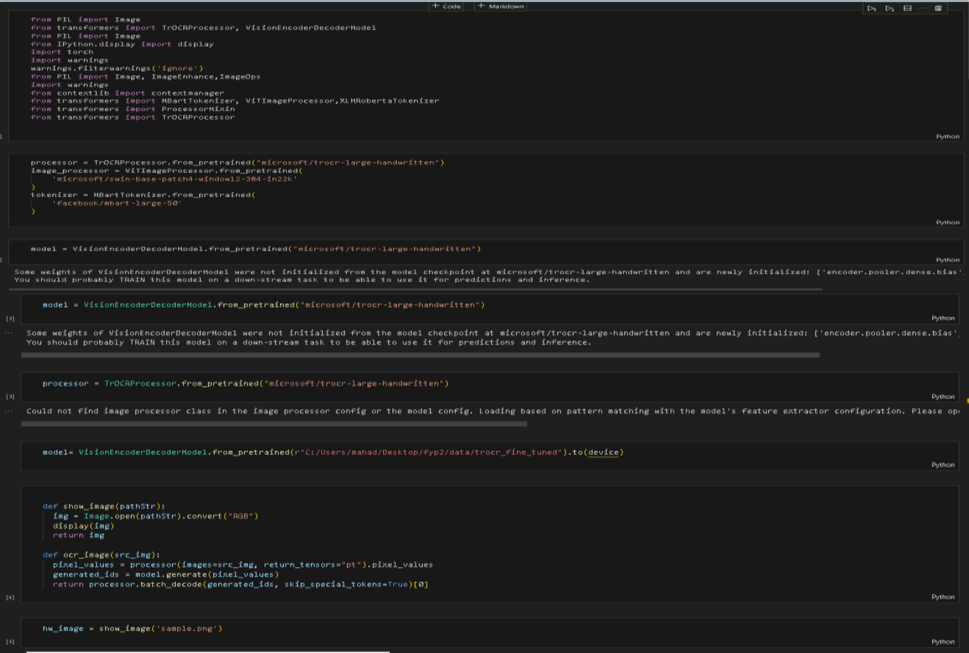
MEDICAL FACILITY
U.S.S. Neverforgotten (DD 178)

DATE
23 Jan 79

℞ (Superscription) gm or ml
(Inscription)
Tr Belladonna 15 ml
Amphogel qsad 120 ml
(Subscription)
M ft Solution

Figure 4: Sample Output

Implementation Code



```
from PIL import Image
from transformers import TrOCRProcessor, VisionEncoderDecoderModel
from PIL import Image
from IPython.display import display
import torch
import warnings
warnings.filterwarnings('ignore')
from PIL import Image, ImageEnhance,ImageOps
import warnings
from contextlib import contextmanager
from transformers import MBartTokenizer, ViTImageProcessor,XLMRobertaTokenizer
from transformers import ProcessorMixin
from transformers import TrOCRProcessor
```

```
processor = TrOCRProcessor.from_pretrained("microsoft/trocr-large-handwritten")
image_processor = ViTImageProcessor.from_pretrained(
    'microsoft/swin-base-patch4-window12-384-in22k'
)
tokenizer = MBartTokenizer.from_pretrained(
    'facebook/mbart-large-50'
)
```

```
model = VisionEncoderDecoderModel.from_pretrained("microsoft/trocr-large-handwritten")
```

```
Some weights of VisionEncoderDecoderModel were not initialized from the model checkpoint at microsoft/trocr-large-handwritten and are newly initialized: ['encoder.pooler.dense.bias',
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
[2]
model = VisionEncoderDecoderModel.from_pretrained("microsoft/trocr-large-handwritten")
```

```
Some weights of VisionEncoderDecoderModel were not initialized from the model checkpoint at microsoft/trocr-large-handwritten and are newly initialized: ['encoder.pooler.dense.bias',
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
[3]
processor = TrOCRProcessor.from_pretrained("microsoft/trocr-large-handwritten")
```

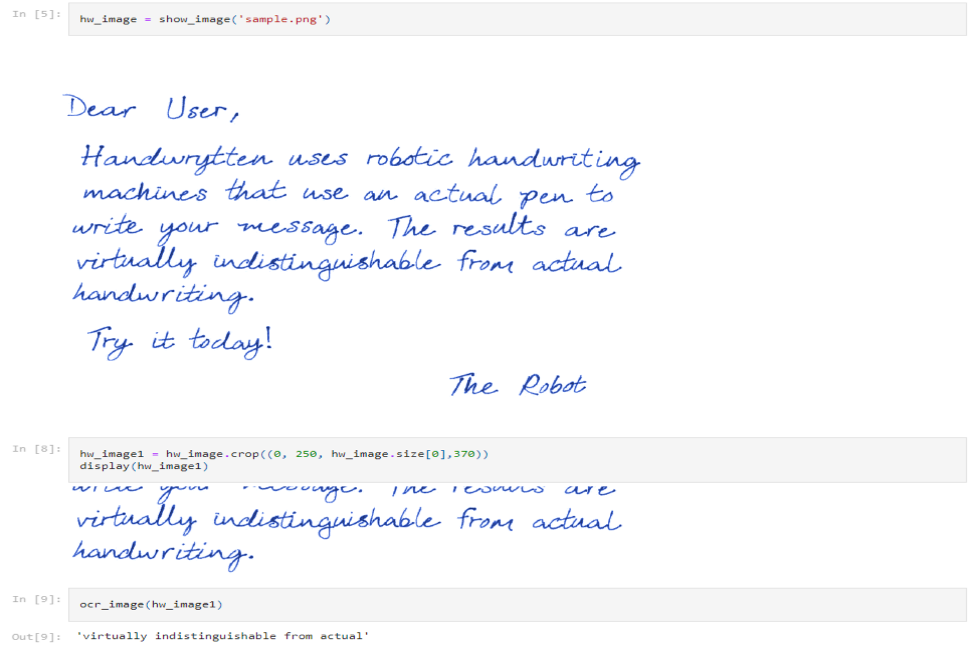
```
Could not find image processor class in the image processor config or the model config. Loading based on pattern matching with the model's feature extractor configuration. Please op
```

```
model= VisionEncoderDecoderModel.from_pretrained(r"C:/Users/mahad/Desktop/fyp2/data/trocr_fine_tuned").to(device)
```

```
[4]
def show_image(pathStr):
  img = Image.open(pathStr).convert("RGB")
  display(img)
  return img

def ocr_image(src_img):
  pixel_values = processor(images=src_img, return_tensors="pt").pixel_values
  generated_ids = model.generate(pixel_values)
  return processor.batch_decode(generated_ids, skip_special_tokens=True)[0]
```

```
[5]
hw_image = show_image('sample.png')
```

Sample Output



```
In [5]: hw_image = show_image('sample.png')
```

Dear User,

Handwrytten uses robotic handwriting machines that use an actual pen to write your message. The results are virtually indistinguishable from actual handwriting.

Try it today!

The Robot

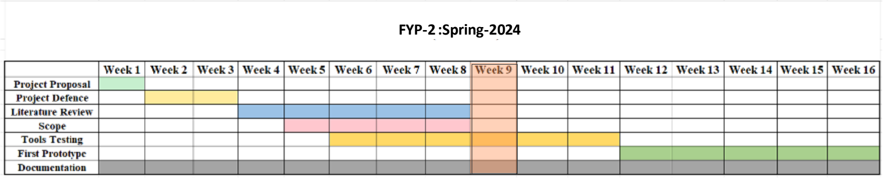
```
In [8]: hw_image1 = hw_image.crop((0, 250, hw_image.size[0],370))
        display(hw_image1)
```

virtually indistinguishable from actual
handwriting.

```
In [9]: ocr_image(hw_image1)

Out[9]: 'virtually indistinguishable from actual'
```

17. Gantt Chart

Gantt Chart



FYP-2 :Spring-2024

	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

18. Literature Review

Literature Review

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 language .
[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 digit 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 zero relation 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 inaccuracie in the zero shot relation extractor and the need for expert validation with external database enrichment relian on data 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 successful 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 in 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

References [1]

1. Filip Zelic and Anuj Sable. A review on on OCR with Tesseract OpenCV and Python. Nanonets, 2023.

2. Kamlesh Solanki . A review on optical character recognition using tensor flow. Medium, 10:39154–39176, 2021.

3. 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 [3]

4. Harald Scheidl. Article on Build handwritten text recognition using tensorflow. Medium, 9:87643–87662, 2018.

5. 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.

6. 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 [2]

7.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

8.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.

9.Wijewardena, W. R. A. D. "Medical Prescription Identification Solution." PhD diss., 2021