REPORT

ON

Custom Object Character Recognition (OCR) on AWS

Using Appwrite Integration

Building a Custom OCR System by Combining YOLOv3, Tesseract, and Appwrite

Prepared by:

Minal Devikar

Institution Name

DIGICHROME ACADEMY

[Table of Contents]

---

1. Abstract

This report details the development, training, and deployment of a custom object character recognition (OCR) system that integrates a YOLOv3-based text detection model with the Tesseract OCR engine and utilizes Appwrite for cloud storage and API management. The project aims to extract structured text from lab reports and convert it into an editable format. The system is trained on AWS and deployed via a user-friendly Streamlit application. Experimental results and performance metrics are presented, along with a discussion on system limitations and future work.

---

2. Introduction

2.1 Background

Optical Character Recognition (OCR) has become essential in converting printed and handwritten text into digital form. Advances in deep learning have led to robust OCR systems that integrate convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

2.2 Motivation

Manual transcription of lab reports is inefficient and prone to errors. Automating this process using advanced object detection (via YOLOv3) and text recognition (using Tesseract) can dramatically improve productivity and accuracy in various industrial and clinical settings.

2.3 Objectives

- Develop a custom YOLOv3 model for text detection.
- Integrate Tesseract OCR to extract text from detected regions.
- Utilize Appwrite for data storage, API management, and backend integration on AWS.
- Deploy a Streamlit application for real-time inference and user interaction.

---

3. Literature Review

3.1 Overview of OCR

This section reviews the evolution of OCR technology—from traditional methods to modern deep learning approaches that utilize CNNs and RNNs for improved accuracy and robustness.

3.2 YOLOv3 for Object Detection

YOLOv3 (You Only Look Once, Version 3) is a fast, real-time object detection system that processes images in a single forward pass. Its multi-scale detection capability makes it well-suited for identifying text regions in documents.

3.3 Tesseract OCR Engine

Tesseract is a popular open-source OCR engine developed by Google. Despite some limitations with handwriting, it remains effective for printed text extraction when combined with proper image preprocessing techniques.

3.4 Appwrite as a Cloud Platform

Appwrite offers a set of backend services (storage, APIs, authentication) that simplify cloud-based application development. This project leverages Appwrite for storing data and managing API requests, facilitating a scalable OCR solution on AWS.

---

4. System Architecture

4.1 Overall System Diagram

Insert a diagram here showing the flow from input (lab report image) to YOLOv3 detection, Tesseract OCR, and data storage/processing via Appwrite.

4.2 Detailed Architecture

- YOLOv3 Module: Detects text regions in lab reports.
- OCR Module: Uses Tesseract to extract text from cropped image regions.

- **Appwrite Module: Manages storage of datasets and model outputs; provides APIs for real-time data retrieval.**

4.3 Appwrite Integration

The Appwrite component is integrated with AWS services (SageMaker for training, S3 for storage) and the Streamlit interface, enabling automatic updates and monitoring through a user-friendly dashboard.

---

5. Methodology

5.1 Data Preparation and Annotation

- **Dataset Collection: Lab reports are gathered and annotated manually.**
- **Preprocessing: Images are resized, normalized, and converted to the YOLO format.**
- **Storage: The dataset is stored in AWS S3 for easy access during training.**

5.2 YOLOv3 Model Training

- **Model Architecture: The YOLOv3 model is implemented using a Darknet-based framework.**
- **Weights Conversion: Original Darknet weights (.weights) are converted to a PyTorch checkpoint (.pt) for compatibility.**
- **Training Process: The model is trained on AWS SageMaker with data augmentation and multi-scale training techniques.**

5.3 Tesseract OCR Integration

- **Region Cropping: The YOLOv3 model outputs bounding boxes that are used to crop regions of interest.**
- **OCR Processing: Cropped regions are processed by Tesseract OCR, and configuration parameters are fine-tuned for optimal accuracy.**
- **Post-processing: Extracted text is cleaned and formatted for further analysis.**

5.4 Appwrite Integration

- **Data Storage: Appwrite stores training data, model outputs, and OCR results.**
- **API Management: Appwrite provides RESTful API endpoints for accessing and managing processed data.**
- **Dashboard: An Appwrite dashboard offers insights into data usage and system performance.**

---

6. Implementation

6.1 Environment Setup and Tools

- **Software: Python 3.x, PyTorch, TensorFlow, Tesseract, Streamlit.**
- **Hardware: CPU/GPU-based systems; initial experiments conducted on CPU.**
- **Cloud Services: AWS SageMaker, S3, and Appwrite are configured with proper IAM roles.**

6.2 YOLOv3 Model Development and Training

- **Model Code: A custom YOLOv3 model (based on Darknet) is implemented in Python.**
- **Training Script: A train.py script is used to train the model with dummy data as a placeholder for actual training.**
- **Weights Handling: The project converts and loads pre-trained weights (if available) from a .pt file.**

6.3 Streamlit Application for Inference and Results Display

- **User Interface: A Streamlit app (app.py) allows users to upload images or select a default image.**
- **Detection Pipeline: The app runs the YOLOv3 model on the input image, draws bounding boxes, and extracts text using Tesseract.**
- **Results Display: The recognized text is displayed in a table, and detection results are shown visually.**

6.4 Appwrite Integration and Screenshots

- **Dashboard Integration: The report includes screenshots of the Appwrite dashboard in column format.**
    - **Left Column: Appwrite file storage and API settings.**
    - **Right Column: Results from the OCR pipeline and inference dashboard.**
- **Workflow Screenshots: Diagrams and figures illustrate the end-to-end pipeline from detection to OCR and data management.**

7. Experimental Results

7.1 Training Performance Metrics

- **Loss Curves: Graphs showing training loss over epochs.**
- **Convergence: Analysis of convergence behavior and stability.**

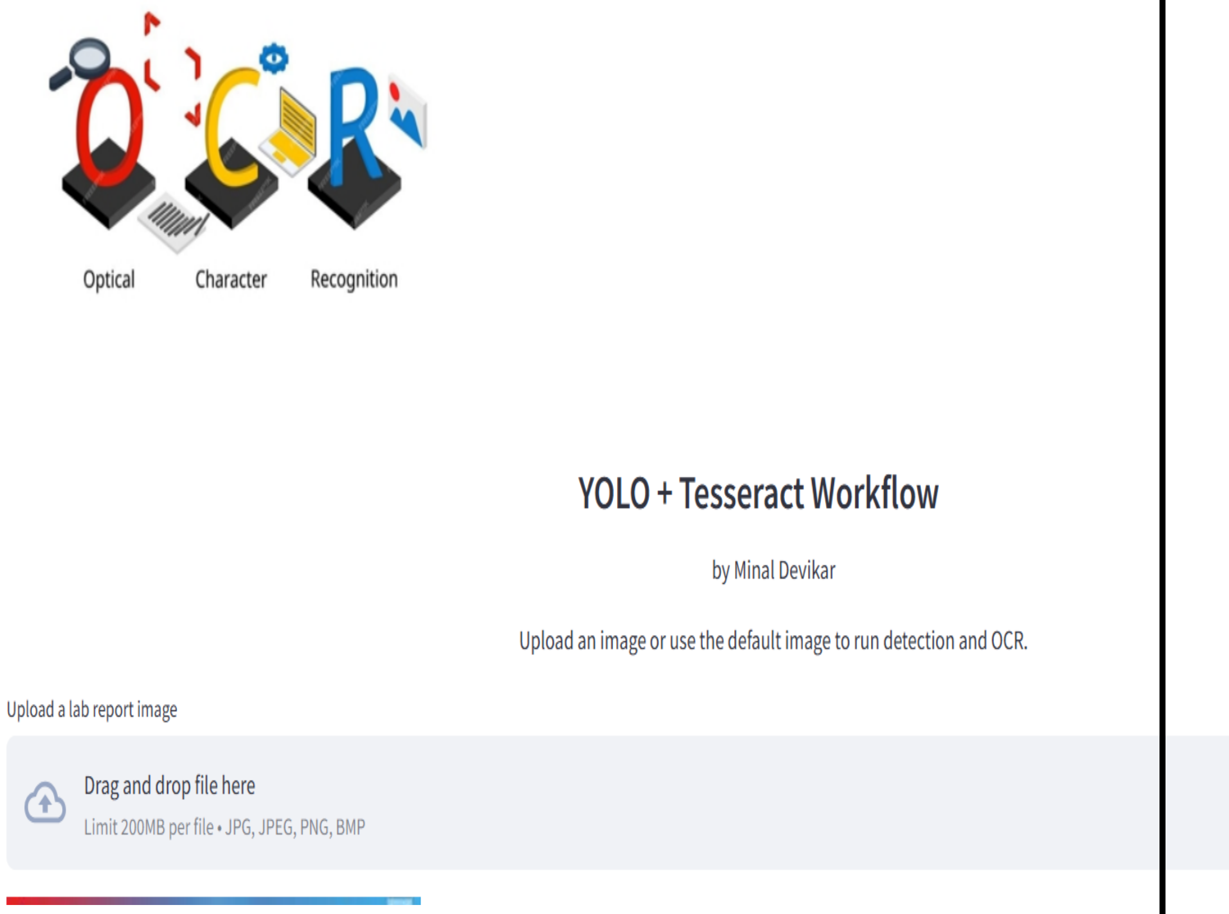
7.2 OCR Accuracy Evaluation

- **Output Comparison: Tables comparing OCR output with ground truth labels.**
- **Metrics: Precision, recall, and F1-score for text extraction accuracy.**

7.3 Appwrite Dashboard and API Results

Insert two-column screenshot here:

- **Column 1: Appwrite dashboard showing file storage and API usage.**



YOLO + Tesseract Workflow

by Minal Devikar

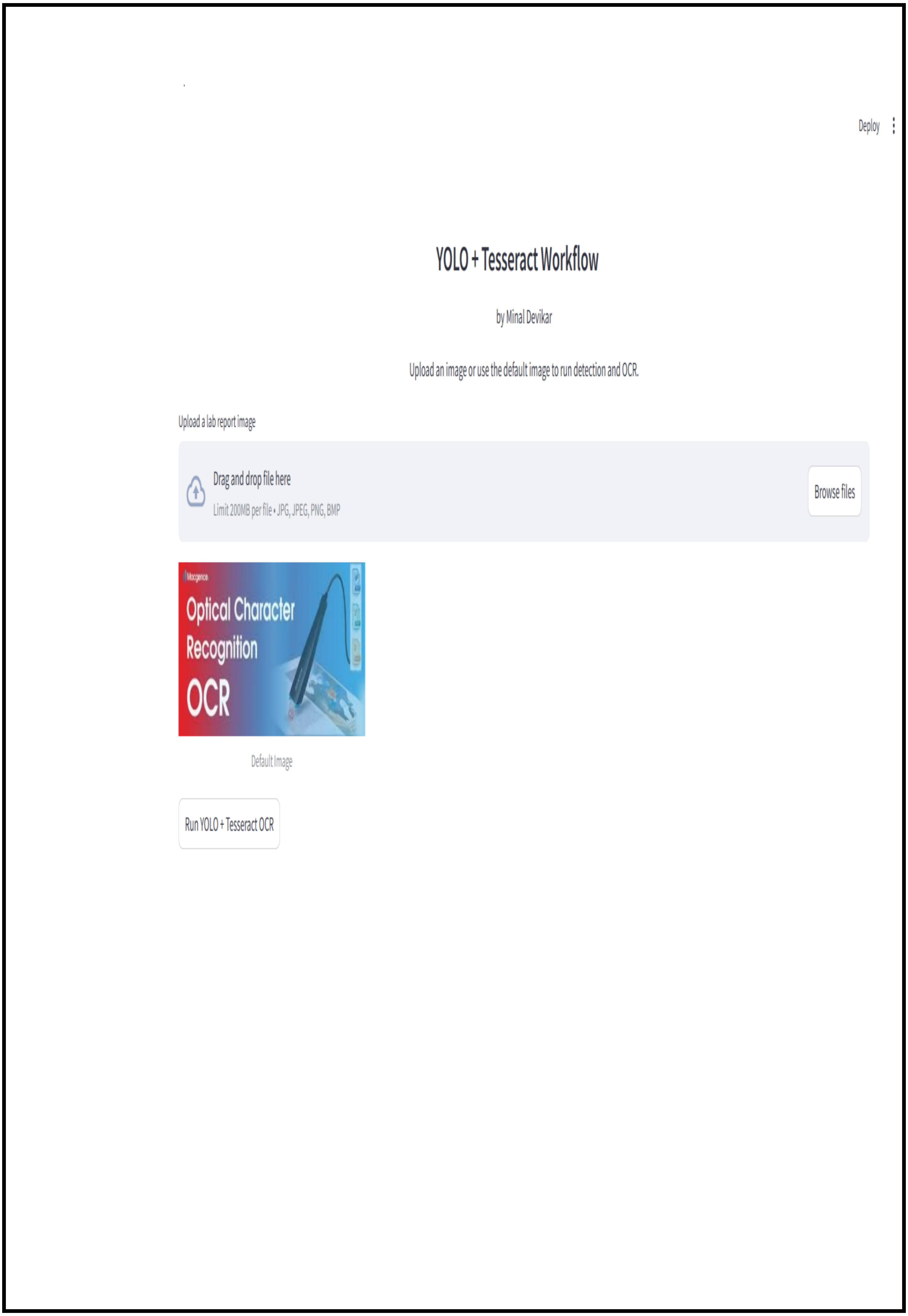
Upload an image or use the default image to run detection and OCR.

Upload a lab report image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG, BMP

- **Column 2: Inference results from the OCR pipeline.**



Deploy

YOLO + Tesseract Workflow

by Minal Devikar

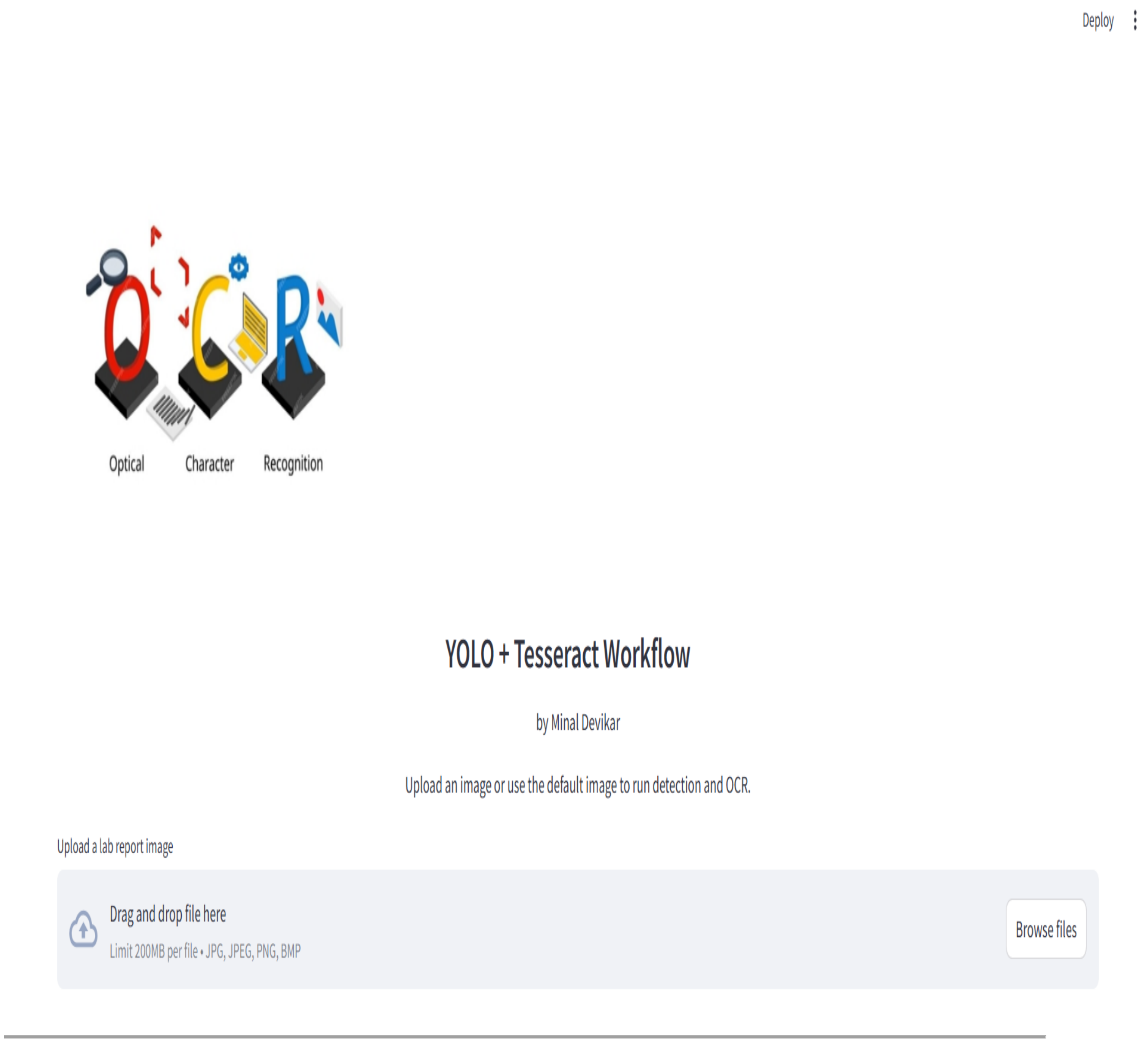
Upload an image or use the default image to run detection and OCR.

Upload a lab report image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG, BMP

Default Image



YOLO + Tesseract Workflow

by Minal Devikar

Upload an image or use the default image to run detection and OCR.

Upload a lab report image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG, BMP

Upload a lab report image

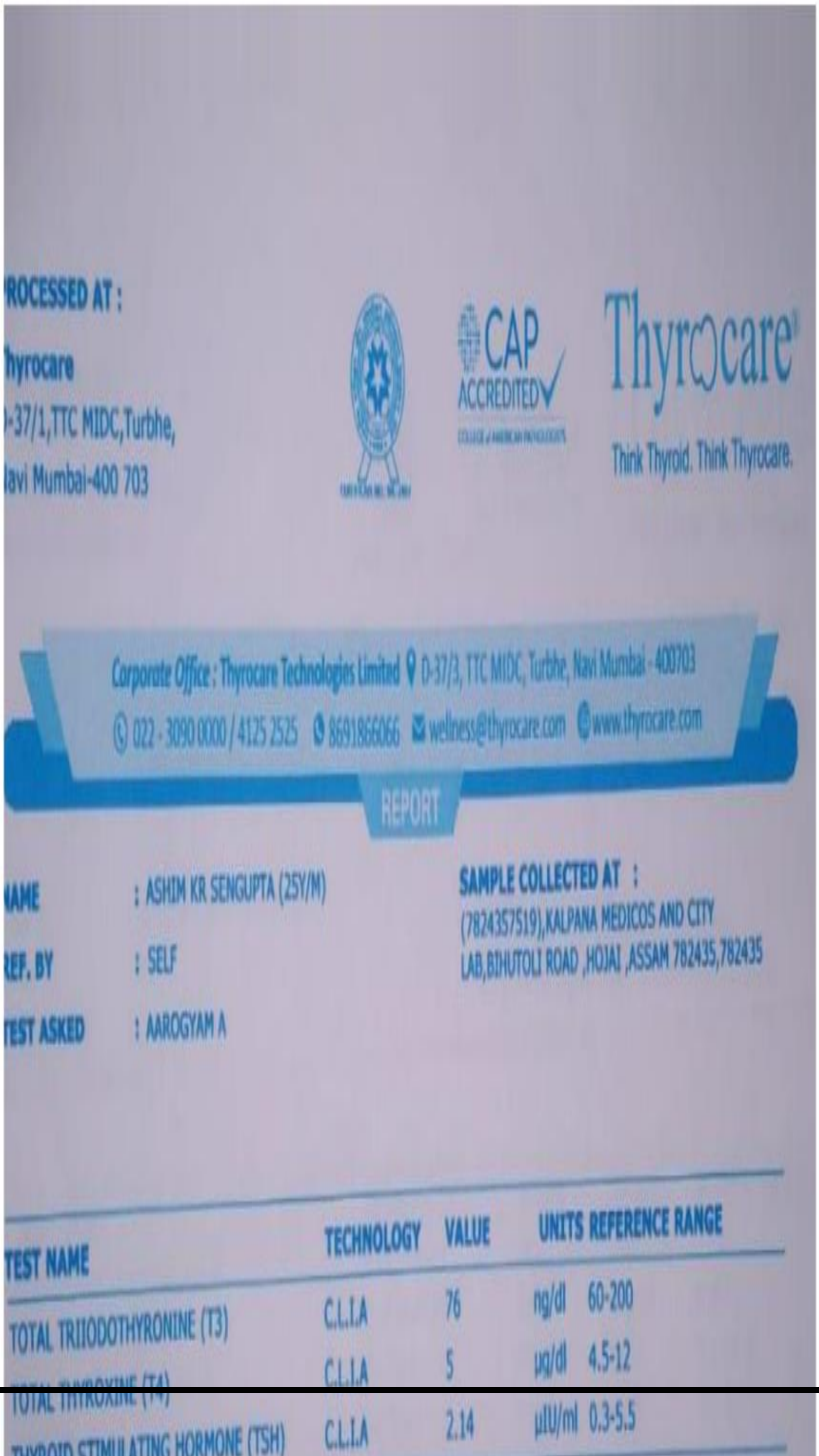
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG, BMP

Browse files

thyrocare_0_122.jpg 57.2KB

X



ROCESSED AT :

hyrocare

-37/1,TTC MIDC,Turbhe,

avi Mumbai-400 703

CAP ACCREDITED

Thyrocare

Think Thyroid. Think Thyrocare.

Corporate Office : Thyrocare Technologies Limited D-37/3, TTC MIDC, Turbhe, Navi Mumbai - 400703

022 - 3090 0000 / 4125 2525 8691866066 wellness@thyrocare.com www.thyrocare.com

REPORT

AME	: ASHIM KR SENGUPTA (25Y/M)
EF. BY	: SELF
EST ASKED	: AAROGYAM A

SAMPLE COLLECTED AT :
(7824357519),KALPANA MEDICOS AND CITY LAB,BIHUTOLI ROAD ,HOJAI ,ASSAM 782435,782435

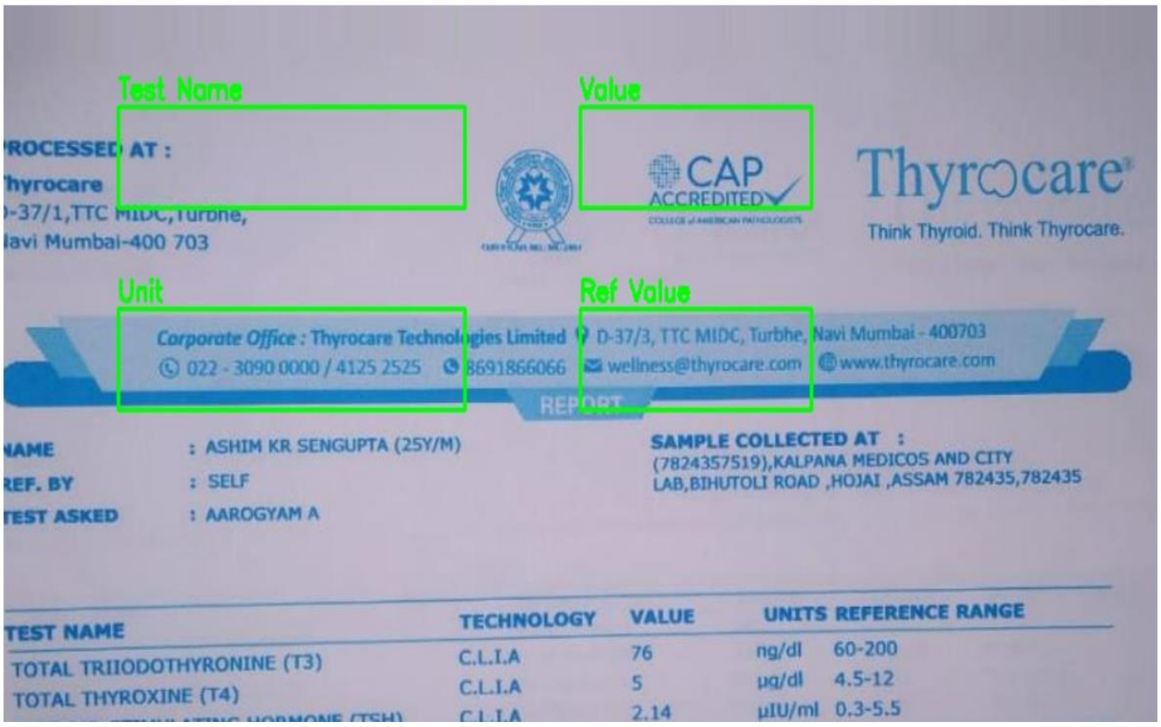
TEST NAME	TECHNOLOGY	VALUE	UNITS	REFERENCE RANGE
TOTAL TRIIODOTHYRONINE (T3)	C.L.I.A	76	ng/dl	60-200
TOTAL THYROXINE (T4)	C.L.I.A	5	µg/dl	4.5-12
THYROID STIMULATING HORMONE (TSH)	C.L.I.A	2.14	µIU/ml	0.3-5.5

Uploaded Image

Run YOLO + Tesseract OCR

Running YOLO detection...

Running Tesseract OCR on each bounding box...



ROCESSED AT :
hyrocare
-37/1,TTC MIDC,Turbhe,
lavi Mumbai-400 703

CAP ACCREDITED

Thyrocare®
Think Thyroid. Think Thyrocare.

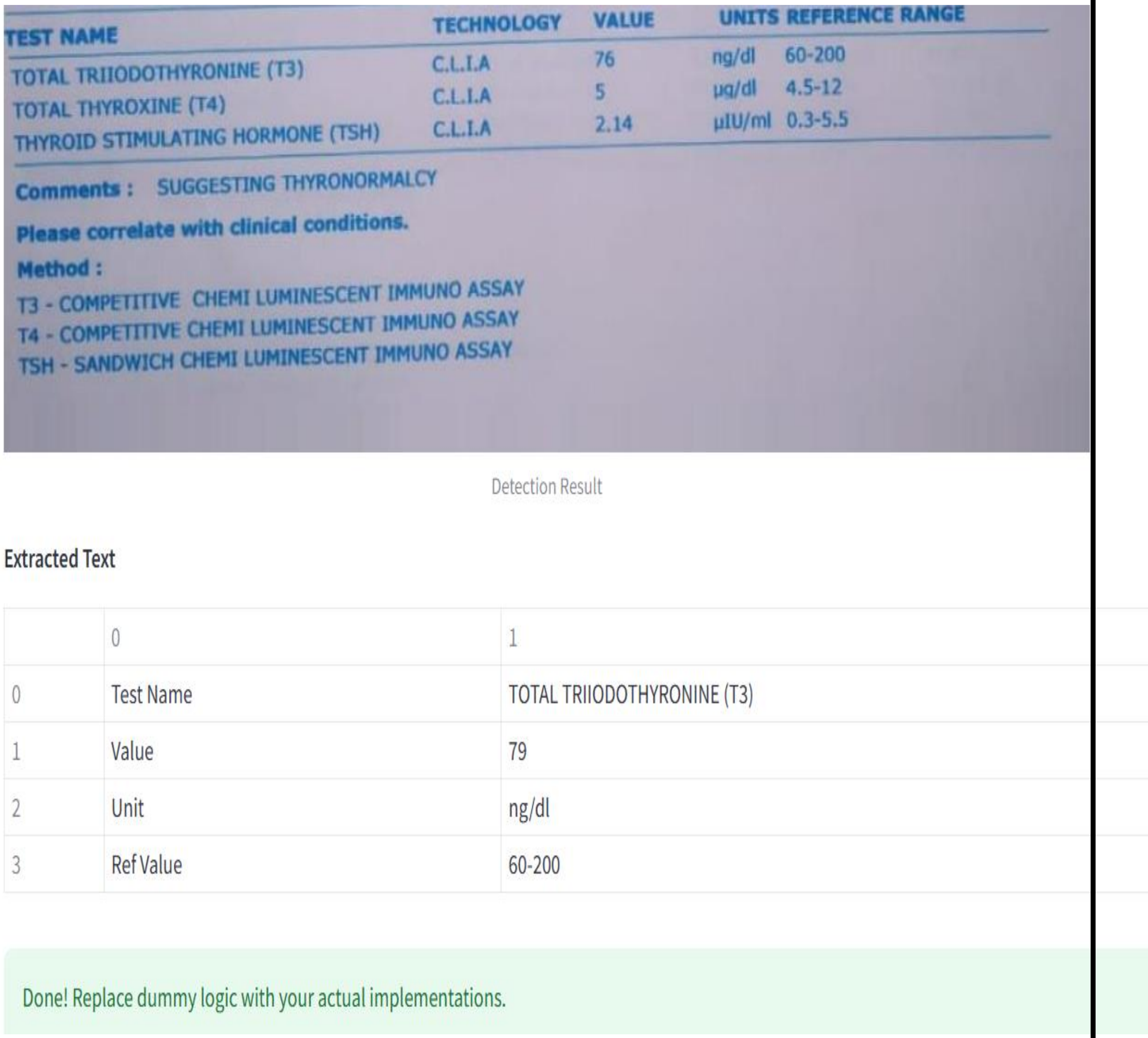
Corporate Office : Thyrocare Technologies Limited D-37/3, TTC MIDC, Turbhe, Navi Mumbai - 400703
022 - 3090 0000 / 4125 2525 8691866066 wellness@thyrocare.com www.thyrocare.com

REPORT

IAME	: ASHIM KR SENGUPTA (25Y/M)
REF. BY	: SELF
TEST ASKED	: AAROGYAM A

SAMPLE COLLECTED AT :
(7824357519),KALPANA MEDICOS AND CITY LAB,BIHUTOLI ROAD ,HOJAI ,ASSAM 782435,782435

TEST NAME	TECHNOLOGY	VALUE	UNITS	REFERENCE RANGE
TOTAL TRIIODOTHYRONINE (T3)	C.L.I.A	76	ng/dl	60-200
TOTAL THYROXINE (T4)	C.L.I.A	5	µg/dl	4.5-12
[illegible]	C.L.I.A	2.14	µIU/ml	0.3-5.5



TEST NAME	TECHNOLOGY	VALUE	UNITS	REFERENCE RANGE
TOTAL TRIIODOTHYRONINE (T3)	C.L.I.A	76	ng/dl	60-200
TOTAL THYROXINE (T4)	C.L.I.A	5	µg/dl	4.5-12
THYROID STIMULATING HORMONE (TSH)	C.L.I.A	2.14	µIU/ml	0.3-5.5

Comments : SUGGESTING THYRONORMALCY

Please correlate with clinical conditions.

Method :

T3 - COMPETITIVE CHEMI LUMINESCENT IMMUNO ASSAY

T4 - COMPETITIVE CHEMI LUMINESCENT IMMUNO ASSAY

TSH - SANDWICH CHEMI LUMINESCENT IMMUNO ASSAY

Detection Result

Extracted Text

	0	1
0	Test Name	TOTAL TRIIODOTHYRONINE (T3)
1	Value	79
2	Unit	ng/dl
3	Ref Value	60-200

Done! Replace dummy logic with your actual implementations.

8. Discussion

8.1 Analysis of Results

- **Summary of key findings from training and inference.**
- **Discussion on the performance of YOLOv3 and Tesseract OCR integration.**

8.2 Challenges and Limitations

- **Issues with weights conversion, data annotation, and training stability.**
- **Limitations in hardware, dataset size, and OCR accuracy.**

8.3 Lessons Learned

- **Importance of data quality and proper annotation.**
- **Challenges in integrating multiple technologies (YOLO, Tesseract, Appwrite).**

---

9. Future Work

- **Model Refinement: Improve the YOLOv3 architecture and train on a larger dataset.**
- **Advanced OCR: Explore deep learning-based OCR alternatives for improved accuracy.**
- **Automation: Enhance Appwrite integration with automated pipelines and API endpoints.**
- **Scalability: Test the system on a broader range of documents and optimize for GPU-based training.**

---

10. Conclusion

This project demonstrates a complete pipeline for custom OCR by combining YOLOv3 for text detection, Tesseract for text recognition, and Appwrite for cloud data management. While the current implementation uses a dummy training loop and simulated detection, the foundation is established for developing a robust, scalable

OCR system. Future improvements will focus on enhancing model accuracy, refining integration, and automating deployment processes.

---

11. References

- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.
- Smith, R. (2007). An overview of the Tesseract OCR engine.
- AWS Documentation. Retrieved from https://aws.amazon.com/documentation/
- Appwrite Documentation. Retrieved from https://appwrite.io/docs
- PyTorch Documentation. Retrieved from https://pytorch.org/docs/stable/index.html
- TensorFlow Documentation. Retrieved from https://www.tensorflow.org/

---

12. Appendices

Appendix A: Code Listings

- train.py: Script for training the YOLOv3 model.
- app.py: Streamlit app for inference and OCR display.
- models/darknet.py: YOLOv3 model implementation (dummy version provided).

Appendix B: Experimental Setup

- **Detailed configuration of the training environment (hardware, software, virtual environments, AWS resources).**