School of Computer Engineering and Technology
MIT World Peace University, Kothrud,
Pune 411 038, Maharashtra - India
2022-2023

Project Report

on

Hematology report data extraction and inference generation

Submitted by

Group B17

Project Members

PRN	Name	Roll no.
1032190599	Preetika Sastry	14
1032192059	Pranav Nirmal	52
1032192095	Ayush Shrivastav	54
1032192119	Aniket Sharma	56

Under the Internal Guidance of

Prof. Vrushali Y. Kulkarni

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

CERTIFICATE

This is to certify that,

Preetika Sastry
Pranav Nirmal
Ayush Shrivastav
Aniket Sharma

of BTech.(Computer Science & Engineering) have completed their project titled "*Haematology report data extraction and inference generation*" and have submitted this Capstone Project Report towards fulfillment of the requirement for the Degree-Bachelor of Computer Science & Engineering (BTech-CSE) for the academic year 2022-2023.

Dr. Vrushali Kulkarni
Project Guide
School of CET
MIT World Peace University, Pune

Dr. Vrushali Kulkarni
Program Head
School of CET
MIT World Peace University, Pune

Internal Examiner:

External Examiner:

Date:

Acknowledgement

The successful completion of this research paper would not have been possible without the invaluable support, guidance, and encouragement from numerous individuals. We would like to express our heartfelt gratitude to our project mentor Dr. Vrushali Y. Kulkarni, project coordinator Prof. Madhura Phatak, all the field specialists and doctors that we consulted, for their contributions and mentorship throughout this research endeavor.

We extend our deepest appreciation to Dr.Vrushali Y. Kulkarni for her unwavering commitment, expertise, and patience. Her guidance and insightful feedback have been instrumental in shaping the direction and quality of this research. We are grateful for the precious time she dedicated to providing us with direction and support.

We would also like to express our sincere gratitude to the project coordinators, Prof. Madhura Phatak and Prof. Shamala Mantri for their exceptional organizational skills and meticulous attention to detail. Their tireless efforts in managing and coordinating various aspects of the research project have been crucial in ensuring its smooth execution.

Furthermore, we are grateful to the field specialists who generously shared their knowledge and expertise. Their valuable insights and practical advice have greatly enriched the research, enabling us to gain a deeper understanding of the subject matter. Their willingness to invest time and effort in mentoring us is deeply appreciated.

Our heartfelt appreciation goes to our parents for their unwavering support and encouragement throughout this research journey. Their love, belief in our abilities, and constant motivation have been the driving force behind our achievements. We are forever grateful for their sacrifices and for instilling in us the values of hard work and perseverance.

In conclusion, we would like to express our profound gratitude to all those mentioned above for their mentorship, contributions, cooperation, and encouragement throughout this research project. Their support has played a pivotal role in its successful completion. We are honored to have had the opportunity to work with such exceptional individuals, and we are grateful for the profound impact they have had on our personal and professional development.

Preetika Sastry

Pranav Nirmal

Ayush Shrivastav

Aniket Sharma

Abstract

Haematology is a critical branch of medicine that deals with the study, diagnosis, and treatment of blood disorders. This field involves the analysis of various blood components such as red blood cells, white blood cells, and platelets, as well as their function and interaction. Haematology reports provide essential information about a patient's blood count, differential, and other factors that aid in the diagnosis and monitoring of blood-related conditions. With the help of Deep learning algorithms and Optical Character Recognition (OCR) technology, haematology reports can be efficiently processed and analysed, allowing for structured and more accurate diagnoses. Haematologists work closely with other medical specialists to provide comprehensive care to patients, making haematology a crucial component of modern medicine.

List of Figures

5.1	High Level System Block Diagram	8
5.2	System Class Diagram	9
5.3	Use Case Diagram	10
7.1	YOLOv8 Architecture……...	15
7.2	EasyOCR Framework……...	18
7.3	Tesseract OCR Architecture……..	19
7.4	GUI directory	22
7.5	Homepage	23
7.6	Result page	23
7.7	Downloaded report	24
8.1	RoboFlow model performance……..	28
9.1	YOLOv8 result……..	38
9.2	RoboFlow 2.0 Object Detection (Fast) result……...	38

List of Tables

1.1	Information ……	1
2.1	Literature Survey……	4
9.1	Object detection model comparison……	38
9.2	OCR Comparison……...	40

Contents

Abstract I
List of Figures II
List of Tables III

1	Introduction		1
	1.1	Problem definition	1
	1.2	Scope	2
2	Objective		3
3	Literature Survey		4
4	Technologies used		7
5	System Design		8
6	System requirements		11
	6.1	Software requirements	11
	6.2	Hardware requirements	11
7	Techniques used		12
	7.1	Libraries used	12
	7.2	Dataset Description and Data Pre-processing	12
	7.3	Module 1: Object Detection	14
	7.4	Module 2: Image Processing	17
	7.5	Module 3: OCR Text Extraction	18
	7.6	Module 4: Report Generation	20
	7.7	Module 5: GUI	22
8	Coding		25
	8.1	Data augmentation	25
	8.2	Annotation	27
	8.3	Model Training	27
	8.4	Image processing and text/ value extraction	30
	8.5	Report generation	32
	8.6	GUI	35
9	Testing		38

10	Conclusion	43
11	Future prospects of the project	44
12	References	45
	A. Base Paper(s)	47
	B. Plagiarism Report from any open source	

Chapter 1
Introduction

In the field of medical diagnostics, accurate and efficient extraction of key data from hematology reports is of paramount importance. These reports contain vital information such as hemoglobin levels, white blood cell (WBC) count, platelet count, red blood cell (RBC) count, neutrophil and lymphophil percentages, as well as patient-specific details like age, name, and sex. Extracting this data accurately and quickly is crucial for timely diagnosis and effective patient care.

Traditionally, the process of data extraction from hematology reports has relied on manual entry, which is not only time-consuming but also prone to human errors. Furthermore, these reports are often in unstructured formats, making it challenging to efficiently extract and utilize the data contained within. Therefore, there is a pressing need for automated methods that can extract and structure the information from hematology reports accurately and reliably

1.1 Problem definition

The traditional manual entry of data from hematology reports is time-consuming and error-prone. Additionally, the unstructured nature of these reports poses a challenge in extracting and utilizing the data effectively. This necessitates the development of automated methods that can accurately and reliably extract key parameters from hematology reports, such as hemoglobin levels, WBC count, platelet count, RBC count, neutrophil and lymphophil percentages, as well as patient-specific details like age, name, and sex.

To address this problem, we propose a novel approach that combines object detection and optical character recognition (OCR) techniques. By utilizing object detection, specific regions of interest in hematology report images, such as tables or sections containing key data elements, can be identified and isolated. Subsequently, OCR techniques are applied to convert the textual content in the identified regions into machine-readable text, enabling the extraction of specific data points.

The extracted data can then be utilized to generate a structured diagnosis report, providing healthcare professionals with an easily interpretable summary of the patient's condition. This automation not only saves time but also reduces the potential for errors that may occur during

manual data entry, leading to more accurate and consistent diagnoses.

In summary, the problem at hand is the manual and error-prone process of data extraction from unstructured hematology reports. The proposed approach aims to overcome this challenge by combining object detection and OCR techniques, facilitating the automated extraction and structuring of vital parameters. This research seeks to contribute to the advancement of medical document processing and improve the efficiency and accuracy of diagnostic procedures, ultimately enhancing patient care.

1.2 Scope

This research project aims to develop an automated system for extracting key data from hematology reports and generating structured diagnosis reports. The focus will be on utilizing object detection and optical character recognition (OCR) techniques to extract parameters such as hemoglobin levels, WBC count, platelet count, RBC count, neutrophil and lymphophil percentages, as well as patient-specific details (age, name, and sex).

The project does not cover the analysis or interpretation of the extracted data for medical diagnosis. It will focus on the extraction process using object detection and OCR. The limitations of the proposed approach will be identified and discussed.

Overall, this project aims to automate data extraction and improve the efficiency of generating structured diagnosis reports in hematology. The system developed will streamline the diagnostic process, leading to enhanced patient care.

Chapter 2
Objective

The objective of this research project is to develop an automated system for efficient extraction of key data from hematology reports and generation of structured diagnosis reports. The project aims to achieve the following objectives:

The project will involve:

- Data Collection and Preparation: Gathering a diverse dataset of hematology reports, augmenting and annotating it with ground truth labels.
- Object Detection Model Development: Training a deep learning-based model to identify and localize relevant regions in hematology report images, such as data sections like name, age, sex, etc.
- OCR Integration and Data Extraction: Implementing OCR techniques to convert textual content within the identified regions into machine-readable text and extracting specific data points.
- Diagnosis Report Generation: Utilizing the extracted data to generate structured diagnosis reports summarizing the patient's condition.
- Evaluation and Performance Metrics: Assessing the system's performance using evaluation metrics like accuracy, precision, recall, and F1 scores.

By achieving these objectives, this research project aims to contribute to the advancement of automated data extraction and diagnosis report generation in the field of hematology. The ultimate goal is to improve the efficiency, accuracy, and quality of patient care by streamlining the diagnostic process and providing healthcare professionals with structured and reliable information extracted from hematology reports.

Chapter 3
Literature Survey

Title	Authors	Year	Aim/Objective	Methodology	Architecture
Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach	WEN YUAN XUE 1 , QING YON G LI 1 , (Mem ber, IEEE), QIYU AN XUE2	2019	Using deep learning, medical laboratory report photos can now be read for text.	There are various phases in the technique employed in the study. The supplied image is first preprocessed to improve contrast and lower noise. Followed by object detection model, Then, Following that, the text portions that were recognised are cropped, scaled, and input into the OCR model for text recognition.	Text detection: Faster R-CNN model is employed to find the areas in the picture that contain text. Text recognition: CRNN with the final layer connected to BLSTM.
Text Mining the History of Medicine	Paul Thom pson, John McNa ught and Sophia Anani adou	2020	By minimising mistakes in OCR-generated text, this OCR correction technique is tailored for historical medical records and increases the effectiveness of search engines. The approach combines a medically-tuned spell checking strategy that is directed by subject-specific language usage with rule-based correction of common mistakes.	To fix OCR problems, use rule-based and spell-checking techniques. They compare the effectiveness of four spellcheckers and choose to base their error-correction method on Hunspell.	Hunspell, rule-based spell checker, is a default dictionary used for OCR correction, and it provides spelling suggestions that are ranked by frequency. It is also mentioned that a medically-enhanced version of Hunspell is used, which includes specialized medical terminology. The proposed method of incorporating corpus-based information and augmenting Hunspell's dictionary with specialized vocabulary helps to refine its suggestions and avoid over-correction of valid words.

Deep Learning for Automated Extraction of Primary Sites from Cancer Pathology Reports	John X. Qiu, Hong-Jun Yoon, Paul A. Fearn, and Georgia D. Tourassi	2016	Highlights how OCR systems based on AI can automate medical transcription and extract information from handwritten clinical notes, leading to a better understanding of entire processes. The project aims to develop an OCR application using deep learning for offline character recognition in the medical sector, addressing challenges posed by large and dispersed datasets.	In the article, word embedding techniques and vector space models (VSMs) are discussed in relation to natural language processing (NLP). In contrast to word embedding methods, which are learnt representations of words taught using deep learning techniques, VSMs are used to encode documents inside a corpus with a feature vector based on word counts. The usage of the word2vec skip-gram approach for training word vectors is also covered in the paper. The learned word vectors may be used to create a document matrix and are used to represent a word's location within a corpus.	Making an OCR model using CNN and RNN. According to the results, random embeddings perform better for the information extraction problem with well-populated classes. Pre-trained embeddings performed better, nevertheless, when the classification challenge comprised classes with low prevalence.
Diagnosis of Diseases from Medical Check-up Test Reports Using OCR Technology with BoW and AdaBoost algorithms	Wisam A. Qader, Musa M. Ameen	2020	a method to diagnose illness using test results from health checkups. The suggested method uses optical character recognition (OCR) technology to transform hard copy test reports into text data that may be edited.	OCR Library Used: Tesseract Feature Selection: BoW Classification Algorithm: Naive Bayes Improved Accuracy via AdaBoost Algorithm	OCR Technology for Disease Diagnosis from Medical Check-up Test Reports Using BoW and AdaBoost Algorithms

AUTOMATED DATA EXTRACTION FROM LAB REPORTS	Vivek Vijayan	2021	The goal is to automate data extraction from laboratory results to increase accuracy and efficiency while lowering manual labour and mistakes.	The process entails extracting the text from the lab results using optical character recognition (OCR). The relevant data is then identified and retrieved from the extracted text using natural language processing (NLP) techniques. Finally, a structured format is used to store the extracted data.	Preprocessing modules for OpenCV Tesseract: a free and open-source OCR engine developed by Google labelImg: a tool for labelling images Faster R-CNN: to assess the object detection models using Mean Average Precision (mAP)

Chapter 4
Technologies Used

LabelImg: LabelImg is a popular open-source graphical image annotation tool used for creating bounding box annotations on images. It allows researchers to manually label and annotate regions of interest, such as tables and data sections, in hematology report images. LabelImg provides an intuitive user interface that simplifies the annotation process and generates annotations in widely used formats such as XML or CSV.

Computer Vision (OpenCV): A popular open-source computer vision library called OpenCV (Open Source Computer Vision Library) offers a wide range of algorithms and methods for processing images and videos. OpenCV was used in this research study for a variety of tasks, including picture preparation, image augmentation, and image modification. It provides a variety of potent tools for improving image quality, lowering noise, and carrying out actions like image thresholding, cropping, and resizing.

Roboflow AutoML Model and YOLOv8 for Object Detection: An automated machine learning framework called Roboflow AutoML makes it easier to develop and use object identification models. Without considerable machine learning knowledge, it enables researchers to train unique models on their annotated dataset. In this project, an object detection model was trained using Roboflow AutoML using the annotated dataset. A real-time object identification system known as YOLOv8 (You Only Look Once) architecture is renowned for its speed and accuracy in recognising several items inside an image. For the best level of accuracy, the YOLOv8x model was trained using our unique annotated dataset.

Optical Character Recognition (OCR) - Google created Tesseract, an open-source OCR engine that has been widely used to extract text from photos. It is customizable for certain use cases and supports a variety of languages. Simple optical character recognition (OCR) libraries like EasyOCR make it easier to extract text from photos. It has a user-friendly interface and supports a variety of languages, making it adaptable for use in a variety of applications. EasyOCR will be used in this project to conduct OCR on the selected regions of interest in the haematology report pictures, turning the text into machine-readable format. EasyOCR uses deep learning models to reliably recognise and extract text from photos.

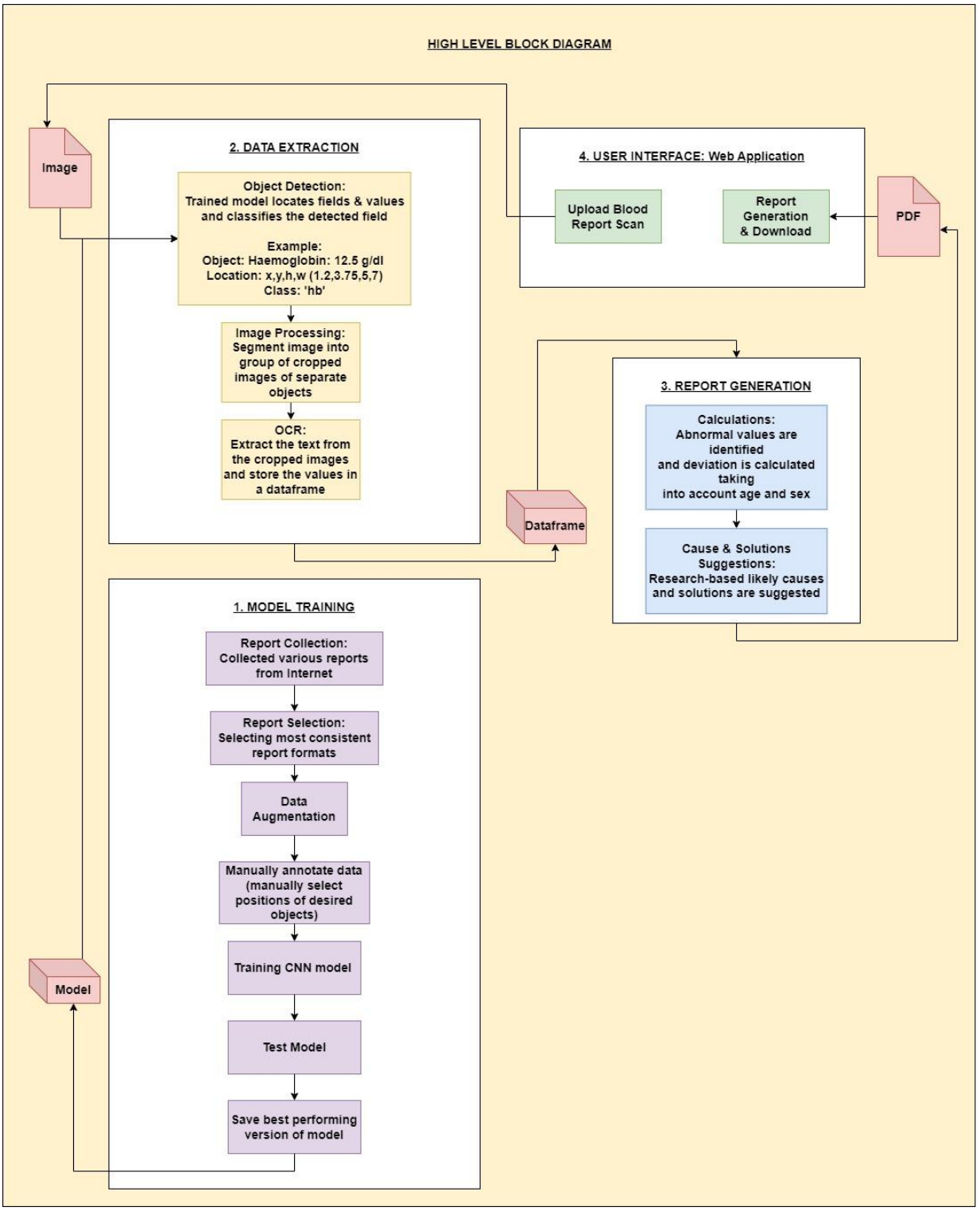
Google Colab: a cloud-based platform that allows group coding and offers free access to computer resources. Python code may be executed on Google Colab, and data processing activities can be carried out along with model training. The YOLOv8x model for this project was trained utilising the Google Colab interface and the Nvidia T4 server grade tensor GPU.

Pandas: a robust Python data analysis and manipulation module. Pandas is useful for processing and organising extracted data from haematology reports because it provides effective data structures and capabilities for managing structured data.

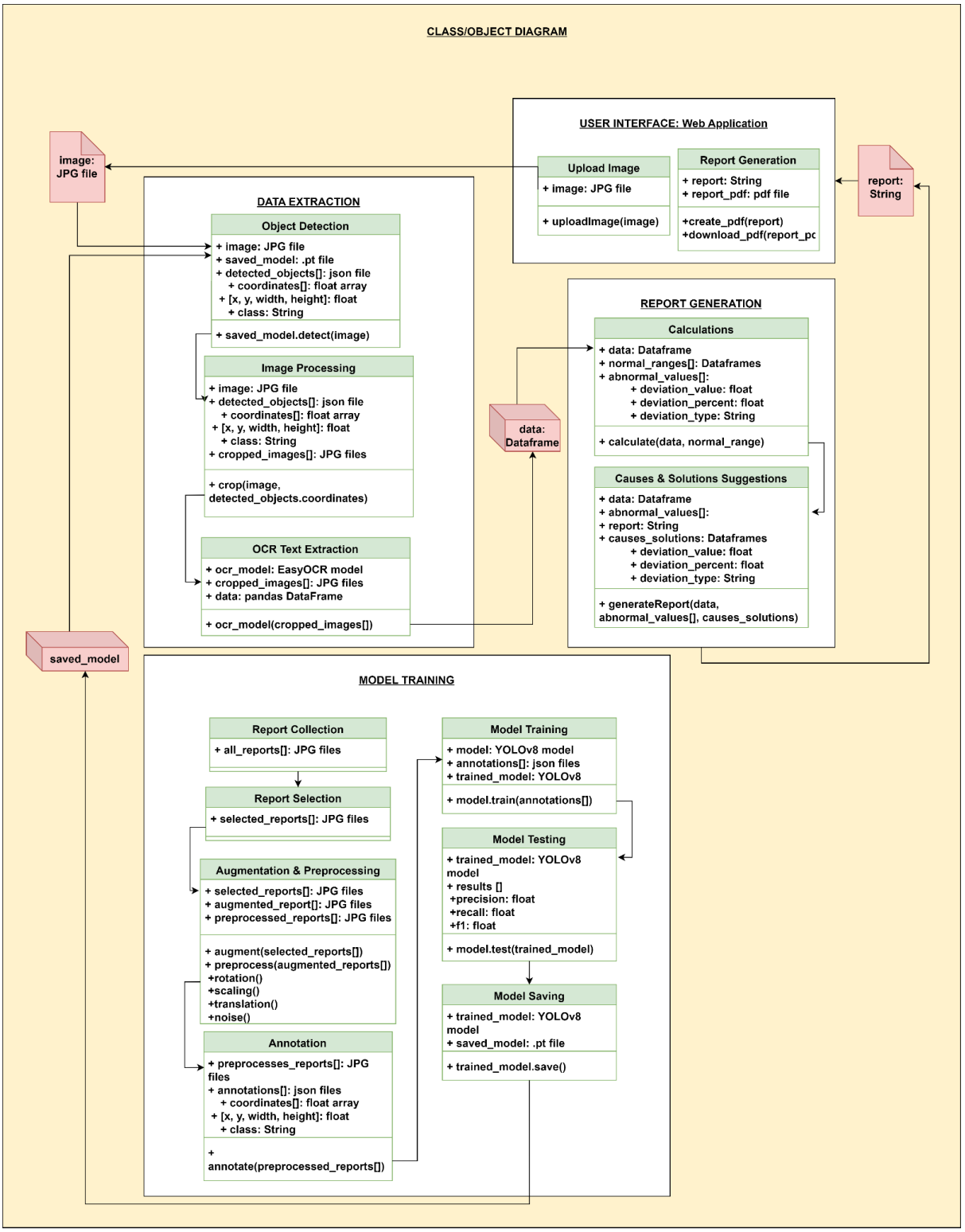
These technologies collectively enable the automation of data extraction from hematology reports through image annotation, computer vision techniques, object detection model training, optical character recognition, and data manipulation tasks.

Chapter 5
System Design

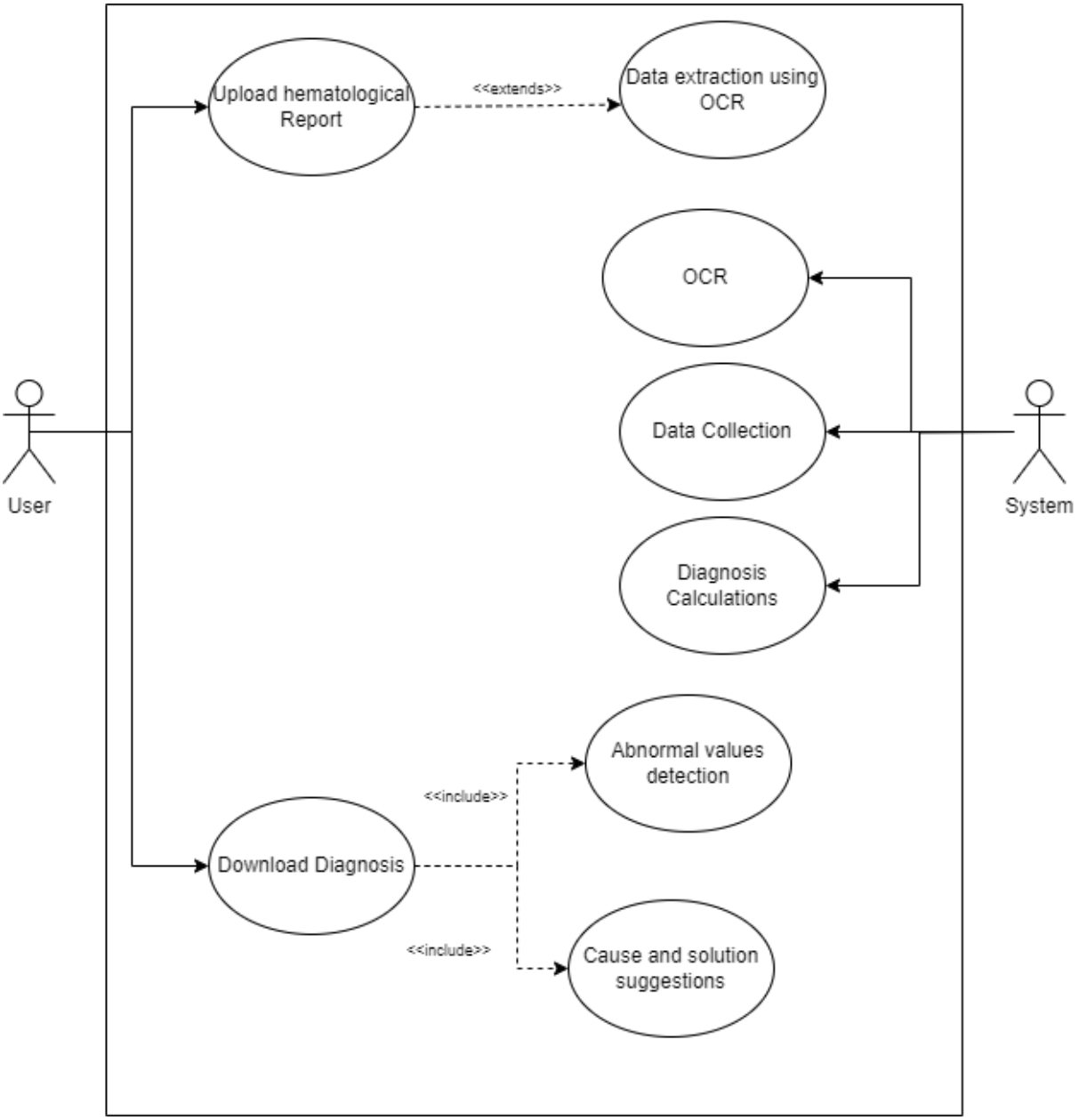
5.1 High Level System Block Diagram



5.2 System Class Diagram



5.3 Use Case Diagram



Chapter 6
System Requirements

6.1 Software Requirements

- Jupyter NB
- Python
- Anaconda
- Text editor
- Web browser (internet connection)

6.2 Hardware Requirements

- Processor: Intel core i5, 8th gen and above
- RAM: 8GB and above
- Storage: 80GB and above
- Display: CRT/ LCD/ LED
- Camera: pixel resolution of at least 5 MP
- Mouse: Compatible mouse
- Keyboard: normal or multimedia

Chapter 7
Techniques Used

7.1 Libraries Used:

- pytesseract
- PIL
- re
- os
- openCV2
- pandas
- numpy
- scipy
- random
- roboflow
- easyocr
- flask

7.2 Dataset Description and Data Pre-processing

The dataset used in this study consists of blood report images collected from the internet. The images were obtained from various laboratories and hospitals, resulting in a diverse range of formats and layouts. The reports were preprocessed using normalisation, binarization, and noise reduction techniques to improve performance. We ran the different formats of reports through our OCR models to determine which formats performed the best. Some reports were found to perform poorly in the OCR models. The future scope of this project could include such reports as well, but due to the limited project scope, we remove these before performing data augmentation.

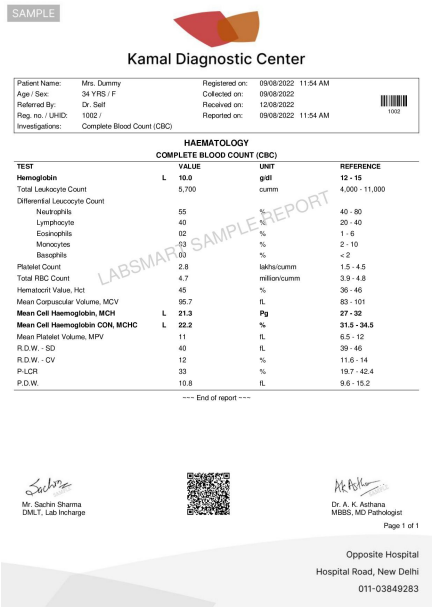
To perform data augmentation, we used OpenCV's inbuilt methods. These included the following:

- Rotation: This involves rotating an image by a given angle. In OpenCV, the getRotationMatrix2D() and warpAffine() functions can be used for rotation.
- Scaling: This involves resizing an image to a new size. In OpenCV, the resize() function can be used to resize an image.
- Translation: This involves shifting an image horizontally or vertically. In OpenCV, the warpAffine() function can be used to translate an image.
- Noise: Generating random noise and scaling it based on the input image and clipping the image within its valid range.
- Blur: Applying gaussian blur to the input image.
- Grayscale: Converting images to grayscale using OpenCV's cvtColor, simplifying the input data by removing the colour information.
- Colour hue: Converting the image to HSV colour space and then adding the hue value to the hue channel, converting the image back to BGR colour space.
- Skew: Skewing the image at a given angle to simulate the distortion or misalignments caused during the scanning of the image.

The above functions were applied in a random variation to each one of the 4 report formats selected so as to increase the size of the dataset for the model training.

Since the model is to be trained for object detection for a similarly structured dataset, augmentation on the same images didn't have any adverse effect on the accuracy.

The selected report formats:



SAMPLE

Kamal Diagnostic Center

Patient Name:	Mrs. Dummy	Registered on:	09/08/2022 11:54 AM
Age / Sex:	34 YRS / F	Collected on:	09/08/2022
Referred By:	Dr. Self	Received on:	12/08/2022
Reg. no. / UHID:	1002 /	Reported on:	09/08/2022 11:54 AM
Investigations:	Complete Blood Count (CBC)		

1002

HAEMATOLOGY

COMPLETE BLOOD COUNT (CBC)

TEST		VALUE	UNIT	REFERENCE
Hemoglobin	L	**10.0**	**g/dl**	**12 - 15**
Total Leukocyte Count		5,700	cumm	4,000 - 11,000
Differential Leucocyte Count				
Neutrophils		55	%	40 - 80
Lymphocyte		40	%	20 - 40
Eosinophils		02	%	1 - 6
Monocytes		03	%	2 - 10
Basophils		00	%	< 2
Platelet Count		2.8	lakhs/cumm	1.5 - 4.5
Total RBC Count		4.7	million/cumm	3.9 - 4.8
Hematocrit Value, Hct		45	%	36 - 46
Mean Corpuscular Volume, MCV		95.7	fL	83 - 101
Mean Cell Haemoglobin, MCH	L	**21.3**	**Pg**	**27 - 32**
Mean Cell Haemoglobin CON, MCHC	L	**22.2**	**%**	**31.5 - 34.5**
Mean Platelet Volume, MPV		11	fL	6.5 - 12
R.D.W. - SD		40	fL	39 - 46
R.D.W. - CV		12	%	11.6 - 14
P-LCR		33	%	19.7 - 42.4
P.D.W.		10.8	fL	9.6 - 15.2

--- End of report ---

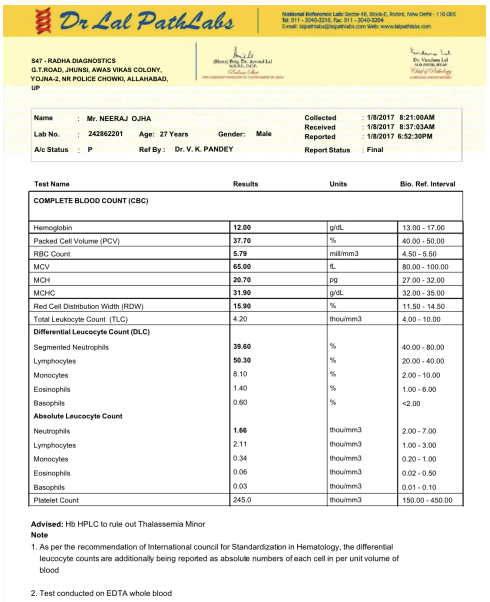
LABSMART SAMPLE REPORT

Mr. Sachin Sharma
DMLT, Lab Incharge

Dr. A. K. Aathana
MBBS, MD Pathologist

Page 1 of 1

Opposite Hospital
Hospital Road, New Delhi
011-03849283



Dr Lal PathLabs

National References Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3045-3255, Fax: 011 - 3045-3264
E-mail: lalpathlabs@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

Name	: Mr. NEERAJ OJHA			Collected	: 1/8/2017 8:21:00AM
Lab No.	: 242862201	Age: 27 Years	Gender: Male	Received	: 1/8/2017 8:37:03AM
				Reported	: 1/8/2017 6:52:30PM
A/c Status	: P	Ref By :	Dr. V. K. PANDEY	Report Status	: Final

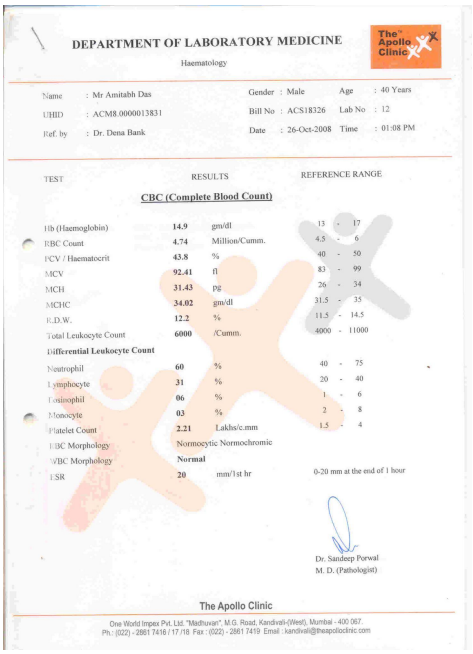
Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
Hemoglobin	**12.00**	g/dL	13.00 - 17.00
Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	**39.60**	%	40.00 - 80.00
Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
Platelet Count	245.0	thou/mm3	150.00 - 450.00

Advised: Hb HPLC to rule out Thalassemia Minor

Note

1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

2. Test conducted on EDTA whole blood



DEPARTMENT OF LABORATORY MEDICINE

Haematology

The Apollo Clinic

Name	: Mr Amitabh Das	Gender : Male	Age : 40 Years
UHID	: ACM8.0000013831	Bill No : ACS18326	Lab No : 12
Ref. by	: Dr. Dena Bank	Date : 26-Oct-2008	Time : 01:08 PM

CBC (Complete Blood Count)

TEST	RESULTS		REFERENCE RANGE
Hb (Haemoglobin)	**14.9**	gm/dl	13 - 17
RBC Count	**4.74**	Million/Cumm.	4.5 - 6
PCV / Haematocrit	**43.8**	%	40 - 50
MCV	**92.41**	fl	83 - 99
MCH	**31.43**	pg	26 - 34
MCHC	**34.02**	gm/dl	31.5 - 35
R.D.W.	**12.2**	%	11.5 - 14.5
Total Leukocyte Count	**6000**	/Cumm.	4000 - 11000
Differential Leukocyte Count			
Neutrophil	**60**	%	40 - 75
Lymphocyte	**31**	%	20 - 40
Eosinophil	**06**	%	1 - 6
Monocyte	**03**	%	2 - 8
Platelet Count	**2.21**	Lakhs/c.mm	1.5 - 4
RBC Morphology	Normocytic Normochromic		
WBC Morphology	**Normal**		
ESR	**20**	mm/1st hr	0-20 mm at the end of 1 hour

Dr. Sandeep Porwal
M. D. (Pathologist)

The Apollo Clinic

One World Impex Pvt. Ltd. "Madhuvan", M.G. Road, Kandivali-(West), Mumbai - 400 067.
Ph.: (022) - 2861 7416 / 17 /18 Fax : (022) - 2861 7419 Email : kandivali@theapolloclinic.com

SMART PATHOLOGY LAB

Accurate | Caring | Instant

9123456789 / 8912345678
smartpatholab@gmail.com

105 -108, SMART VISION COMPLEX, HEALTHCARE ROAD, OPPOSITE HEALTHCARE COMPLEX. MUMBAI - 689578

Yash M. Patel
Age : 21 Years
Sex : Male
PID : 555

Sample Collected At:
125, Shivam Bungalow, S G Road,
Mumbai
Ref. By: **Dr. Hiren Shah**

Registered on: 02:31 PM 02 Dec, 22
Collected on: 03:11 PM 02 Dec, 22
Reported on: 04:35 PM 02 Dec, 22

Complete Blood Count (CBC)

Investigation	Result		Reference Value	Unit
Primary Sample Type :	Blood			
HEMOGLOBIN				
Hemoglobin (Hb)	**12.5**	Low	13.0 - 17.0	g/dL
RBC COUNT				
Total RBC count	5.2		4.5 - 5.5	mill/cumm
BLOOD INDICES				
Packed Cell Volume (PCV)	**57.5**	High	40 - 50	%
Mean Corpuscular Volume (MCV) Calculated	87.75		83 - 101	fL
MCH Calculated	27.2		27 - 32	pg
MCHC Calculated	32.8		32.5 - 34.5	g/dL
RDW	13.6		11.6 - 14.0	%
WBC COUNT				
Total WBC count	9000		4000-11000	cumm
DIFFERENTIAL WBC COUNT				
Neutrophils	60		50 - 62	%
Lymphocytes	31		20 - 40	%
Eosinophils	1		00 - 06	%
Monocytes	7		00 - 10	%
Basophils	1		00 - 02	%
PLATELET COUNT				
Platelet Count	320000		150000 - 410000	cumm

Instruments: Fully automated cell counter - Mindray 300

Interpretation: Further confirm for Anemia

Thanks for Reference

****End of Report****

Medical Lab Technician
(DMLT, BMLT)

Dr. Payal Shah
(MD, Pathologist)

Dr. Vimal Shah
(MD, Pathologist)

Generated on : 02 Dec, 2022 05:00 PM

Page 1 of 1

Sample Collection 9123456789

7.3 Module 1: Object Detection

Object detection models are used while working with our scanned blood reports to find important objects inside the document. In our case these objects include a set of the label and the value (Eg: Haemoglobin: 13). The patterns, forms, and traits associated with the items of interest, such as certain values or data points in the blood report, are taught to these models during training. The model can precisely extract the desired information by learning the locations of these items. In order to train object detection models for scanned blood reports, our manually annotated data is given to the models. Images of blood reports that have had the relevant objects, such as values or data points, noted or highlighted are known as annotated data. The model then learns to connect the visual characteristics of these tagged items with the labels or categories that belong to them.

Roboflow AutoML: We leveraged Roboflow's AutoML product to streamline the process of using our annotated data to train an object detection model. AutoML optimizes the model's performance against the provided data and provides detailed metrics and visualizations for model evaluation. We chose Roboflow as it allows users to upload their labeled image datasets in various annotation formats and provides tools for data augmentation, data cleaning, and preprocessing.

YOLOv8: The YOLOv8 object detection model is what we chose to employ. A family of real-time object identification models for computer vision known as YOLO (You Only Look Once) work through full pictures or videos in a single pass, thus the name. In order for YOLO to work, each picture must first be divided into a grid, after which bounding boxes and class probabilities must be predicted for each grid cell. This technique enables YOLO to swiftly and precisely detect objects, including those that are partially hidden. The most recent version, YOLOv8, was published in 2021 and is the one we're using.

The 'backbone' and 'head' of the fully convolutional neural network used by YOLOv8 may be considered its two fundamental components. YOLOv8 is built on a modified version of the CSPDarknet53 architecture. This design, which consists of 53 convolutional layers, makes advantage of cross-stage partial connections to improve information flow across the network's various tiers. The YOLOv8 algorithm's head consists of a number of convolutional layers and a series of fully linked layers. Forecasting the object detection bounding boxes, objectness ratings, and class probabilities fall within the purview of these layers. One of YOLOv8's distinctive features is the use of a self-attention mechanism in the network's brain.

With the aid of this mechanism, the model is able to change its attention to various aspects of the image according to their relevance to the task at hand. The capability of YOLOv8 to carry out multi-scale object detection is another crucial feature. The model makes use of a feature pyramid network to find items in an image that are different sizes and scales. The numerous layers of this feature pyramid network recognise items at various scales, enabling the model to recognise both large and small things in an image.

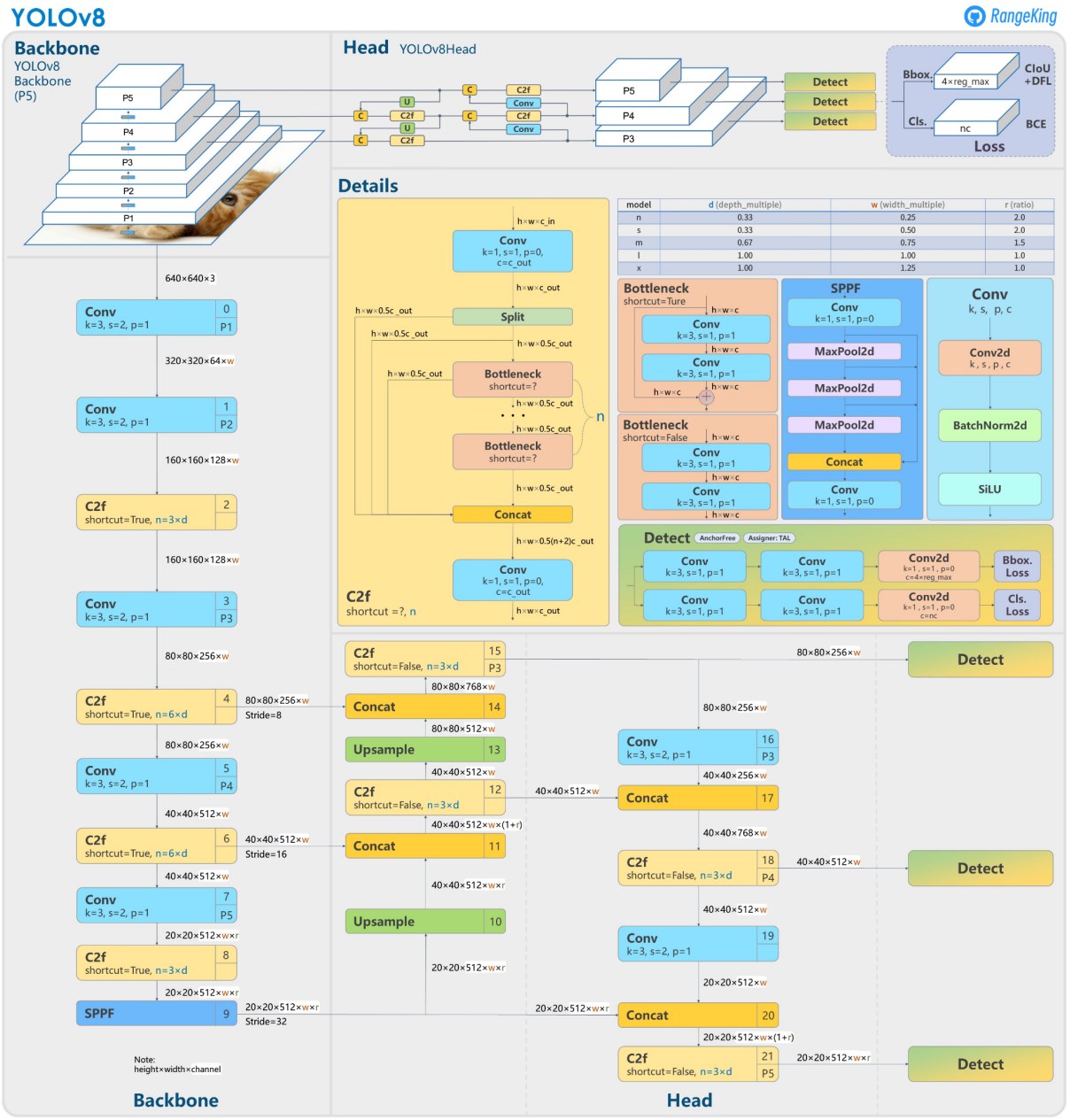
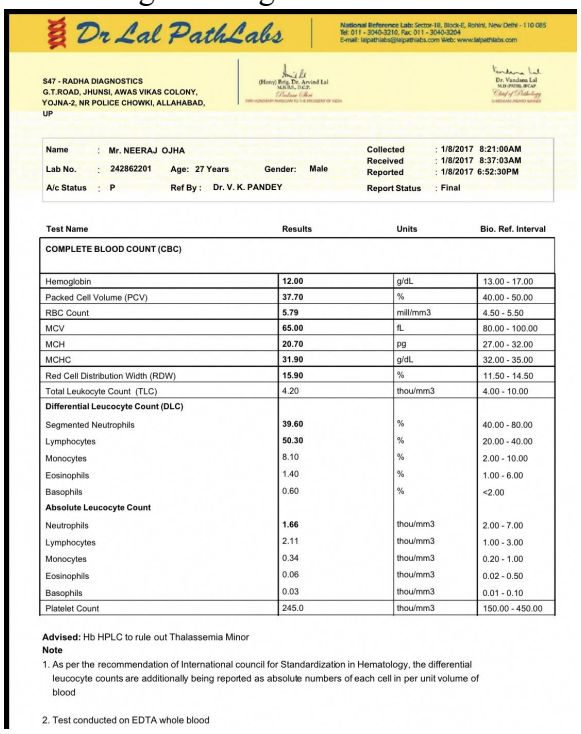


Fig 7.1. YOLOv8 Architecture

Input for module 1

1. Original image



Dr Lal PathLabs

National Reference Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3040-3210, Fax: 011 - 3040-3204
E-mail: lalpathlabs@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

(Hony) Brig. Dr. Arvind Lal
M.B.B.S., M.D.P.
Pathologist

Dr. Vandana Lal
M.D.(PATH), DCP
Chief of Pathology

Name	: Mr. NEERAJ OJHA			Collected	: 1/8/2017 8:21:00AM
Lab No.	: 242862201	Age: 27 Years	Gender: Male	Received	: 1/8/2017 8:37:03AM
				Reported	: 1/8/2017 6:52:30PM
A/c Status	: P	Ref By : Dr. V. K. PANDEY		Report Status	: Final

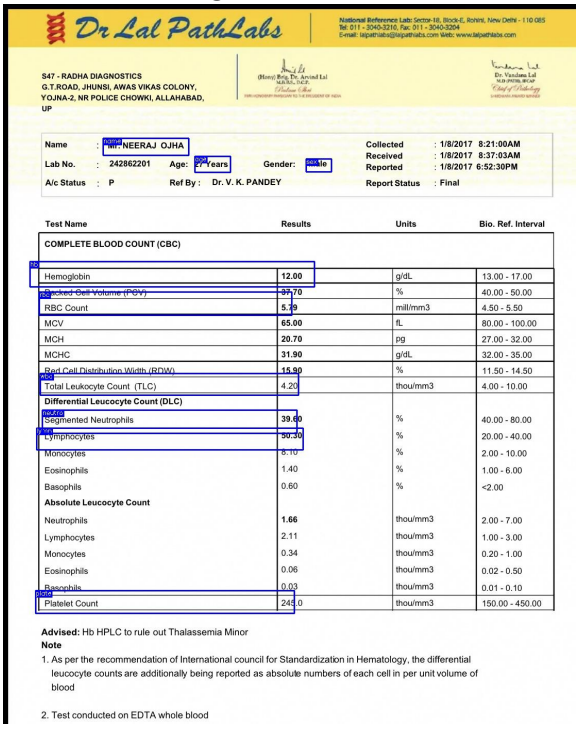
Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
Hemoglobin	**12.00**	g/dL	13.00 - 17.00
Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	**39.60**	%	40.00 - 80.00
Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
Platelet Count	245.0	thou/mm3	150.00 - 450.00

Advised: Hb HPLC to rule out Thalassemia Minor
Note
1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

2. Test conducted on EDTA whole blood

Output from module 1

1. Original image with predicted bounding boxes:



Dr Lal PathLabs

National Reference Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3040-3210, Fax: 011 - 3040-3204
E-mail: lalpathlabs@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

(Hony) Brig. Dr. Arvind Lal
M.B.B.S., M.D.P.
Pathologist

Dr. Vandana Lal
M.D.(PATH), DCP
Chief of Pathology

Name	: name Mr. NEERAJ OJHA			Collected	: 1/8/2017 8:21:00AM
Lab No.	: 242862201	Age: age 27 Years	Gender: sex male	Received	: 1/8/2017 8:37:03AM
				Reported	: 1/8/2017 6:52:30PM
A/c Status	: P	Ref By : Dr. V. K. PANDEY		Report Status	: Final

Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
hb Hemoglobin	**12.00**	g/dL	13.00 - 17.00
hct Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
wbc Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
neutro Segmented Neutrophils	**39.60**	%	40.00 - 80.00
lymp Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
plate Platelet Count	245.0	thou/mm3	150.00 - 450.00

Advised: Hb HPLC to rule out Thalassemia Minor
Note
1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

2. Test conducted on EDTA whole blood

2. (json): coordinates (x,y,w,h) and confidence of the classes detected by the model.

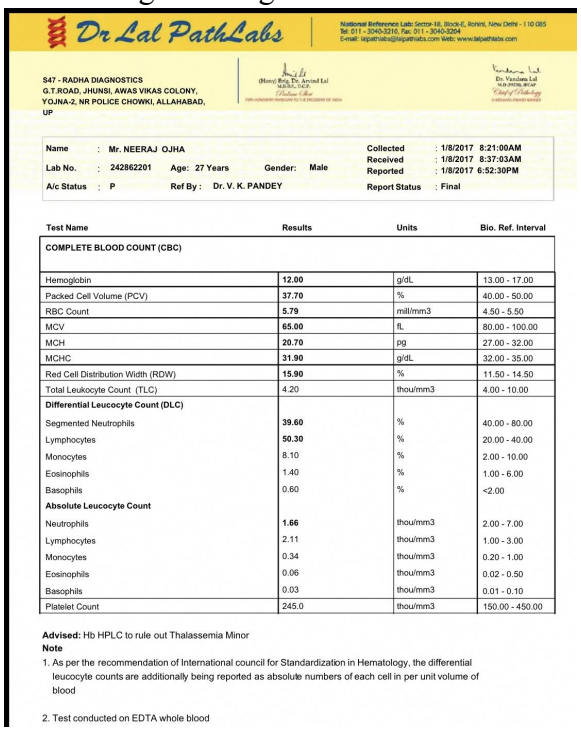

```
{'predictions': [{'x': 322.5, 'y': 737.0, 'width': 501.0, 'height': 42.0, 'confidence': 0
.8670761585235596, 'class': 'wbc', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 277.0, 'y': 276.5, 'width': 166.0, 'height': 33.0, 'confidence':
0.8346527814865112, 'class': 'name', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 315.0, 'y': 580.5, 'width': 490.0, 'height': 45.0, 'confidence':
0.7787464261054993, 'class': 'rbc', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 328.0, 'y': 523.5, 'width': 552.0, 'height': 49.0, 'confidence':
0.7568410634994507, 'class': 'hb', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 323.5, 'y': 843.0, 'width': 517.0, 'height': 42.0, 'confidence':
0.7463908195495605, 'class': 'lymp', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 314.0, 'y': 1159.0, 'width': 502.0, 'height': 44.0, 'confidence'
0.7463420629501343, 'class': 'plate', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 406.5, 'y': 310.5, 'width': 69.0, 'height': 31.0, 'confidence':
.7365566492080688, 'class': 'age', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 612.5, 'y': 312.5, 'width': 49.0, 'height': 25.0, 'confidence':
.69354353352356, 'class': 'sex', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 323.0, 'y': 809.5, 'width': 494.0, 'height': 45.0, 'confidence':
0.5823017954826355, 'class': 'neutro', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}], 'image': {'width': '1112', 'height': '1396'}}
```

7.4 Module 2: Image Processing

The main aim of our Image Processing module is to take the output of our object recognition module and use it to separate our single scanned image into multiple smaller cropped images. Cropping is a key component of many computer vision tasks, especially Optical Character Recognition (OCR) tasks. To increase the accuracy of the OCR output and lower noise, it is frequently required to crop or isolate particular regions of interest, such as the prediction regions. Once the object detection model has determined the location of the desired objects, i.e. the x, y, h(height), w(width) coordinates are stored in the output json file and used to crop the image.

Input for module 2	Output from module 2

1. Original image:



Dr Lal PathLabs

National Reference Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3040-3210, Fax: 011 - 3040-3204
E-mail: labpathlab@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

(Hon) Brig. Dr. Arvind Lal
M.B.B.S., D.C.P.
Medical Director

Dr. Vandana Lal
M.D. (PATH), DCP
Chief of Pathology

Name	: Mr. NEERAJ OJHA			Collected	: 1/8/2017 8:21:00AM
Lab No.	: 242862201	Age: 27 Years	Gender: Male	Received	: 1/8/2017 8:37:03AM
				Reported	: 1/8/2017 6:52:30PM
A/c Status	: P	Ref By : Dr. V. K. PANDEY		Report Status	: Final

Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
Hemoglobin	**12.00**	g/dL	13.00 - 17.00
Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	**39.60**	%	40.00 - 80.00
Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
Platelet Count	245.0	thou/mm3	150.00 - 450.00

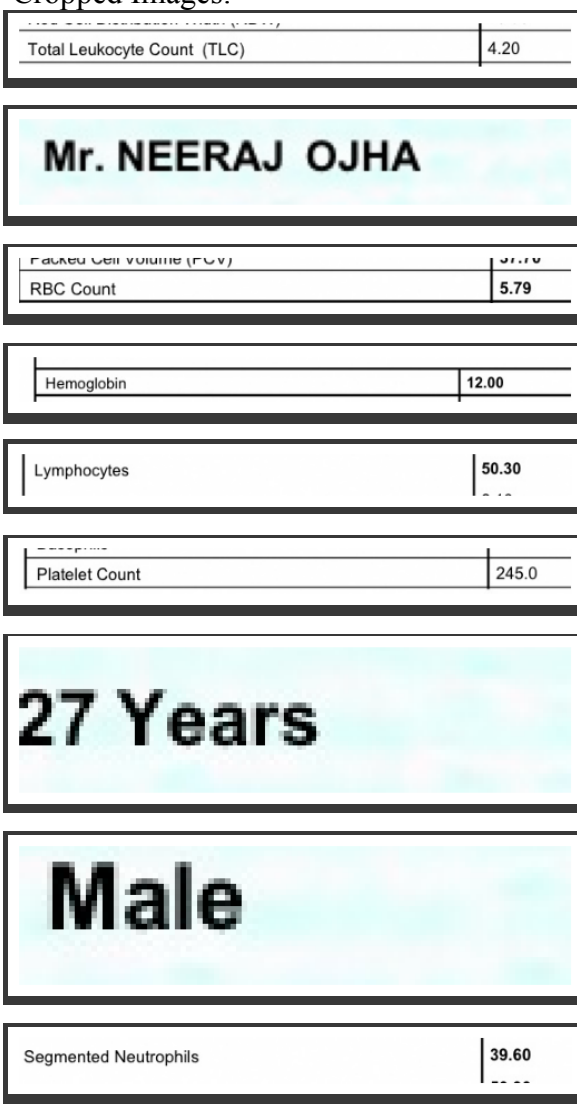
Advised: Hb HPLC to rule out Thalassemia Minor
Note
1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

2. Test conducted on EDTA whole blood

2. Coordinates for the classes detected by the module 1:


```
{'predictions': [{'x': 322.5, 'y': 737.0, 'width': 501.0, 'height': 42.0, 'confidence': 0
.8670761585235596, 'class': 'wbc', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 277.0, 'y': 276.5, 'width': 166.0, 'height': 33.0, 'confidence':
0.8346527814865112, 'class': 'name', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 315.0, 'y': 580.5, 'width': 490.0, 'height': 45.0, 'confidence':
0.7787464261054993, 'class': 'rbc', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 328.0, 'y': 523.5, 'width': 552.0, 'height': 49.0, 'confidence':
0.7568410634994507, 'class': 'hb', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 323.5, 'y': 843.0, 'width': 517.0, 'height': 42.0, 'confidence':
0.7463908195495605, 'class': 'lymp', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 314.0, 'y': 1159.0, 'width': 502.0, 'height': 44.0, 'confidence':
0.7463420629501343, 'class': 'plate', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 406.5, 'y': 310.5, 'width': 69.0, 'height': 31.0, 'confidence': 0
.7365566492080688, 'class': 'age', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 612.5, 'y': 312.5, 'width': 49.0, 'height': 25.0, 'confidence': 0
.693543553352356, 'class': 'sex', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}, {'x': 323.0, 'y': 809.5, 'width': 494.0, 'height': 45.0, 'confidence':
0.5823017954826355, 'class': 'neutro', 'image_path': '4.jpg', 'prediction_type':
'ObjectDetectionModel'}], 'image': {'width': '1112', 'height': '1396'}}
```


Cropped Images:



7.5 Module 3: OCR Text Extraction

After the cropping process, the next step in the process was extracting the text from the cropped images. The text extraction was done using pre-trained OCR models. We performed text extraction using both the EasyOCR and Tesseract models and compared their performance. Our model of choice was ultimately EasyOCR as it gave us better results for our problem statement.

Easy OCR

Combining deep learning methods, EasyOCR uses LSTM for sequence labelling, ResNet for feature extraction, and the Connectionist Temporal Classification (CTC) algorithm for decoding.

Feature Extraction: ResNet is utilised in EasyOCR for feature extraction. In order to learn hierarchical representations of visual features, it is first pretrained on large-scale image classification tasks, such as the ImageNet dataset.

Sequence Labelling: EasyOCR uses Long Short-Term Memory (LSTM) networks for sequence labelling. LSTMs are used in the context of OCR to identify and label each distinct character or symbol among the retrieved characteristics. The input properties of the annotated data used to train the LSTM networks are connected to the corresponding character labels. The networks learn to capture the dependencies and context within the sequence of extracted characteristics by utilising the memory cells and gates within the LSTM architecture, enabling precise character recognition.

Decoding (CTC): After the LSTM networks have independently tagged each character, EasyOCR decodes the sequence using the Connectionist Temporal Classification (CTC) method. The OCR output is varied in length, and the CTC algorithm manages sequences with repeated or missing characters. It provides a probability distribution for each possible letter sequence while also taking into consideration the labelling produced by the LSTM networks. EasyOCR can convert the series of character labels into the final recognised text using the CTC decoding technique.

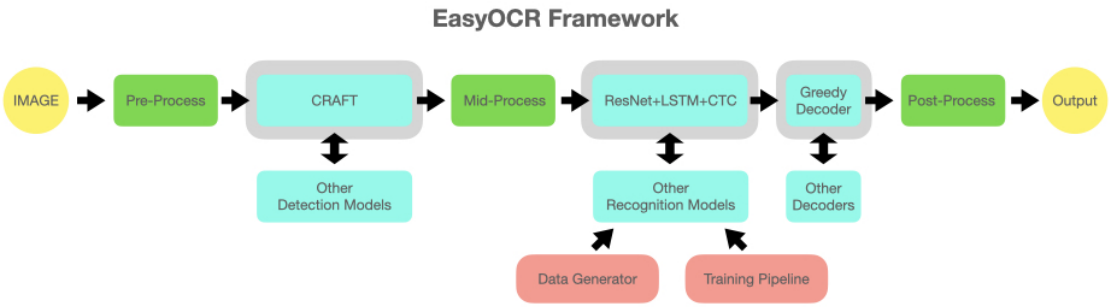


Fig 7.2. EasyOCR Framework

Tesseract

A new neural network subsystem configured as a text line recognizer is part of Tesseract 4.00. It is based on the Python-based LSTM implementation from OCRopus and was adapted for Tesseract in C++. CNNs are frequently used to identify images with a single character. To overcome these issues, RNNs—of which LSTM is a well-liked variant—should be utilised in many situations, including ours, where the character sequences make up text of any length. For this reason, we chose LSTM-based models. Preprocessing on the input picture comprises scaling, deskewing, binarization, and noise reduction. After that, character segmentation is done to isolate certain characters. Then, Tesseract uses its deep learning model to extract information from each character. These networks are trained using character images that have been labelled, then backpropagation is used to alter internal parameters to increase recognition accuracy. The trained model receives the extracted features during classification and assigns probability to each character class. The recognised output is determined by the character with the highest likelihood.

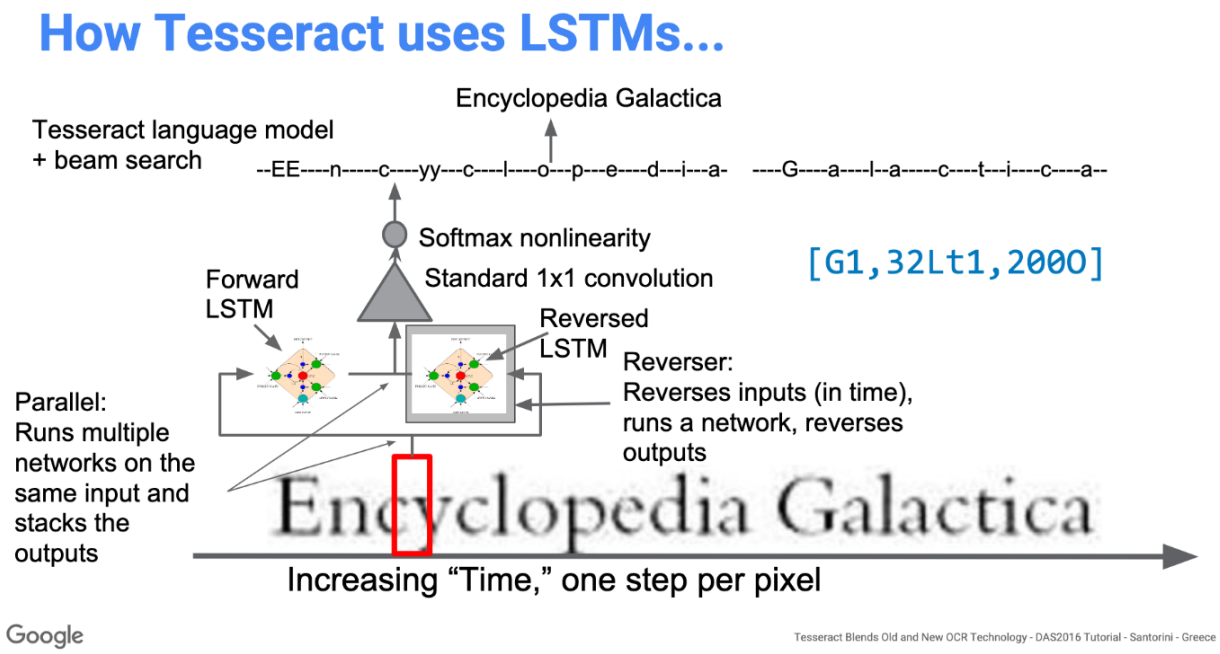
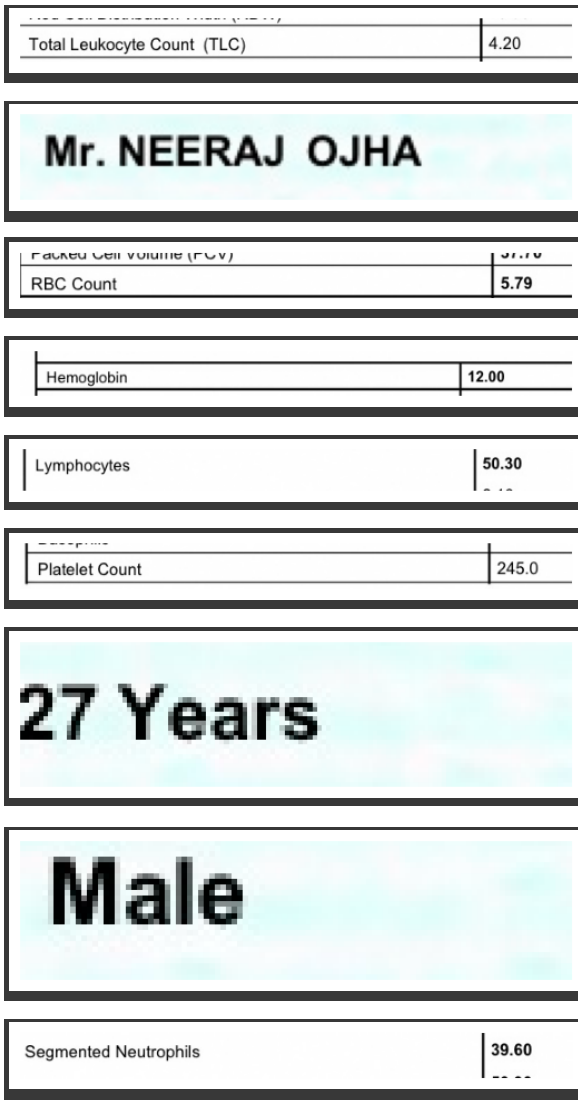


Fig 7.3. Tesseract OCR Architecture

<table>
<tr><th>Input for module 3</th><th>Output from module 3</th></tr>
<tr><td>Cropped Images (from module 2)
 
</td><td>1. Easy OCR output

['Total Leukocyte Count (TLC)', '4.20']

['Mr. NEERAJ OJHA']

['Tacneu', 'VUiunt', 'RBC Count', '5.79']

['Hemoglobin', '12.00']

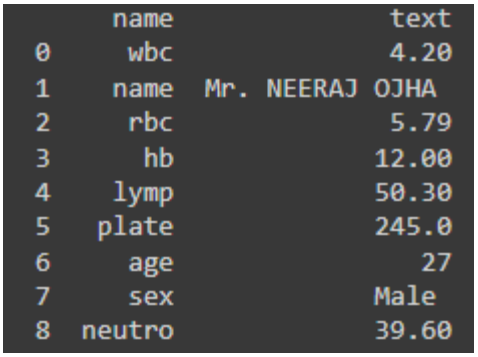
['Lymphocytes', '50.30']

['Platelet Count', '245.0']

['27 Years']

['Male']

['Segmented Neutrophils', '39.60']

2. Key values extraction from the OCR result:
 
</td></tr>
</table>

7.6 Module 4: Report Generation

The last stage of our problem statement was to generate a report based on the extracted values. We collected information about the normal ranges of values, potential causes and suggested treatments based on age and sex. The medical information was collected from research papers, Indian lab reports and other reputed sources.

General Values & Likely Causes: [5], [6], [7], [8]
For Age and Gender based Normal Values: [1], [2], [3]
WBC Counts for Elderly : [4]
RBC Counts for Adults: [9]

RBC Counts for Adolescents: [10]
Normal Ranges for Children: [11], [12]

Values from the OCR model were passed to a newly structured Dataframe. The values were then processed by age-group as different age groups had different normal ranges and different likely causes. (Eg: Smoking is an unlikely cause in children). For some values ('RBC Count' and 'Haemoglobin'), the patient's sex was also taken into account as the differences between the normal ranges for each gender was significant. We maintained 4 normal-range tables, one for each age group. The age groups were divided as follows:

Children: 0-10 years
Adolescent: 10-18 years
Adult: 18-50 years
Elderly: 50+ years

Each item's values were calculated as either HIGH, LOW or Normal based on the normal-range table. The difference in normal ranges between Children and Adults can be seen below:

A	B	C	D	E	F	G	H	I	J
value	sex	age	wbc	rbc	lymp	neutro	plate	hb	Column1
high			30.00	5.90	70.00	65.00	450.00	20.00	Age Group: 0-10yrs
low			5.00	2.70	20.00	20.00	150.00	9.00	
Causes_high			Bacterial/Parasitic Infection, Leukemia, Inflammation	Dehydration, Lung or Heart issues, Kidney diseases, Blood	Viral/bacterial, parasitic/fungal infections, Inherited	Acute Bacterial Infection, Some Viral Infections, Inflammation,	Recent Blood Loss, Mild to moderate iron	Smoking, Living at higher altitudes, Dehydration,	
Causes_low			Viral Infection, Blood/Bone Disorder, Autoimmune Condition, Chemo/Radiation	Anemia, Vitamin B-12 or folate deficiency,	Viral Infections (such as HIV/AIDS/Influenza/COVID),	Infection (Generally Bacterial, commonly -	Bacterial Infection, Autoimmune Condition, Viral	Anemia, Iron or Vitamin B-12 or folate deficiency,	
solutions_high			High counts caused by recent infections will generally resolve naturally.	Professional Assistance Required: Follow up with	High counts caused by recent infections can	High counts caused by recent infections can	For Treating High Platelet Counts related to Iron	For Mild Cases: Increase water intake, Avoid supplements	
solutions_low			Low counts caused by recent infections will generally resolve naturally.	For Treating Mild Anaemia - B-12 Deficiency:	Acute cases will resolve naturally.	Acute cases will resolve naturally.	Professional Assistance Required: Follow up with	For Treating Mild Anaemia - B-12 Deficiency:	
value	sex	age	wbc	rbc	lymp	neutro	plate	hb	Column1
high			10.50	5.00	40.00	75.00	450.00	17.00	Age Group: 18-50yrs
low			4.50	4.50	20.00	40.00	150.00	13.00	
Causes_high			Bacterial/Parasitic Infection, Leukemia, Steroid Use, Pregnancy,Inflammation	Dehydration, Lung or Heart issues, Kidney diseases, Blood	Recent infection (most commonly viral),	Acute Bacterial Infection, Some Viral Infections, Inflammation,	Recent Blood Loss, Mild to moderate iron	Living at higher altitudes, Dehydration, Heart/Lung	
Causes_low			Viral Infection, Blood/Bone Disorder, Autoimmune Condition, Chemo/Radiation	Anemia, Vitamin B-12 or folate deficiency,	Blood diseases like Hodgkin's disease, Autoimmune	Infection (Generally Bacterial, commonly -	Viruses such as mononucleosis/ HIV/AIDS,	Anemia, Iron or Vitamin B-12 or folate deficiency,	
solutions_high			High counts caused by recent infections will generally resolve naturally.	For mild cases: Exercise can help improve heart and lung	High counts caused by recent infections can	Acute cases will resolve naturally.	For Treating High Platelet Counts related to Iron	For Mild Cases: Increase water intake, Avoid supplements,	
solutions_low			Low counts caused by recent infections will generally resolve naturally.	For Treating Mild Anaemia - B-12 Deficiency:	Low counts caused by recent infections will	Acute cases will resolve naturally.	Professional Assistance Required: Follow up with	For Treating Mild Anaemia - B-12 Deficiency:	

For any abnormal (HIGH/LOW) values, a more detailed report was generated. This included

the percentage deviation from the normal value, the most likely causes for the deviation and the suggested advice. Deviations were considered 'MILD' if deviation was lower than 10%, otherwise it was categorised as 'Significant'.

7.7 Module 5: GUI

To connect all the modules together, provide a simple user interface which will allow the user to upload a blood report image and get the structured pdf inference based off of it. We decided to go with flask which is a web application framework developed in python.

The complete directory of the application looks like the figure mentioned below:

```
D:.
ª   app.py
ª   best.pt
+---env1
ª   +---Scripts
ª   +---share
ª       +---doc
ª       +---man
+---report_generation
ª       normal_1.xlsx
ª       normal_2.xlsx
ª       normal_3.xlsx
ª       normal_4.xlsx
+---static
ª   ª   logo.png
ª   ª   prediction.jpg
ª   +---predict
ª   ª       uploaded_image.jpg
+---templates
ª       index.html
ª       result.html
+---tmp
ª       report.pdf
ª       temp_html_file.html
ª       uploaded_image.jpg
+---__pycache__
        app.cpython-39.pyc
        hello.cpython-39.pyc
```

Fig 7.4. GUI directory

Where the app.py consists of the program for all the modules (backend), best.pt is the YOLOv8x model file, env1 is the virtual environment, report_generation folder consists of the dataset for the module 4 (Report generation), static folder consists of the images to be displayed on the web page (prediction.jpg- output from Roboflow model, predict-output from YOLOv8x), templates consists of the frontend html pages, tmp consists of the temporary data that gets generated during the runtime.

Homepage:

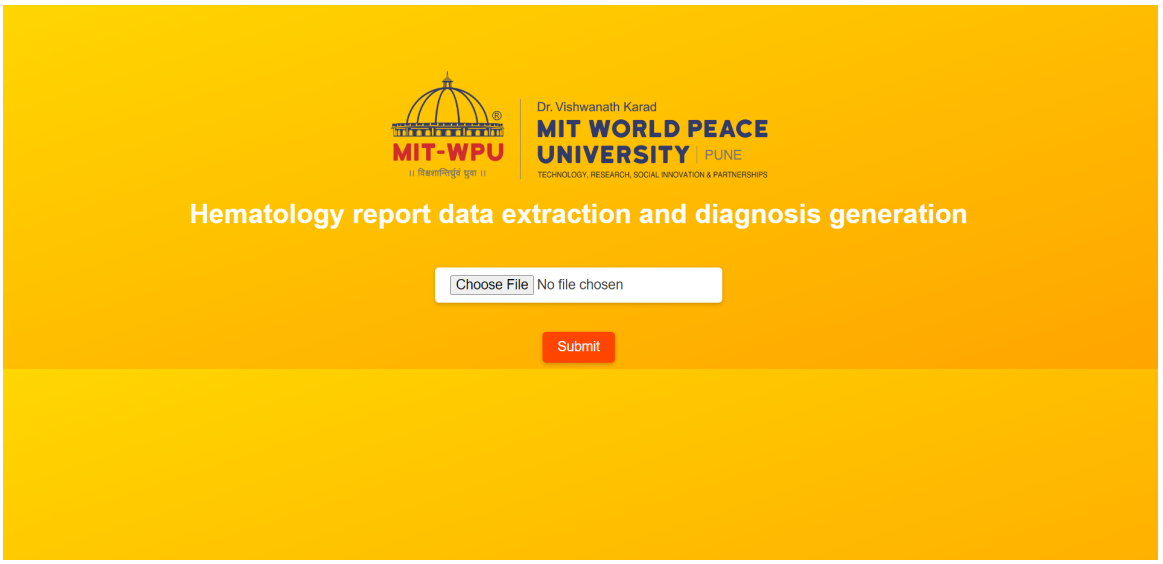


Fig 7.5. Homepage

Result:

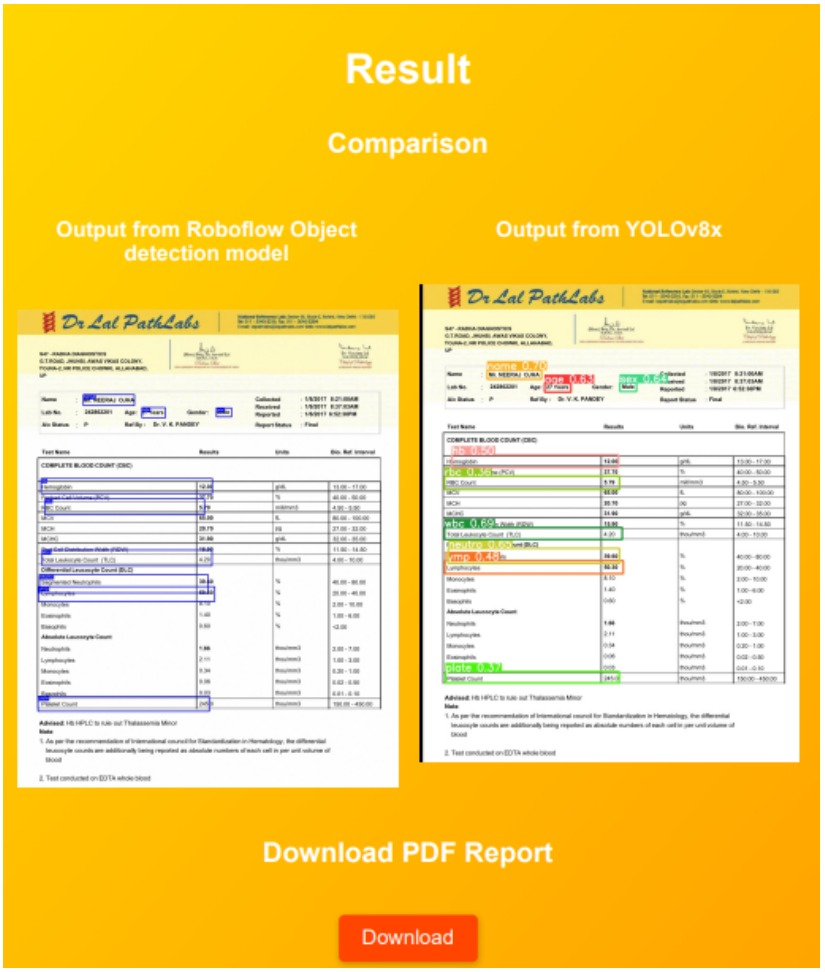


Fig 7.6. Result page

Downloaded report:

1 / 1 | − 100% + |

```
-----------------------------------------------------------------------Generated Report-----------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------------------------------------
GENERAL INFORMATION
-----------------------------------------------------------------------------------------------------------------------------------------------------------------

Name: Mr. NEERAJ OJHA
Sex: Male
Age: 27
Age Group: Adult (18-50 years old)

-----------------------------------------------------------------------------------------------------------------------------------------------------------------
ALL RESULTS
-----------------------------------------------------------------------------------------------------------------------------------------------------------------

WBC Level: LOW
RBC Level: Normal
LYMPHOCYTE Level: HIGH
NEUTROPHIL Level: LOW
PLATELET Level: Normal
HAEMOGLOBIN Level: LOW

-----------------------------------------------------------------------------------------------------------------------------------------------------------------
DETAILED REPORT OF ABNORMAL RESULTS
-----------------------------------------------------------------------------------------------------------------------------------------------------------------

1.
WBC Level:
 - 6.67 % LOWER than normal
 - MILD deviation from normal range

Potential Causes: (Common causes are listed first)
Viral Infection, Blood/Bone Disorder, Autoimmune Condition,  Chemo/Radiation Therapy, Medication Side Effect


2.
LYMPHOCYTE Level:
 - 25.75 % HIGHER than normal
 - !! Significant deviation from normal range

Potential Causes: (Common causes are listed first)
Recent infection (most commonly viral),
Long-term inflammation (such as rheumatoid arthritis),
Medication Side Effect, Trauma,
Spleen Removal, Certain types of cancer (like leukemia or lymphoma)


Suggestions:
High counts caused by recent infections can resolve naturally.

For more severe symptoms contact a medical professional.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------

3.
NEUTROPHIL Level:
 - 1.0 % LOWER than normal
 - MILD deviation from normal range

Potential Causes: (Common causes are listed first)
Infection (Generally Bacterial, commonly -  Tuberculosis, Dengue, HIV), Inflammation, Chemotherapy, Blood Cancer (rare), Leukemia (rare)


4.
HAEMOGLOBIN Level:
 - 14.29 % LOWER than normal
 - !! Significant deviation from normal range
```

Fig 7.7. Download report

Chapter 8
Coding

8.1 Data augmentation

As explained above (6.2) about the data preprocessing and data generation procedure, OpenCV library was extensively used to augment the 4 report formats selected to populate the dataset.

```python
import cv2
import numpy as np
import random
import os

# Rotation
def rotate(image, angle):
    # Calculate image center
    center = tuple(np.array(image.shape[1::-1]) / 2)
    # Build the rotation matrix
    matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
    # Perform the actual rotation and return the result
    return cv2.warpAffine(image, matrix, image.shape[1::-1], flags=cv2.INTER_LINEAR)

def add_noise(image, scale=0.1):
    """
    Add random noise to the input image.
    """
    # Generate random noise and scale it based on the input image
    noise = np.random.normal(scale=scale, size=image.shape)
    # Add the noise to the image
    noisy_image = image + noise
    # Clip the pixel values to be within the valid range [0, 255]
    noisy_image = np.clip(noisy_image, 0, 255).astype(np.uint8)
    return noisy_image

def blur(image, kernel_size=3):
    """
    Apply Gaussian blur to the input image.
    """
    # Apply Gaussian blur to the image using the specified kernel size
    blurred_image = cv2.GaussianBlur(image, (kernel_size, kernel_size), 0)
    return blurred_image

def grayscale(image):
    if image.shape[-1] > 1:
        result = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        result = image
    return result

def color_hue(image, hue):
    if image.shape[-1] >= 3:
        # Convert the image to HSV color space
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
        # Add the hue value to the Hue channel
        hsv[:, :, 0] = (hsv[:, :, 0] + hue) % 180
        # Convert the image back to BGR color space
        result = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    else:
        result = image
    return result
```

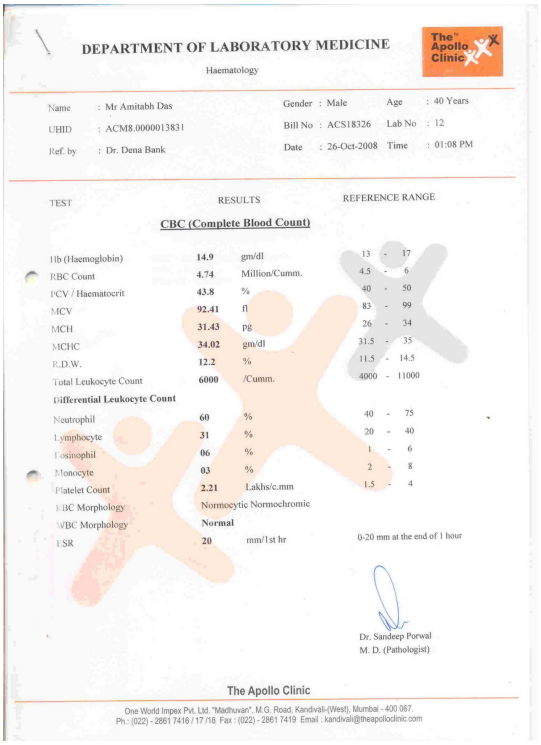
```python
def skew(image, angle=1, scale=1.0):
    """
    Skew the input image by a given angle.
    """
    # Define the transformation matrix
    matrix = np.float32([[1, 0, 0], [0, 1, 0]])
    angle_radians = angle * np.pi / 180.0
    matrix[0, 1] = np.tan(angle_radians)
    matrix[1, 0] = np.tan(angle_radians)
    # Apply the transformation to the image
    skewed_image = cv2.warpAffine(image, matrix, (image.shape[1], image.shape[0]))
    return skewed_image

def translate_x(image, shift=10):
    """
    Translate the input image on the x-axis by a given number of pixels.
    """
    # Define the transformation matrix
    matrix = np.float32([[1, 0, shift], [0, 1, 0]])
    # Apply the transformation to the image
    translated_image = cv2.warpAffine(image, matrix, (image.shape[1], image.shape[0]))
    return translated_image

def translate_y(image, shift=10):
    """
    Translate the input image on the y-axis by a given number of pixels.
    """
    # Define the transformation matrix
    matrix = np.float32([[1, 0, 0], [0, 1, shift]])
    # Apply the transformation to the image
    translated_image = cv2.warpAffine(image, matrix, (image.shape[1], image.shape[0]))
    return translated_image

# Apply one or more image variations to an input image and save the result
def apply_image_variations(img, output_path, variations):
    # Load the input image
    input_image = img

    # Apply each specified variation to the image
    for variation in variations:
        if variation == 'rotate_left':
            input_image = rotate(input_image, random_float())
        elif variation == 'rotate_right':
            input_image = rotate(input_image, -1 * random_float())
        elif variation == 'add_noise':
            input_image = add_noise(input_image)
        elif variation == 'blur':
            input_image = blur(input_image)
        elif variation == 'grayscale':
            input_image = grayscale(input_image)
        elif variation == 'color_hue':
            input_image = color_hue(input_image, 30)
        elif variation == 'translate_x':
            input_image = translate_x(input_image)
        elif variation == 'translate_y':
            input_image = translate_y(input_image)
        elif variation == 'skew':
            input_image = skew(input_image, random_float())

    # Save the resulting image to the specified output path
    cv2.imwrite(output_path, input_image)
```

```python
def get_random_elements():
    input_list = ['add_noise', 'color_hue', 'skew', 'translate_y', 'translate_x']
    num_elements = random.randint(1, len(input_list))
    random.shuffle(input_list)
    return input_list[:num_elements]

def random_float():
    return random.uniform(-3, 3)

def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            images.append([filename,img])
    return images

def main():
    FOLDER_NAME = 'Pathology Reports'
    for [filename, image] in load_images_from_folder(FOLDER_NAME):
        for i in range(20):
            variations = get_random_elements()
            output_filename = f"Edited/edited-{filename}_{'-'.join(variations)}.jpg"
            print(f"[Output]: {output_filename}")
            apply_image_variations(image, output_filename, variations)
```

```python
main()
```

```
[Output]: Edited/edited-1.jpg_color_hue-translate_x-add_noise.jpg
[Output]: Edited/edited-1.jpg_skew-color_hue-translate_y-translate_x.jpg
[Output]: Edited/edited-1.jpg_translate_x-add_noise-skew-color_hue.jpg
[Output]: Edited/edited-1.jpg_color_hue-add_noise.jpg
[Output]: Edited/edited-1.jpg_translate_x-add_noise-translate_y-color_hue-skew.jpg
[Output]: Edited/edited-1.jpg_color_hue-add_noise-translate_x-skew.jpg
[Output]: Edited/edited-1.jpg_skew-color_hue.jpg
[Output]: Edited/edited-1.jpg_color_hue-translate_x-skew-add_noise.jpg
[Output]: Edited/edited-1.jpg_skew-color_hue-translate_x-translate_y-add_noise.jpg
[Output]: Edited/edited-1.jpg_skew-translate_x.jpg
[Output]: Edited/edited-1.jpg_translate_y-color_hue-translate_x-skew.jpg
[Output]: Edited/edited-1.jpg_add_noise-translate_x-color_hue-skew-translate_y.jpg
[Output]: Edited/edited-1.jpg_translate_y.jpg
[Output]: Edited/edited-1.jpg_add_noise-skew-color_hue-translate_x-translate_y.jpg
[Output]: Edited/edited-1.jpg_skew-color_hue-translate_y-add_noise.jpg
[Output]: Edited/edited-1.jpg_skew-translate_x.jpg
[Output]: Edited/edited-1.jpg_skew-translate_y.jpg
[Output]: Edited/edited-1.jpg_translate_y-skew-color_hue-add_noise-translate_x.jpg
[Output]: Edited/edited-1.jpg_color_hue-translate_x-skew.jpg
[Output]: Edited/edited-1.jpg_translate_y-translate_x.jpg
[Output]: Edited/edited-10.png_skew-color_hue-translate_x.jpg
[Output]: Edited/edited-10.png_translate_x-translate_y-add_noise-skew-color_hue.jpg
[Output]: Edited/edited-10.png_add_noise-translate_y.jpg
[Output]: Edited/edited-10.png_translate_x-translate_y-add_noise-skew.jpg
[Output]: Edited/edited-10.png_translate_y-translate_x-add_noise.jpg
[Output]: Edited/edited-10.png_color_hue-add_noise.jpg
[Output]: Edited/edited-10.png_translate_y-skew.jpg
[Output]: Edited/edited-10.png_translate_x.jpg
[Output]: Edited/edited-10.png_translate_x.jpg
[Output]: Edited/edited-10.png_skew-add_noise.jpg
[Output]: Edited/edited-10.png_color_hue-translate_x-skew-translate_y.jpg
[Output]: Edited/edited-10.png_translate_x-color_hue-skew-add_noise-translate_y.jpg
[Output]: Edited/edited-10.png_skew-color_hue-add_noise-translate_x.jpg
[Output]: Edited/edited-10.png_translate_x-add_noise-skew-translate_y-color_hue.jpg
[Output]: Edited/edited-10.png_add_noise.jpg
```

Original image



DEPARTMENT OF LABORATORY MEDICINE

Haematology

The Apollo Clinic

Name : Mr Amitabh Das | Gender : Male | Age : 40 Years
UHID : ACM8.0000013831 | Bill No : ACS18326 | Lab No : 12
Ref. by : Dr. Dena Bank | Date : 26-Oct-2008 | Time : 01:08 PM

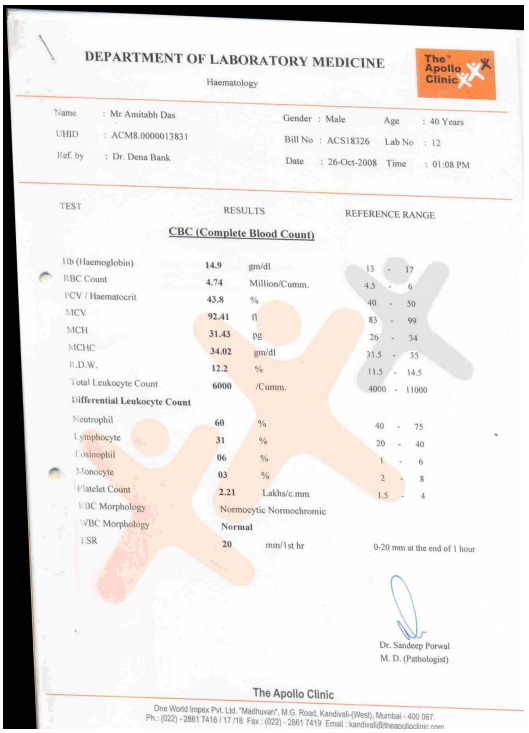
TEST	RESULTS		REFERENCE RANGE
CBC (Complete Blood Count)			
Hb (Haemoglobin)	14.9	gm/dl	13 - 17
RBC Count	4.74	Million/Cumm.	4.5 - 6
PCV / Haematocrit	43.8	%	40 - 50
MCV	92.41	fl	83 - 99
MCH	31.43	pg	26 - 34
MCHC	34.02	gm/dl	31.5 - 35
R.D.W.	12.2	%	11.5 - 14.5
Total Leukocyte Count	6000	/Cumm.	4000 - 11000
Differential Leukocyte Count			
Neutrophil	60	%	40 - 75
Lymphocyte	31	%	20 - 40
Eosinophil	06	%	1 - 6
Monocyte	03	%	2 - 8
Platelet Count	2.21	Lakhs/c.mm	1.5 - 4
RBC Morphology	Normocytic Normochromic		
WBC Morphology	**Normal**		
ESR	20	mm/1st hr	0-20 mm at the end of 1 hour

Dr. Sandeep Porwal
M. D. (Pathologist)

The Apollo Clinic

One World Impex Pvt. Ltd. "Madhuvan", M.G. Road, Kandivali-(West), Mumbai - 400 067.
Ph.: (022) - 2861 7416 / 17 /18 Fax : (022) - 2861 7419 Email : kandivali@theapolloclinic.com

Edited image after augmentation



DEPARTMENT OF LABORATORY MEDICINE

Haematology

The Apollo Clinic

Name : Mr Amitabh Das | Gender : Male | Age : 40 Years
UHID : ACM8.0000013831 | Bill No : ACS18326 | Lab No : 12
Ref. by : Dr. Dena Bank | Date : 26-Oct-2008 | Time : 01:08 PM

TEST	RESULTS		REFERENCE RANGE
CBC (Complete Blood Count)			
Hb (Haemoglobin)	14.9	gm/dl	13 - 17
RBC Count	4.74	Million/Cumm.	4.5 - 6
PCV / Haematocrit	43.8	%	40 - 50
MCV	92.41	fl	83 - 99
MCH	31.43	pg	26 - 34
MCHC	34.02	gm/dl	31.5 - 35
R.D.W.	12.2	%	11.5 - 14.5
Total Leukocyte Count	6000	/Cumm.	4000 - 11000
Differential Leukocyte Count			
Neutrophil	60	%	40 - 75
Lymphocyte	31	%	20 - 40
Eosinophil	06	%	1 - 6
Monocyte	03	%	2 - 8
Platelet Count	2.21	Lakhs/c.mm	1.5 - 4
RBC Morphology	Normocytic Normochromic		
WBC Morphology	**Normal**		
ESR	20	mm/1st hr	0-20 mm at the end of 1 hour

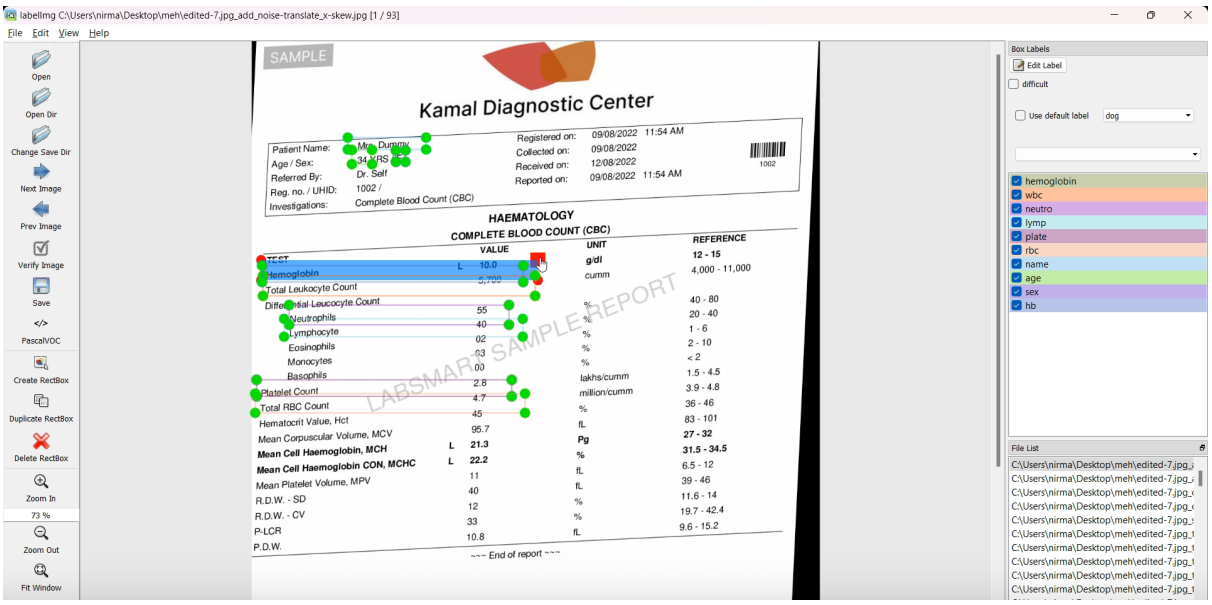
Dr. Sandeep Porwal
M. D. (Pathologist)

The Apollo Clinic

One World Impex Pvt. Ltd. "Madhuvan", M.G. Road, Kandivali-(West), Mumbai - 400 067.
Ph.: (022) - 2861 7416 / 17 /18 Fax : (022) - 2861 7419 Email : kandivali@theapolloclinic.com

8.2 Annotation

For annotation of the newly generated data, we used LabelImg tool, an open source tool that allows us to annotate images by drawing bounding boxes for the desired classes and export the annotations in xml, json or txt formats supported by various popular dataset norms/ object detection models like PASCAL VOC, yolo, coco, etc.



8.3 Model training

For model training we went with a 70-20-10 training, validation and testing split.

Initial train-test-val split code:

```
In [1]: import random
        import os


        def split_dataset(image_files, train_ratio=0.7, val_ratio=0.2, test_ratio=0.1):
            # shuffle the image files
            random.shuffle(image_files)

            # calculate the number of images for each set
            num_train = int(len(image_files) * train_ratio)
            num_val = int(len(image_files) * val_ratio)
            num_test = int(len(image_files) * test_ratio)

            # split the image files into the sets
            train_files = image_files[:num_train]
            val_files = image_files[num_train:num_train+num_val]
            test_files = image_files[num_train+num_val:]

            return train_files, val_files, test_files
```

```
In [3]: # Set folder path
        folder_path = 'meh'

        # Get list of image files in folder
        image_files = [os.path.join(folder_path, file) for file in os.listdir(folder_path) if file.endswith('.jpg')]

        # Shuffle image files
        random.shuffle(image_files)

        # Split image files into train and test sets
        train_split = 0.7
        num_train = int(len(image_files) * train_split)
        train_files = image_files[:num_train]
        test_files = image_files[num_train:]

        # Print number of files in train and test sets
        print('Number of training files:', len(train_files))
        print('Number of testing files:', len(test_files))

        Number of training files: 51
        Number of testing files: 22
```

ROBOFLOW 2.0 OBJECT DETECTION (FAST)

Roboflow facilitates the deployment of computer vision models to various platforms and frameworks, including cloud environments, edge devices, and web applications. It provides model export in formats compatible with popular frameworks, making it easy to integrate trained models into production pipelines. Roboflow also offers APIs and SDKs for seamless integration with existing software systems and workflows. Using APIs for model training.

```
from roboflow import Roboflow
import os
from PIL import Image
rf = Roboflow(api_key="DWr88H9uUstVlVubbYIQ")
project = rf.workspace().project("medi-9sqs7")
model = project.version(1).model

pred_json=model.predict('4.jpg', confidence=40, overlap=30).json()
# infer on a local image
print(pred_json)


# visualize your prediction
model.predict('4.jpg', confidence=40, overlap=30).save("prediction.jpg")

# infer on an image hosted elsewhere
# print(model.predict("URL_OF_YOUR_IMAGE", hosted=True, confidence=40, overlap=30).json())
```

```
loading Roboflow workspace...
loading Roboflow project...
{'predictions': [{'x': 424.0, 'y': 330.5, 'width': 228.0, 'height': 61.0, 'confidence': 0.853738307952880
```

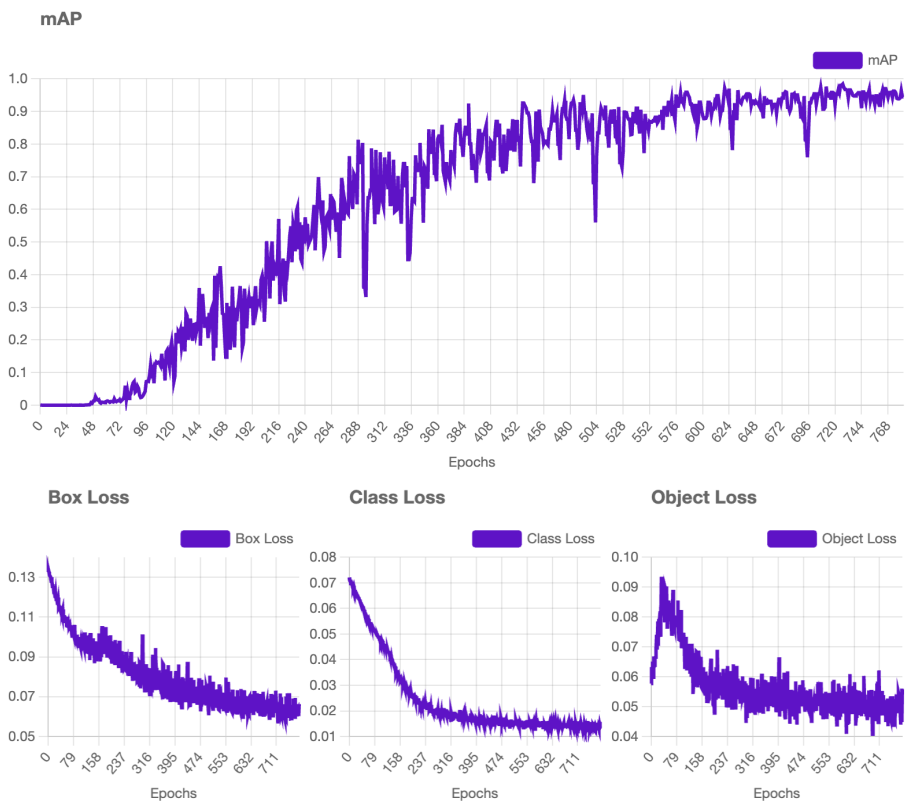


Fig 8.1. RoboFlow model performance

YOLOv8

Setting up data loaders to deliver batches of pictures and related annotations during training and loading the dataset and annotations. The model uses an input picture to generate bounding box predictions and class probabilities for the text sections during training rounds. Create a loss function that takes both categorization and localization aspects into account. Combining classification loss (such as cross-entropy loss) with bounding box regression loss (such as smooth L1 loss) is a popular choice. To minimise the specified loss function, train the YOLOv8 model using gradient descent-based optimisation techniques like Adam or SGD.

Evaluate the trained model on the validation set after training for a sufficient number of epochs. Examine performance indicators like precision, recall, and mean average precision (mAP) to judge how well the model can properly identify text areas.

```
[ ] model = YOLO(f'/yolov8n.pt')

Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt to /yolov8n.pt...
100%|[redacted]| 6.23M/6.23M [00:00<00:00, 67.8MB/s]
```

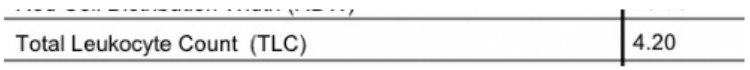
```
!yolo task=detect mode=train model=yolov8x.pt data='/content/datasets/medi-2/data.yaml' epochs=500 imgsz=600 plots=True

Ultralytics YOLOv8.0.20 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
yolo/engine/trainer: task=detect, mode=train, model=yolov8x.yaml, data=/content/datasets/medi-2/data.yaml, epochs=500, patience=50, batch
2023-05-12 05:28:00.055576: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-05-12 05:28:01.576057: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Overriding model.yaml nc=80 with nc=9

                   from  n    params  module                                       arguments
  0                  -1  1      2320  ultralytics.nn.modules.Conv                  [3, 80, 3, 2]
  1                  -1  1    115520  ultralytics.nn.modules.Conv                  [80, 160, 3, 2]
  2                  -1  3    436800  ultralytics.nn.modules.C2f                   [160, 160, 3, True]
  3                  -1  1    461440  ultralytics.nn.modules.Conv                  [160, 320, 3, 2]
  4                  -1  6   3281920  ultralytics.nn.modules.C2f                   [320, 320, 6, True]
  5                  -1  1   1844480  ultralytics.nn.modules.Conv                  [320, 640, 3, 2]
  6                  -1  6  13117440  ultralytics.nn.modules.C2f                   [640, 640, 6, True]
  7                  -1  1   3687680  ultralytics.nn.modules.Conv                  [640, 640, 3, 2]
  8                  -1  3   6969600  ultralytics.nn.modules.C2f                   [640, 640, 3, True]
  9                  -1  1   1025920  ultralytics.nn.modules.SPPF                  [640, 640, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.Concat                [1]
 12                  -1  3   7379200  ultralytics.nn.modules.C2f                   [1280, 640, 3]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.Concat                [1]
 15                  -1  3   1948800  ultralytics.nn.modules.C2f                   [960, 320, 3]
 16                  -1  1    922240  ultralytics.nn.modules.Conv                  [320, 320, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.Concat                [1]
 18                  -1  3   7174400  ultralytics.nn.modules.C2f                   [960, 640, 3]
 19                  -1  1   3687680  ultralytics.nn.modules.Conv                  [640, 640, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.Concat                [1]
 21                  -1  3   7379200  ultralytics.nn.modules.C2f                   [1280, 640, 3]


                   wbc          9          9      0.747          1      0.995      0.576
Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 219, best model saved as best.pt.
To update EarlyStopping(patience=50) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

269 epochs completed in 1.202 hours.
Optimizer stripped from runs/detect/train18/weights/last.pt, 136.7MB
Optimizer stripped from runs/detect/train18/weights/best.pt, 136.7MB

Validating runs/detect/train18/weights/best.pt...
Ultralytics YOLOv8.0.20 🚀 Python-3.10.11 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 68132235 parameters, 0 gradients, 257.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00,  1.22it/s]
                   all          9         81      0.921      0.887      0.965      0.594
                   age          9          9          1      0.856      0.995       0.51
                    hb          9          9      0.894          1      0.995      0.662
                  lymp          9          9      0.847          1      0.984      0.596
                  name          9          9      0.946          1      0.995       0.59
                neutro          9          9       0.98          1      0.995      0.659
                 plate          9          9      0.833      0.554      0.806      0.572
                   rbc          9          9      0.943      0.889      0.984      0.643
                   sex          9          9          1      0.679      0.984      0.487
                   wbc          9          9      0.848          1      0.951      0.629
Speed: 0.2ms pre-process, 22.8ms inference, 0.0ms loss, 1.1ms post-process per image
Results saved to runs/detect/train18
```

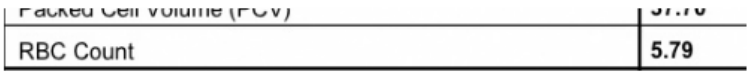
8.4 Image processing and text/ value extraction

Based on the results from the object detection models, we slice the original image according to the respective detected classes and their x, y, w, h. Through thorough testing, we decided to extend the cropped images by 50 pixels which resulted in better accuracy from the OCR module.

```
# load the image
img = cv2.imread('4.jpg')
im=Image.open('4.jpg')
# Initialize an empty dataframe to store the results
df = pd.DataFrame(columns=['name', 'text'])

# Process each prediction from RCNN
for pred in pred_json['predictions']:
    # Crop the image based on the object coordinates
    x, y, w, h = float(pred['x']), float(pred['y']), float(pred['width']), float(pred['height'])
    #print(x,y,w,h)
    #cropped_img = img[y:h+y, x:w+x]
#     img_res = im.crop((x, y, x+w, y+h))
#     img_res.show()
    roi_x = int(x - w / 2)
    roi_y = int(y - h / 2)
    roi_width = int(w)+50
    roi_height = int(h)
    cropped_img = img[roi_y:roi_y+roi_height, roi_x:roi_x+roi_width]
    plt.imshow(cropped_img)
    plt.axis("off")
    plt.show()
```

Example of cropped images:





Utilising an OCR model, text and value extraction from photographs or scanned documents includes automatically identifying and extracting pertinent textual information, such as words, phrases, or numeric values, to facilitate effective data retrieval and analysis.

Comparing EasyOCR and Tesseract in various scenarios, we chose to go with EasyOCR because of higher accuracy, cleaner output and faster speed.

```
[ ] import cv2
    import pytesseract
    import pandas as pd
    from PIL import Image
    import re
    from matplotlib import pyplot as plt


    # function to extract numeric information from a string
    def extract_numeric_info(text):
        return re.sub(r'[^\d.]+', '', text)

    # Function to check if a string is numeric
    def is_numeric(value):
        try:
            float(value)
            return True
        except ValueError:
            return False

    # initialize tesseract OCR with 'digits' as the allowed characters
    pytesseract.pytesseract.tesseract_cmd = r'/usr/bin/tesseract'
    tess_config = r'--psm 7'  # Set OCR mode to treat the image as a single line of text


    # load the image
    img = cv2.imread('4.jpg')
    im=Image.open('4.jpg')
    # Initialize an empty dataframe to store the results
    df = pd.DataFrame(columns=['name', 'text'])

    # Process each prediction from RCNN
    for pred in pred_json['predictions']:
        # Crop the image based on the object coordinates
```

```
#     img_res.show()
    roi_x = int(x - w / 2)
    roi_y = int(y - h / 2)
    roi_width = int(w)+50
    roi_height = int(h)
    cropped_img = img[roi_y:roi_y+roi_height, roi_x:roi_x+roi_width]


    #cv2.imshow('ImageWindow', cropped_img)
    #cv2.waitKey(0)


    # Perform OCR using Tesseract on the cropped image
    if pred['class'] in ['name', 'sex']:
        text = pytesseract.image_to_string(cropped_img, config=tess_config)
        output = reader.readtext(cropped_img,detail=0)
        text = text.replace("\n", "")
    else:
        text = pytesseract.image_to_string(cropped_img)
        output = reader.readtext(cropped_img,detail=0)
        text = text.replace("\n", "")

    # Extract numeric information for classes other than 'name' and 'sex'
    if pred['class'] not in ['name', 'sex','age']:
        #text = extract_numeric_info(text)
        numeric_values = [value for value in output if is_numeric(value)]
        text = ' '.join(numeric_values)
    if pred['class'] in ['age']:
        print(text)
        numeric_values = [extract_numeric_info(value) for value in output]
        text = ' '.join(numeric_values)
```

```
    # Add the name and extracted text to the dataframe
    df = df.append({'name': pred['class'], 'text': text}, ignore_index=True)
    print(output)


print(df)
```

After extracting the data from the OCR output, we store it in the form of a structured pandas dataframe

```
     name              text
0     wbc              4.20
1    name  Mr. NEERAJ OJHA
2     rbc              5.79
3      hb             12.00
4    lymp             50.30
5   plate             245.0
6     age                27
7     sex              Male
8  neutro             39.60
```

8.5 Report generation

1. The output from our OCR model was passed into a Dataframe structured as follows. This represented a single 'item' to be processed.

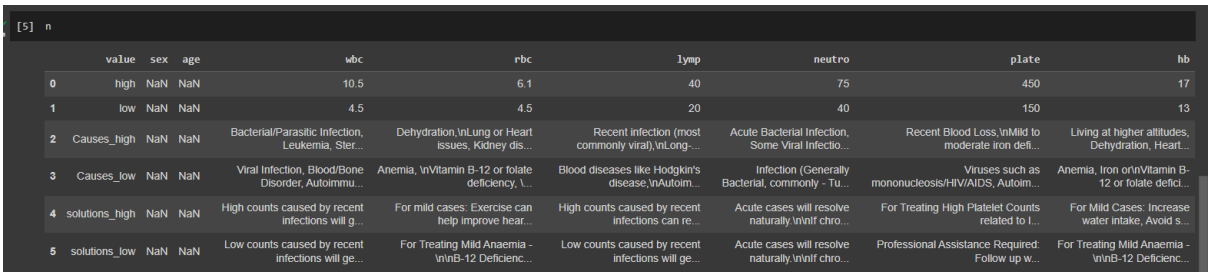
```
[4] item
```

	name	sex	age	wbc	rbc	lymp	neutro	plate	hb
0	Mr. Neeraj Ohja	Male	27	4.2	5.79	50.3	39.6	245.0	12

2. Normal range table was pulled based on the item's age group. HB and RBC values were updated by gender for adult and adolescent age groups.

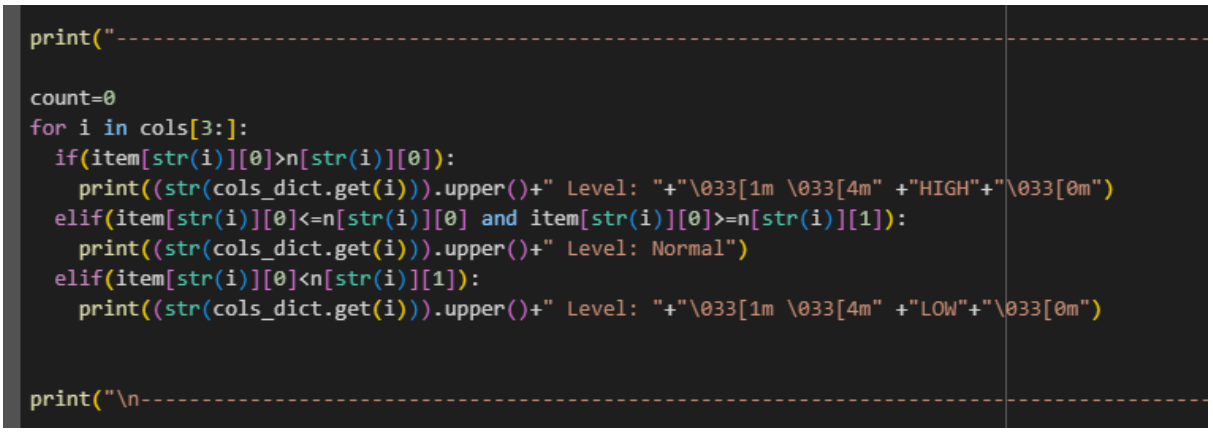
3. The item's values were calculated as either HIGH, LOW or Normal based on the normal-range table.

   The general normal range table for 'Adults' is shown below:



[5] n

	value	sex	age	wbc	rbc	lymp	neutro	plate	hb
0	high	NaN	NaN	10.5	6.1	40	75	450	17
1	low	NaN	NaN	4.5	4.5	20	40	150	13
2	Causes_high	NaN	NaN	Bacterial/Parasitic Infection, Leukemia, Ster...	Dehydration,\nLung or Heart issues, Kidney dis...	Recent infection (most commonly viral),\nLong-...	Acute Bacterial Infection, Some Viral Infectio...	Recent Blood Loss,\nMild to moderate iron defi...	Living at higher altitudes, Dehydration, Heart...
3	Causes_low	NaN	NaN	Viral Infection, Blood/Bone Disorder, Autoimmu...	Anemia, \nVitamin B-12 or folate deficiency, \...	Blood diseases like Hodgkin's disease,\nAutoim...	Infection (Generally Bacterial, commonly - Tu...	Viruses such as mononucleosis/HIV/AIDS, Autoim...	Anemia, Iron or\nVitamin B-12 or folate defici...
4	solutions_high	NaN	NaN	High counts caused by recent infections will g...	For mild cases: Exercise can help improve hear...	High counts caused by recent infections can re...	Acute cases will resolve naturally.\n\nIf chro...	For Treating High Platelet Counts related to I...	For Mild Cases: Increase water intake, Avoid s...
5	solutions_low	NaN	NaN	Low counts caused by recent infections will ge...	For Treating Mild Anaemia - \n\nB-12 Deficienc...	Low counts caused by recent infections will ge...	Acute cases will resolve naturally.\n\nIf chro...	Professional Assistance Required: Follow up w...	For Treating Mild Anaemia - \n\nB-12 Deficienc...



```
print("----------------------------------------------------------------------------------------------------

count=0
for i in cols[3:]:
  if(item[str(i)][0]>n[str(i)][0]):
    print((str(cols_dict.get(i))).upper()+" Level: "+"\033[1m \033[4m" +"HIGH"+"\033[0m")
  elif(item[str(i)][0]<=n[str(i)][0] and item[str(i)][0]>=n[str(i)][1]):
    print((str(cols_dict.get(i))).upper()+" Level: Normal")
  elif(item[str(i)][0]<n[str(i)][1]):
    print((str(cols_dict.get(i))).upper()+" Level: "+"\033[1m \033[4m" +"LOW"+"\033[0m")


print("\n----------------------------------------------------------------------------------------------------
```

4. Code for generating the detailed report for abnormal values:

```
count=1
for i in cols[3:]:
  if(item[str(i)][0]>n[str(i)][0]):
    print(str(count)+". ",end="")
    count=count+1
    perc=str(round((   (item[str(i)][0]-n[str(i)][0])  /   n[str(i)][0]  )*100,2))
    print(str(cols_dict.get(i)).upper()+" Level:\n - "+perc+" % HIGHER than normal")
    if(float(perc)<=10):
      print(" - MILD deviation from normal range")
    elif(float(perc)>10):
      print(" - !! Significant deviation from normal range")
    print("\033[1m \033[4m" +"\nPotential Causes: (Common causes are listed first)\n"+"\033[0m\n"+n[str(i)][2]+"\n")
    print("\033[1m \033[4m" +"\nSuggestions:\n"+"\033[0m\n"+n[str(i)][4]+"\n----------
  elif(item[str(i)][0]<n[str(i)][1]):
    print(str(count)+".",end="")
    count=count+1
    perc=str(round((   (n[str(i)][1]-item[str(i)][0])  /   n[str(i)][1]  )*100,2))
    print(str(cols_dict.get(i)).upper()+" Level:\n - "+perc+" % LOWER than normal")
    if(float(perc)<=10):
      print(" - MILD deviation from normal range")
    elif(float(perc)>10):
      print(" - !! Significant deviation from normal range")
    print("\033[1m \033[4m" +"\nPotential Causes: (Common causes are listed first)\n"+"\033[0m\n"+n[str(i)][3]+"\n")
    print("\033[1m \033[4m" +"\nSuggestions:\n"+"\033[0m\n"+n[str(i)][5]+"\n----------
```

5. The full report is generated and looks as follows:

```
-----------------------------------------------------------------Generated Report-----------------------------------------------------------------


-------------------------------------------------------------------------------------------------------------------------------------------------------
GENERAL INFORMATION
-------------------------------------------------------------------------------------------------------------------------------------------------------

Name: Mr. Neeraj Ohja
Sex: Male
Age: 27
Age Group: Adult (18-50 years old)

-------------------------------------------------------------------------------------------------------------------------------------------------------
ALL RESULTS
-------------------------------------------------------------------------------------------------------------------------------------------------------

WBC Level:  LOW
RBC Level: Normal
LYMPHOCYTE Level:  HIGH
NEUTROPHIL Level:  LOW
PLATELET Level: Normal
HAEMOGLOBIN Level:  LOW

-------------------------------------------------------------------------------------------------------------------------------------------------------
DETAILED REPORT OF ABNORMAL RESULTS
-------------------------------------------------------------------------------------------------------------------------------------------------------

1.WBC Level:
 - 6.67 % LOWER than normal
 - MILD deviation from normal range

Potential Causes: (Common causes are listed first)

Viral Infection, Blood/Bone Disorder, Autoimmune Condition,  Chemo/Radiation Therapy, Medication Side Effect


Suggestions:

Low counts caused by recent infections will generally resolve naturally.

Otherwise professional assistance may be required:  Follow up with your healthcare provider for prescriptions/further testing
-------------------------------------------------------------------------------------------------------------------------------------------------------

2. LYMPHOCYTE Level:
 - 25.75 % HIGHER than normal
 - !! Significant deviation from normal range

Potential Causes: (Common causes are listed first)

Recent infection (most commonly viral),
Long-term inflammation (such as rheumatoid arthritis),
Medication Side Effect, Trauma,
```

```
Suggestions:

High counts caused by recent infections can resolve naturally.

For more severe symptoms contact a medical professional.
--------------------------------------------------------------------------------------------------------------------------------------------------------------

3.NEUTROPHIL Level:
 - 1.0 % LOWER than normal
 - MILD deviation from normal range

Potential Causes: (Common causes are listed first)

Infection (Generally Bacterial, commonly -  Tuberculosis, Dengue, HIV), Inflammation, Chemotherapy, Blood Cancer (rare), Leukemia (rare)


Suggestions:

Acute cases will resolve naturally.

If chronic/severe, professional assistance is required:  Follow up with your healthcare provider for prescriptions/further testing
--------------------------------------------------------------------------------------------------------------------------------------------------------------

4.HAEMOGLOBIN Level:
 - 7.69 % LOWER than normal
 - MILD deviation from normal range

Potential Causes: (Common causes are listed first)

Anemia, Iron or
Vitamin B-12 or folate deficiency,
Inherited Diseases (thalassemia),
Bleeding


Suggestions:

For Treating Mild Anaemia -

B-12 Deficiency: Meat, fish, eggs, dairy products, yeast extract

Folate Deficiency: green vegetables such as broccoli, Brussels sprouts and peas.

Iron Deficiency:Spinach, Meat, eggs, Fortified Bread, Beans, Dal

For Supplemental Injections/Tablets, contact your healthcare provider.

More serious symptoms require professional consultation
--------------------------------------------------------------------------------------------------------------------------------------------------------------
```

8.6 GUI

Backend: app.py consists of the code from all the modules above, with additional functions to create pdf and hande the web components.

```
app.py 9+ ●   index.html   result.html
app.py > ...
app = Flask(__name__)
def image_proc(image_path,pred_json): ···
# Function to pass the image to the Roboflow API
def pass_to_roboflow(image_path): ···


def pass_to_yolo(image_path): ···

def extract_numeric_info(text): ···

# Function to check if a string is numeric
def is_numeric(value): ···

# Function for report generation
def generate_report(df): ···

# Function for PDF creation
def create_pdf(report): ···
```

app.py 9+ ● index.html result.html

app.py > ...

```python
@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        # Get the uploaded image file from the request
        image_file = request.files['image']

        # Save the image file to a temporary location
        image_path = 'tmp/uploaded_image.jpg'
        image_file.save(image_path)

        # Pass the image to the Roboflow API
        output = pass_to_roboflow(image_path)

        output_yolo= pass_to_yolo(image_path)

        ocr_output=image_proc(image_path,output)

        # Generate report
        report = generate_report(ocr_output)

        # Create a PDF
        pdf_path = create_pdf(report)

        # Render the result page with the PDF path
        return render_template('result.html', pdf_path=pdf_path, image_path=image_path)

    return render_template('index.html')
```

```python
@app.route('/image/<path:filename>')
def get_image(filename):
    return send_file('/tmp/' + filename, mimetype='image/jpg')

@app.route('/download_pdf')
def download_pdf():
    pdf_path = request.args.get('pdf_path')
    return send_file(pdf_path, as_attachment=True)

if __name__ == '__main__':...
```

Frontend: We have 2 pages for the above application, index.html (Homepage) and result.html to show the output.

app.py 9+ | index.html | result.html

templates > index.html > html > head > style > body

```html
        }
        #logo {
            width: 500px;
            height: auto;
            display: block;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <img src="{{url_for('static', filename='logo.png')}}" alt="Logo" id="logo">
    <h1>Hematology report data extraction and diagnosis generation</h1>
    <form method="POST" enctype="multipart/form-data" action="/">
        <input type="file" name="image" accept="image/*" required>
        <br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

app.py 9+ | index.html | result.html

templates > result.html > html > body

```html
</head>
<body>
    <h1>Result</h1>
    <!--<div>
        <h2>Sliced Images</h2>
        {% for sliced_image in sliced_images %}
            <img src="{{ sliced_image }}">
        {% endfor %}
    </div>-->
    <div>
        <h2>Comparison</h2>
        <div style="display: flex; justify-content: center;">
            <div style="flex: 1; padding: 10px;">
                <h3>Output from Roboflow Object detection model</h3>
                <img src= "{{url_for('static', filename='prediction.jpg')}}">
            </div>
            <div style="flex: 1; padding: 10px;">
                <h3>Output from YOLOv8x</h3>
                <img src="{{url_for('static', filename='predict/uploaded_image.jpg')}}">
            </div>
        </div>
    </div>
    <div>
        <h2>Download PDF Report</h2>
        <a href="{{ url_for('download_pdf', pdf_path=pdf_path) }}">Download</a>
    </div>
</body>
</html>
```

Chapter 9
Testing

Module 1: Object detection model comparison:

	mAP (%)	Precision (%)	Recall (%)	Training time	Epochs
ROBOFLOW 2.0 OBJECT DETECTION (FAST)	98.4	90.4	94.7	1Hr	790
YOLOv8x	96.5	92.1	88.7	1.202 Hr	269

Table 9.1. Object detection model comparison

YOLOv8 result

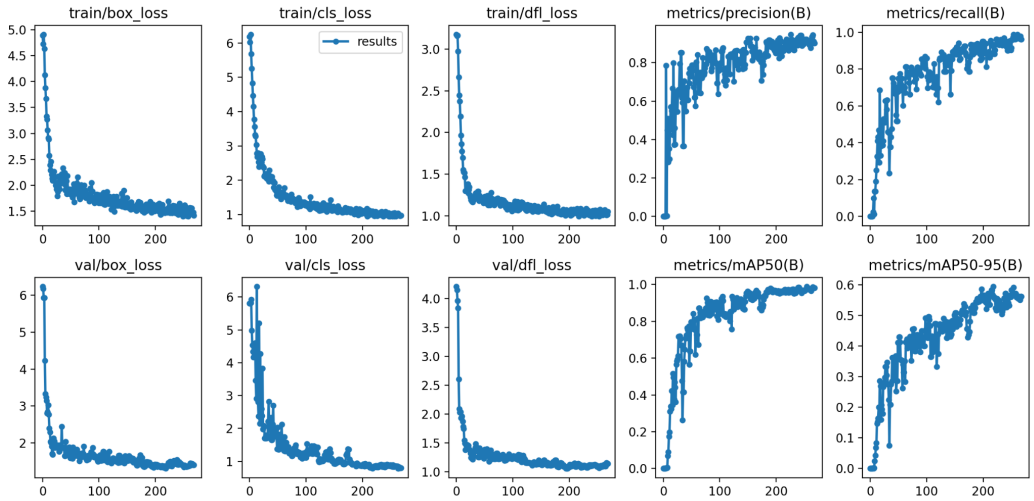


Fig 9.1. YOLOv8x result

ROBOFLOW 2.0 OBJECT DETECTION (FAST) result

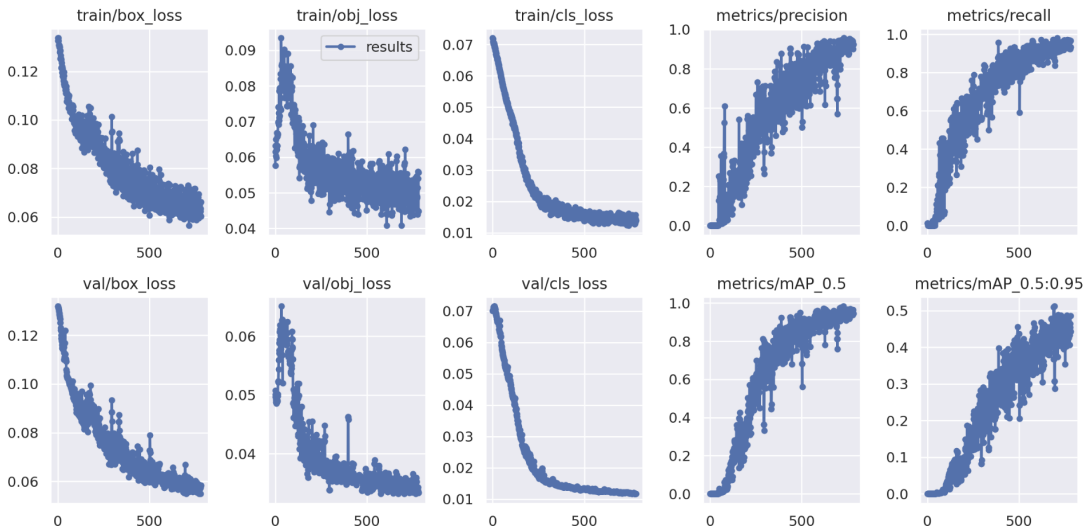
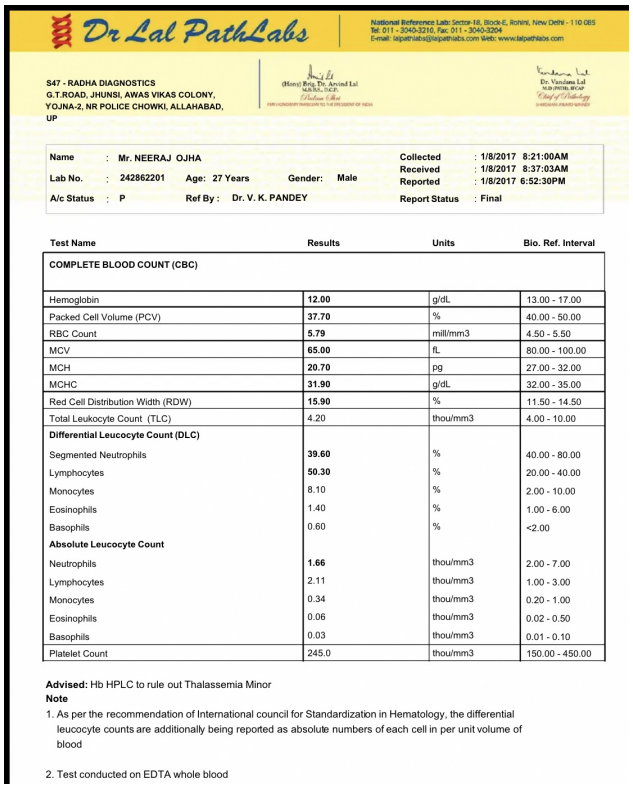


Fig 9.2. Roboflow 2.0 Object detection (fast) result

Input for testing object detection models:



Dr Lal PathLabs

National Reference Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3040-3210, Fax: 011 - 3040-3204
E-mail: lalpathlabs@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

(Hony) Brig. Dr. Arvind Lal
MBBS, DGO
Pathologist

Dr. Vandana Lal
MD(PATH), FCAP
Chief of Pathology

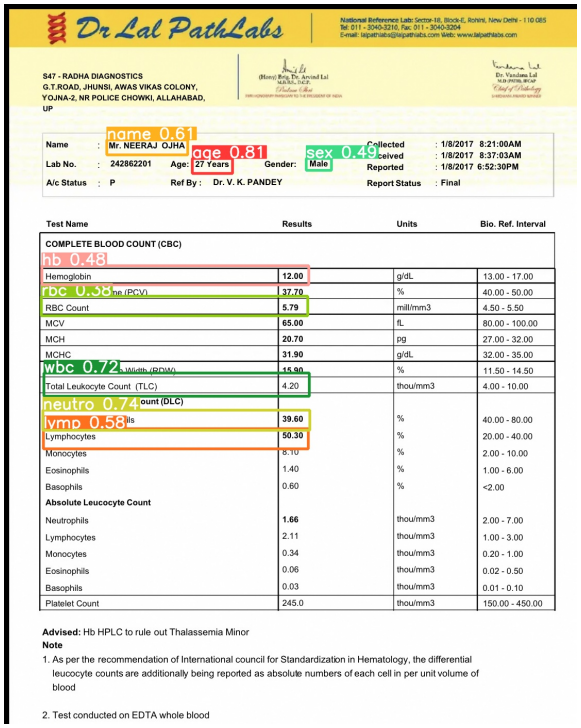
Name	: Mr. NEERAJ OJHA	Collected	: 1/8/2017 8:21:00AM
Lab No.	: 242862201 Age: 27 Years Gender: Male	Received	: 1/8/2017 8:37:03AM
		Reported	: 1/8/2017 6:52:30PM
A/c Status	: P Ref By : Dr. V. K. PANDEY	Report Status	: Final

Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
Hemoglobin	**12.00**	g/dL	13.00 - 17.00
Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	**39.60**	%	40.00 - 80.00
Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
Platelet Count	245.0	thou/mm3	150.00 - 450.00

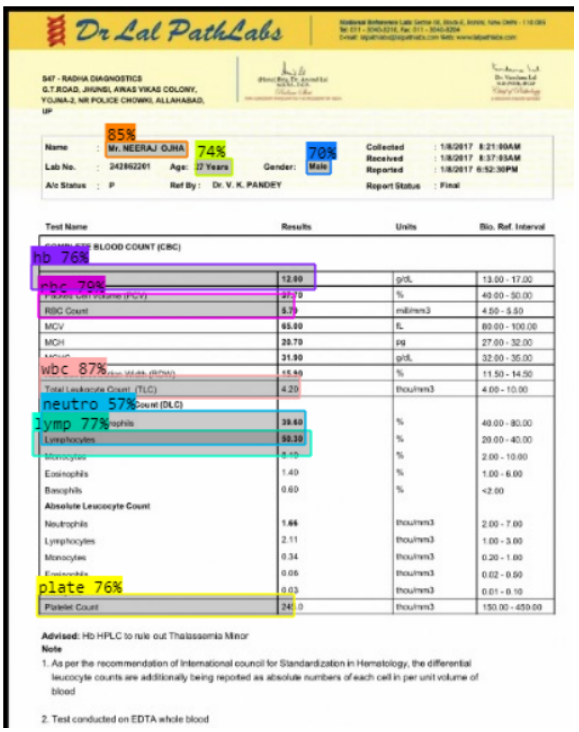
Advised: Hb HPLC to rule out Thalassemia Minor
Note
1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

2. Test conducted on EDTA whole blood

YOLOv8x output



ROBOFLOW 2.0 OBJECT DETECTION (FAST) output



Inference: Comparing both the models, going through the output and the evaluation metrics, we can notice that Roboflow 2.0 object detection model performs better.. In the above comparison of the predicted objects, we can notice the higher confidence and accuracy in the Roboflow 2.0 model compared to the YOLOv8x.
Conclusion: Based on the results obtained by training the roboflow 2.0 object detection (fast) and YOLOv8x model, we chose to go with roboflow 2.0 due to the following reasons:

- Better mAP and recall score
- Comparable precision score
- Faster
- Direct API provision for easier deployment

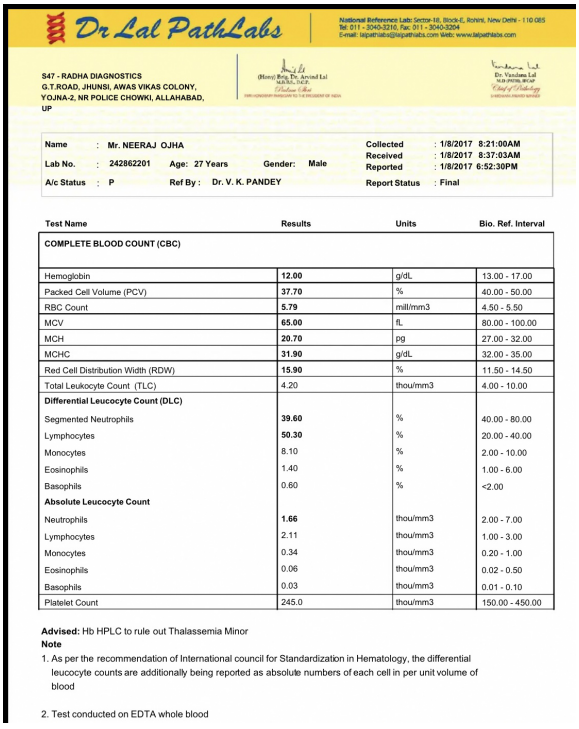
Module 2: OCR comparison:

Image chosen for testing:

```
import os
from PIL import Image
import easyocr
import cv2
import pytesseract
import pandas as pd
import re
reader = easyocr.Reader(['en'])
```

```
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take
Progress: |                                        | 100.0% Complete
Progress: |                                        | 100.0% Complete
```

```
[16] original_image=Image.open('a.jpg')
     display(original_image)
```



Dr Lal PathLabs

National Reference Lab: Sector-18, Block-E, Rohini, New Delhi - 110 085
Tel: 011 - 3045-3216, Fax: 011 - 3045-3204
E-mail: lalpathlabs@lalpathlabs.com Web: www.lalpathlabs.com

S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY,
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD,
UP

(Hony) Brig. Dr. Arvind Lal
M.B.B.S., M.D.
Pathology Editor

Dr. Vandana Lal
M.D. (PATH), IBCAP
Head of Pathology

Name	: Mr. NEERAJ OJHA		Collected : 1/8/2017 8:21:00AM
Lab No.	: 242862201 Age: 27 Years	Gender: Male	Received : 1/8/2017 8:37:03AM
			Reported : 1/8/2017 6:52:30PM
A/c Status	: P Ref By : Dr. V. K. PANDEY		Report Status : Final

Test Name	Results	Units	Bio. Ref. Interval
COMPLETE BLOOD COUNT (CBC)			
Hemoglobin	**12.00**	g/dL	13.00 - 17.00
Packed Cell Volume (PCV)	**37.70**	%	40.00 - 50.00
RBC Count	**5.79**	mill/mm3	4.50 - 5.50
MCV	**65.00**	fL	80.00 - 100.00
MCH	**20.70**	pg	27.00 - 32.00
MCHC	**31.90**	g/dL	32.00 - 35.00
Red Cell Distribution Width (RDW)	**15.90**	%	11.50 - 14.50
Total Leukocyte Count (TLC)	4.20	thou/mm3	4.00 - 10.00
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	**39.60**	%	40.00 - 80.00
Lymphocytes	**50.30**	%	20.00 - 40.00
Monocytes	8.10	%	2.00 - 10.00
Eosinophils	1.40	%	1.00 - 6.00
Basophils	0.60	%	<2.00
Absolute Leucocyte Count			
Neutrophils	**1.66**	thou/mm3	2.00 - 7.00
Lymphocytes	2.11	thou/mm3	1.00 - 3.00
Monocytes	0.34	thou/mm3	0.20 - 1.00
Eosinophils	0.06	thou/mm3	0.02 - 0.50
Basophils	0.03	thou/mm3	0.01 - 0.10
Platelet Count	245.0	thou/mm3	150.00 - 450.00

Advised: Hb HPLC to rule out Thalassemia Minor
Note
1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood
2. Test conducted on EDTA whole blood

EasyOCR output

```
[13] ocr_result = reader.readtext('a.jpg',detail=0)
```

```
ocr_result
```

```
['Natonal Reterence Lob: Sector-[8. Block E Roh
ni New Dcthl:',
 'Ki0 085',
 '8 Dr Lal PathLals',
 'Exai: 121pot-32toapetrOansc',
 '3m032cbwwlaipachlabs com',
 'LL',
 '547',
 'RADHA DIAGNOSTICS',
 '(Honv)',
 "'Rrkea",
 'Dc. Anin { Lal',
```

Tesseract output

```
pytesseract.pytesseract.tesseract_cmd = r'/usr/bin/tesseract'
```

```
img=cv2.imread('a.jpg',1)
print(pytesseract.image_to_string(img))
```

```
S47 - RADHA DIAGNOSTICS
G.T.ROAD, JHUNSI, AWAS VIKAS COLONY, Sedum hee
YOJNA-2, NR POLICE CHOWKI, ALLAHABAD, Sa
aoamNesa
uP
Name Mr. NEERAJ OJHA
Lab No. 242862201 Age: 27 Years Gender: Male
Alc Status : P RefBy: Dr.V.K. PANDEY
Test Name Results

COMPLETE BLOOD COUNT (CBC)
```

'Dr Vindan_ Lall',	Hemoglobin
'MRnIn',	Packed Cell Volume (PCV)
'DCr',	RBC Count
'G.TROAD, JHUNSI; AWAS VIKAS COLONY_',	
'Whooramdr',	(Hony) Brig. De, Arvind Lal
'TATot Oer',	Mis ber.
'{di',	
'YOJNA-2, NR POLICE CHOWKI, ALLAHABAD_',	Collected
'UP',	Received
'Name',	Reported
'Mr: NEERAJ OJHA',	
'Collected',	Report Status
'1/8/2017 8.21.OOAM',	
'Received',	Units
'1/8/2017 8.37:03AM',	
'Lab No_',	gldL
'242862201',	%
'Age:',	
'27 Years',	mill/mm3
'Gender:',	
'Male',	Vandana Lat
'Reported',	Dr. Vandana Lal
'1/8/2017 6.52:30PM',	Nip ear
'Alc Status',	Chiff Dibetgy
'Ref By',	
'Dr. V.K PANDEY',	: 1/8/2017 8:21:00AM
'Report Status',	18/2017 8:37:03AM
'Final',	: 1/8/2017 6:52:30PM
'Test Name',	
'Results',	
'Units',	
'Bio. Ref: Interval',	: Final
'COMPLETE BLOOD COUNT (CBC)',	
'Hemoglobin',	Bio. Ref. Interval
'12.00',	
'gldL',	13.00 - 17.00
'13.00 - 17.00',	40.00 - 50.00
'Packed Cell Volume (PCV)',	4.50 - 5.50
'37.70',	80.00 - 100.00
'40.00',	
'50.00',	
'RBC Count',	
'5.79',	27.00 - 32.00
'milllmm3',	
'4.50 - 5.50',	
'MCV',	
'65.00',	Red Cell Distribution Width (RDW)
'80.00',	Total Leukocyte Count (TLC)
'100.00',	Differential Leucocyte Count (DLC)
'MCH',	Segmented Neutrophils
'20.70',	Lymphocytes
'pg',	
'27.00',	Monocytes
'32.00',	
'MCHC',	Eosinophils
'31.90',	
'gldL',	Basophils
'32.00',	
'35.00',	Absolute Leucocyte Count
'Red Cell Distribution Width (RDW)',	
'15.90',	Neutrophils
'11.50 _',	
'14.50',	Lymphocytes
'Total Leukocyte Count (TLC)',	
'4.20',	Monocytes
'thoulmm3',	Eosinophils
'4.00 -',	
'10.00',	Basophils
'Differential Leucocyte Count (DLC)',	
'Segmented Neutrophils',	%
'39.60',	
'40.00 - 80.00',	thou/mm3
'Lymphocytes',	
'50.30',	%
'20.00 - 40.00',	%
'Monocytes',	%
'8.10',	%
'2.00 - 10.00',	%
'Eosinophils',	
'1.40',	thou/mm3

'1.00 - 6.00', 'Basophils', '0.60', '<2.00', 'Absolute Leucocyte Count', 'Neutrophils', '1.66', 'thoulmm3', '2.00 - 7.00', 'Lymphocytes', '2.11', 'thoulmm3', '1.00 - 3.00', 'Monocytes', '0.34', 'thoulmm3', '0.20 - 1.00', 'Eosinophils', '0.06', 'thoulmm3', '0.02 - 0.50', 'Basophils', '0.03', 'thoulmm3', '0.01 - 0.10', 'Platelet Count', '245.0', 'thoulmm3', '150.00', '450.00', 'Advised: Hb HPLC to rule out Thalassemia Minor', 'Note', '1. As per the recommendation of International council for Standardization in Hematology, the differential', 'leucocyte counts', 'additionally', 'reported as absolute numbers ofeach cell in per unit volume of', 'blood', '2. Test conducted on EDTA whole blood', 'Vi ', '617td', 'J', 'being', 'are']	thou/mm3 thou/mm3 thou/mm3 thou/mm3 32.00 - 35.00 11.50 - 14.50 4.00 - 10.00 40.00 - 80.00 20.00 - 40.00 2.00 - 10.00 1.00 - 6.00 <2.00 2.00 - 7.00 1.00 - 3.00 0.20 - 1.00 0.02 - 0.50 0.01 - 0.10 Platelet Count thou/mm3 150.00 - 450.00 Advised: Hb HPLC to rule out Thalassemia Minor Note 1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood 2. Test conducted on EDTA whole blood

Inference: Going through the direct output by the OCR's, we can notice a accurate and cleaner output from the EasyOCR compared to the Tesseract.

Conclusion: Based on the results obtained from direct comparison of EasyOCR and Tesseract, we chose to go with EasyOCR due to the following reasons:

- Higher accuracy
- Cleaner output
- Slightly faster

Chapter 10
Conclusion

Our research has demonstrated that utilising deep learning OCR models for blood report scanning may be a successful method of gathering pertinent information for illness prediction models. In the study on haematology report OCR models, the efficiency of several strategies for automated text extraction from haematology reports was investigated. Utilising its object identification capabilities, the YOLOv8 model proved to be a reliable and effective method for identifying text areas within the reports. The whole computer vision workflow was streamlined by the full platform offered by Roboflow for data storage, annotation, and model training. On the other hand, Tesseract OCR and EasyOCR OCR models showed their language support and convenience of use, allowing accurate text extraction from the reports. The best OCR model to use will rely on a variety of criteria, including project-specific objectives, resource availability, and performance requirements. Each technique has its own advantages. Overall, by utilising OCR technology, this project improved the automation and effectiveness of haematology report processing.

Chapter 11
Future prospect

Future scope for this study includes exploring more advanced OCR models and algorithms, using more advanced data pre-processing techniques, and testing on a larger dataset with a wider variety of report formats. Additionally, incorporating more features and data such as patient demographics, medical history, and other laboratory results could help implement a better disease prediction model.

- Broader and more varied data set: The performance of the object detection and OCR models may be evaluated more thoroughly by expanding the dataset to include a wider and more varied collection of haematology papers.
- Multi-modal Information Integration: Including additional data with the extracted text, such as patient demographics, medical history, or other test findings, can provide the reports a more complete analysis. The creation of more complex illness prediction models may be made possible by the integration of different modalities, such as pictures or time-series data.
- Real-time Processing and Deployment: For applications that need quick results, real-time processing of haematology reports may be essential.
- Advanced OCR Models: Researching more sophisticated OCR models, such as those based on advanced deep learning (Transformer) architectures, may enhance the reliability and accuracy of the text extraction procedure.
- Better Data Pre-processing: Using cutting-edge data pre-processing methods including picture enhancement modification can increase the accuracy of OCR.

References

1. S. H. Park et al., “Establishment of Age- and Gender-Specific Reference Ranges for 36 Routine and 57 Cell Population Data Items in a New Automated Blood Cell Analyzer, Sysmex XN-2000,” Ann Lab Med, vol. 36, no. 3, pp. 244–249, May 2016, doi: 10.3343/alm.2016.36.3.244.
2. “Complete Blood Count (CBC) Test - Normal Ranges and details,” Aster Labs. https://www.asterlabs.in/resources/blogs/complete-blood-count-cbc-test-normal-ranges-and-details (accessed May 15, 2023).
3. K. K. Sehgal, D. Tina, U. Choksey, R. J. Dalal, and K. J. Shanaz, “Reference range evaluation of complete blood count parameters with emphasis on newer research parameters on the complete blood count analyzer Sysmex XE-2100,” Indian Journal of Pathology and Microbiology, vol. 56, no. 2, p. 120, Apr. 2013, doi: 10.4103/0377-4929.118698.
4. G. Nilsson, P. Hedberg, and J. Ohrvik, “White blood cell count in elderly is clinically useful in predicting long-term survival,” J Aging Res, vol. 2014, p. 475093, 2014, doi: 10.1155/2014/475093.
5. “LSUHSC School of Allied Health Professions.” https://alliedhealth.lsuhsc.edu/cl/default.aspx (accessed May 15, 2023).
6. L. Dean, “Table 1, Complete blood count,” 2005. https://www.ncbi.nlm.nih.gov/books/NBK2263/table/ch1.T1/ (accessed May 15, 2023).
7. “Complete Blood Count - Health Encyclopedia - University of Rochester Medical Center.” https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=167&contentid=complete_blood_count (accessed May 15, 2023).
8. “Hemoglobin: Normal, High, Low Levels, Age, Chart, Treatment & Symptoms,” MedicineNet. https://www.medicinenet.com/hemoglobin/article.htm (accessed May 15, 2023).
9. “Red Blood Cell Count - Health Encyclopedia - University of Rochester Medical Center.” https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=167&contentid=red_blood_cell_Count (accessed May 15, 2023).
10. J. PLUNCEVIC GLIGOROSKA, S. GONTAREV, B. DEJANOVA, L. TODOROVSKA, D. SHUKOVA STOJMANOVA, and S. MANCHEVSKA, “Red Blood Cell Variables in Children and Adolescents regarding the Age and Sex,” Iran J Public Health, vol. 48, no. 4, pp. 704–712, Apr. 2019, Accessed: May 15, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6500523/
11. “Blood differential test.” https://www.ucsfbenioffchildrens.org/Medical Tests/003657 (accessed May 15, 2023).
12. “Pediatric Reference Ranges.” https://www.healthcare.uiowa.edu/path_handbook/appendix/heme/pediatric_normals.html (accessed May 15, 2023).
13. W. Xue, Q. Li and Q. Xue, "Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach," in IEEE Access, vol. 8, pp. 407-416, 2020, doi: 10.1109/ACCESS.2019.2961964.
14. Q. Ye and D. Doermann, ‘‘Text detection and recognition in imagery: A survey,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 7, pp. 1480–1500, Jul. 2015.
15. Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, ‘‘Detecting text in natural image with connectionist text proposal network,’’ in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 56–72.
16. M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, ‘‘Textboxes: A fast text detector with a single deep neural network,’’ in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 4161–4167.
17. B. Shi, X. Bai, and C. Yao, ‘‘An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 11, pp. 2298–2304, Nov. 2017.
18. K. R. Pugh, W. E. Mencl, A. R. Jenner, L. Katz, S. J. Frost, J. R. Lee, S. E. Shaywitz, and B. A. Shaywitz, “Neurobiological studies of reading and reading disability,” J. Commun. Disord., vol. 34, no. 6, pp. 479– 492, 2001.
19. S. Mori, C. Y. Suen, and K. Yamamoto, “Historical Review of OCR Research and Development,” Proc. IEEE, vol. 80, no. 7, pp. 1029–1058, 1992.
20. A. Verma, S. Arora, and P. Verma, “OCR-OPTICAL CHARACTER RECOGNITION,” pp. 181–191.
21. C. Patel, A. Patel, and D. Patel, “Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study,” Int. J. Comput. Appl., vol. 55, no. 10, pp. 50–56, 2012.
22. R. Mithe, S. Indalkar, and N. Divekar, “Optical Character Recognition,” Int. J. Recent Technol. Eng., vol. 2, no. 1, pp. 72–75, 2013.

23. S. Rakshit and S. Basu, “Recognition of Handwritten Roman Script Using Tesseract Open source OCR Engine,” Natl. Conf. NAQC, pp. 141–145, 2010.

Base Papers

- W. Xue, Q. Li and Q. Xue, "Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach," in IEEE Access, vol. 8, pp. 407-416, 2020, doi: 10.1109/ACCESS.2019.2961964.

- W. A. Qader and M. M. Ameen, "Diagnosis of Diseases from Medical Check-up Test Reports Using OCR Technology with BoW and AdaBoost algorithms," 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019, pp. 205-210, doi: 10.1109/IEC47844.2019.8950605.

- Thompson, P., Batista-Navarro, R. T., Kontonatsios, G., Carter, J., Toon, E., McNaught, J., Timmermann, C., Worboys, M., & Ananiadou, S. (2016). Text Mining the History of Medicine. PloS one, 11(1), e0144717. https://doi.org/10.1371/journal.pone.0144717

Plagiarism check

turnitin

Similarity Report ID: oid:28480:35801762

PAPER NAME	AUTHOR
B17-BTECH_REPORT-2023 (1).pdf	pranav

WORD COUNT	CHARACTER COUNT
7526 Words	41709 Characters
PAGE COUNT	**FILE SIZE**
56 Pages	11.2MB
SUBMISSION DATE	**REPORT DATE**
May 20, 2023 3:01 PM GMT+5:30	May 20, 2023 3:02 PM GMT+5:30

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 9% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 7% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Manually excluded text blocks

PART - B

INDIVIDUAL CONTRIBUTION

Hematology report data extraction and diagnosis generation

Pranav Nirmal
1032192059
PB52

Data Augmentation

1. To implement data augmentation techniques such as rotation, skewing, scaling, and noise addition to improve the performance of the object detection module.

2. Scope:
   The generated dataset will contribute to model training.

3. Contribution:
   Augmentation of the 4 selected blood report formats, augmenting using OpenCV functions like skewing, color hue change, adding blur and noise, X-Y translation.

4. Module Dependencies:
   4.1. Frameworks:
      4.1.1. OpenCV
      4.1.2. Pillow

5. Module Testing Strategies:
   5.1. Manual testing of the dataset generated

6. Module Deployment:
   One time execution of the module to increase the size of the dataset.

Object detection module

1. Training a deep learning-based model to identify and localize relevant regions in hematology report images, such as data sections like name, age, sex, hemoglobin, wbc, platelets, etc. This approach of object detection improves the accuracy of the OCR module and helps extract clean data.

2. Scope:
   This module aims at detecting the coordinates of the data sections (as mentioned above) accurately for the next modules: image processing and OCR.

3. Contribution:
   Complete training, testing and validation of ROBOFLOW 2.0 OBJECT DETECTION (FAST) and YOLOv8x models.
   Comparison of both the models under different circumstances.

4. Software Requirements:
    - 4.1. Jupyter NB
    - 4.2. Python
    - 4.3. Anaconda
    - 4.4. Text editor
    - 4.5. Web browser (internet connection)
    - 4.6. Google colab
    - 4.7. Cloud storage (Google drive)

5. Hardware Requirements
    - 5.1. Processor: Intel core i5, 8th gen and above
    - 5.2. RAM: 8GB and above
    - 5.3. Storage: 80GB and above
    - 5.4. GPU: Nvidia T4 (server grade GPU), 15.6 GB usable VRAM
    - 5.5. Display: CRT/ LCD/ LED
    - 5.6. Camera: pixel resolution of at least 5 MP
    - 5.7. Mouse: Compatible mouse
    - 5.8. Keyboard: normal or multimedia

6. Module Dependencies:
    - 6.1. Frameworks:
        - 6.1.1. TensorFlow
        - 6.1.2. PyTorch
        - 6.1.3. OpenCV

    - 6.2. Object Detection Libraries:
        - 6.2.1. ROBOFLOW 2.0 OBJECT DETECTION (FAST) API
        - 6.2.2. YOLO (You Only Look Once)

    - 6.3. Additional Computer Vision Libraries:
        - 6.3.1. Pillow (Python Imaging Library)
        - 6.3.2. NumPy
        - 6.3.3. SciPy

    - 6.4. Miscellaneous Dependencies:
        - 6.4.1. Pandas (for data manipulation and analysis)
        - 6.4.2. Matplotlib (for data visualization)

7. Module Implementation:
    - 7.1. Training the models with the annotated dataset of the blood reports
    - 7.2. For YOLOv8, selecting the ‘X’ variant (after comparison with ‘S’ and ‘L’ version) which has the highest accuracy but takes longer to train and run.

7.3. For the Roboflow object detection model, we went with the 'Fast' variant which offered faster training and runtime.

7.4. Hyperparameters for the YOLOv8x model are:

7.4.1. Epochs: 500

7.4.2. Image size: 600px

7.5. Hyperparameters for the Roboflow object detection model are:

7.5.1. Epochs: 790

7.5.2. Image size: 500px

8. Module Testing Strategies:

	mAP (%)	Precision (%)	Recall (%)	Training time	Epochs
ROBOFLOW 2.0 OBJECT DETECTION (FAST)	98.4	90.4	94.7	1Hr	790
YOLOv8x	96.5	92.1	88.7	1.202 Hr	269

8.1. Testing with different formats of report:

8.1.1. Roboflow 2.0 performs better in terms of accuracy and speed.

9. Module Deployment:

Ease of deployment:

Roboflow 2.0 has been trained online, the platform offers direct API calls to the model and provides a clean json output with coordinates, confidence and class name.

Image processing

1. To apply various image processing techniques like denoising, resizing, color correction, and feature extraction to enhance the quality of images and extract relevant information.

2. Scope: This module aims at slicing the image based on the output from the object detection module i.e. the coordinates of the classes predicted.

3. Contribution:

Development of the function for slicing the image and providing it as an input for the OCR module.

4. Software Requirements:

4.1. Jupyter NB

4.2. Python

4.3. Anaconda

4.4. Text editor

4.5. Web browser (internet connection)

4.6. Google colab

4.7. Cloud storage (Google drive)

5. Hardware Requirements
   5.1. Processor: Intel core i5, 8th gen and above
   5.2. RAM: 4GB and above
   5.3. Storage: 80GB and above
   5.4. Display: CRT/ LCD/ LED
   5.5. Camera: pixel resolution of at least 5 MP
   5.6. Mouse: Compatible mouse
   5.7. Keyboard: normal or multimedia

6. Module Dependencies:
   6.1. Frameworks:
      6.1.1. OpenCV
      6.1.2. Pillow

7. Module Implementation:
   Based on the results from the object detection models, we slice the original image according to the respective detected classes and their x, y, w, h. Through thorough testing, we decided to extend the cropped images by 50 pixels which resulted in better accuracy from the OCR module.

8. Module Testing Strategies:
   8.1. Manual output cropped image evaluation.
   8.2. Evaluation on the basis of OCR module accuracy.

PART - B

INDIVIDUAL CONTRIBUTION

Hematology Report Data Extraction and Inference Generation

Problem Statement: Hematology report data extraction and inference generation

Name of the Student: Preetika Sastry

Module Titles:

- **Data Collection for Object Detection Module**
- **Report Generation Module**
- **GUI Module**

1. Data Collection for Object Detection Module

Project's Module Objectives

- To build a dataset of blood reports for our object detection module
- Get an understanding of the type of data we are dealing with

Project's Module Scope

- The dataset would be used to train our object detection module for recognizing various fields and values that are relevant to our problem statement.

My Contribution

- Personally collected roughly 30 reports through Internet searches
- Used the data collected to identify report trends – Indian standards, normal ranges of values and units commonly used in Indian reports
- This information was ultimately applied in the Report Generation section as well
- The collected reports were given to other team members for annotation

Project's Module -

1. Software Requirements
   - Web Browser (Internet Connection)
   - Google Drive

2. Hardware Requirements

- Processor: Intel core i5, 8th gen and above
- RAM: 4GB and above
- Storage: 80GB and above
- Display: CRT/ LCD/ LED
- Camera: pixel resolution of at least 5 MP
- Mouse: Compatible mouse
- Keyboard: normal or multimedia

3. Module Dependencies:
   - EasyOCR
   - Tesseract

4. Module Implementation
   - Reports were collected via browsing the internet
   - Reports were uploaded to a shared google drive folder
   - Report trends were noted for further modules

2. Module Testing Strategies:
   - Reports were all passed through our OCR model once to determine which report formats generated the most accurate text extraction results.

3. Module Deployment:
   - One time execution of the OCR module to identify the desired report formats to be chosen for further augmentation and annotation.

2. <u>Report Generation Module</u>

Project's Module Objectives

- To generate useful insights based on the extracted text from the blood reports

Project's Module Scope

- To collect medical data (normal ranges, likely causes, solutions) from reputable sources and store the findings in tables segregated based on age and gender.
- To perform calculation on the user's extracted data to determine abnormal result values and percentage deviations.
- To provide the user with a detailed report of inferences based on their blood test results.
- The report should be easily consumable for the average reader without medical knowledge

My Contribution

- Collected information from 50+ government websites, publications, medical literature sources regarding normal ranges of all fields (hb, wbc, rbc, leukocytes, neutrophils, etc).
- Determined that normal ranges differ significantly between 4 age groups: children, adolescents, adults, elderly.
- Maintained an excel sheet of 4 separate normal ranges tables for different age groups
- Adjusted normal values based on gender.
- Collected information about the most likely causes and treatment suggestions for both high and low values for each field based on age and gender
- Wrote a python program to calculate deviations from the normal range and the percentage. Wrote code to handle different units in different reports.
- Wrote code to display all the abnormal results along with possible causes and solutions in a clear report format, with explanations that are easy for the user to understand.

Project's Module -

1. Software Requirements
   - Python
   - Anaconda
   - Text editor
   - Jupyter NB
   - Web Browser (Internet Connection)
   - MS Excel

2. Hardware Requirements
   - Processor: Intel core i5, 8th gen and above
   - RAM: 4GB and above
   - Storage: 80GB and above
   - Display: CRT/ LCD/ LED
   - Camera: pixel resolution of at least 5 MP
   - Mouse: Compatible mouse
   - Keyboard: normal or multimedia

3. Module Dependencies:
   - Pandas
   - Numpy
   - openpyxl

4. Module Implementation
   - Collected normal ranges information
   - Collected causes/solutions information
   - Wrote a program to identify and calculate abnormal deviations
   - Wrote a program to display the information in report format

- Merged code with web app code and converted report to downloadable pdf format

5. Module Testing Strategies:
   - Code was tested against various reports with different units/formats to identify issues
6. Module Deployment:
   - Report generation code was merged and deployed with our web application. The user can download a pdf version of the generated report through the web application.

3. GUI Module

Project's Module Objectives

- To connect all the modules together, provide a simple user interface which will allow the user to upload a blood report image and get the structured pdf inference based off of it.

Project's Module Scope

- This module aims at bringing all the other modules together and facilitating a smooth process flow.
- Provides the user with a simple interface to upload their report and download results

My Contribution

- Aided with the backend coding of the 'download report' section of our web GUI
- Helped to fix bugs with pdf generation
- Aided with the UI design of the frontend

Project's Module -

7. Software Requirements
   - Python
   - Anaconda
   - Text editor
   - Web browser (internet connection)

8. Hardware Requirements
   - Processor: Intel core i5, 8th gen and above
   - RAM: 4GB and above
   - Storage: 80GB and above

- Display: CRT/ LCD/ LED
- Camera: pixel resolution of at least 5 MP
- Mouse: Compatible mouse
- Keyboard: normal or multimedia

9. Module Dependencies:
   - Flask
   - Pdfkit
   - request

10. Module Implementation
    - Converted display statements to variables for pdf generation
    - Fixed bugs relating to issues with storing String in a variable
    - Corrected escape code parsing issues
    - Corrected data type issues

11. Module Testing Strategies:
    - Code was tested against various reports with different units/formats to identify issues

12. Module Deployment:
    - Web application was deployed locally

PART - B

INDIVIDUAL CONTRIBUTION

Problem Statement: Hematology report data extraction and diagnosis generation

Name of the Student: Ayush Shrivastav

Module Title: Annotations, OCR module, frontend

Project's Module Objectives –

- GUI module
- To connect all the modules together, provide a simple user interface which will allow the user to upload a blood report image and get the structured pdf inference based off of it.

Project's Module Scope –

- This module aims at bringing all the other modules together and facilitating a smooth process flow.

Project's Module(s) - *Individual Contribution*

- Complete designing of the frontend.
- Structuring the backend based off of all the modules

- Hardware & Software requirements
  - Software: Python, Anaconda, Text editor, Web browser (internet connection)
  - Hardware:
    - Processor: Intel core i5, 8th gen and above
    - RAM: 8GB and above
    - Storage: 80GB and above
    - Display: CRT/ LCD/ LED
    - Camera: pixel resolution of at least 5 MP
    - Mouse: Compatible mouse
    - Keyboard: normal or multimedia
- Module Interfaces
  - Object Detection: Accepts input images and returns the detected objects' bounding boxes and corresponding class labels.
  - Image Processing: Takes the input images and performs cropping based on the detected objects' bounding boxes.
  - OCR: Receives the cropped images and extracts the text information from them.
  - Report Generation: Takes the extracted text data and performs data cleaning, abnormal value calculation, and generates a detailed report.

    - Data Preparation: Accepts the input dataset and performs data augmentation techniques to expand the dataset for training the object detection model.

- Module Dependencies
    - Frameworks (apart from the frameworks used in other modules):
    - Flask
    - Pdfkit
    - Request

- Module Design
    - Object Detection: Utilizes YoloV8 model and Roboflow for training and detection.
    - Image Processing: Uses the detected object bounding boxes to crop the images accordingly.
    - OCR: Implements EasyOCR and Pytesseract libraries for text extraction.
    - Report Generation: Performs data cleaning, abnormal value calculation, and generates a comprehensive report.
    - Data Preparation: Applies data augmentation techniques to expand the training dataset.

- Module Implementation
    - Setting up the python virtual environment (env1):
    - Installing all the libraries and frameworks required by the modules
    - Setting up the Flask application environment
    - Directory for images: static
    - Directory for html templates: templates
    - Directory to store temporary data that gets created during report generation
- Module Testing Strategies
    - Object Detection: Testing the accuracy of the detection algorithm by evaluating the precision and recall metrics.
    - Image Processing: Verifying the correctness of the cropping process by comparing the cropped images with the original images.
    - OCR: Evaluating the accuracy of text extraction by comparing the extracted text with the ground truth.
    - Report Generation: Testing the data cleaning and abnormal value calculation processes, and verifying the correctness of the generated report.
    - Data Preparation: Verifying the effectiveness of data augmentation techniques by evaluating the impact on the object detection model's performance.

- Module Deployment
    - Locally run.

PART - B

INDIVIDUAL CONTRIBUTION

Hematology report data extraction and diagnosis generation

Aniket Sharma
1032192119
PB56

Module Title: Data Sourcing, OCR Module, and Frontend Development

Project's Module Scope - Individual Perspective:
The scope of my module includes sourcing the necessary data, developing the OCR module, and contributing to the frontend development to provide a user-friendly interface.

Project's Module Objectives –

Data Sourcing:

To collect a diverse dataset of images or documents to be used for training and testing the system. Data collected from various Indian pathology lab web portals so to have Indian standard template.

Project's Modules - Individual Contribution:
Responsible for collecting a diverse dataset of images or documents for training and testing the system.

Module Interfaces:
Interfaces with various sources (open datasets, web scraping, etc.) to collect the required dataset for training and testing.

Module Dependencies:
Depends on the availability of diverse data sources and may require data preprocessing or formatting.

Module Design:
Develop strategies to collect a diverse dataset, including open datasets, web scraping, or other relevant sources.

Module Implementation:
Collect and preprocess the dataset from various sources, ensuring its compatibility with the system.

Module Testing Strategies:

Test the collected dataset to ensure its diversity and suitability for training and testing the system.

Module Deployment:
Integrate the collected dataset into the system for training and testing purposes.

OCR Module:

To develop an OCR module that accurately recognizes and extracts text from images or documents.

Project's Modules - Individual Contribution:
Responsible for developing the OCR module to accurately recognize and extract text from images or documents.

Module Interfaces:
Interfaces with input images or documents and provides the recognized and extracted text as output.

Module Dependencies:
Depends on the input images or documents and requires access to OCR libraries or APIs for text recognition and extraction.

Module Design:
Select and implement appropriate OCR algorithms or libraries for text recognition and extraction.

Module Implementation:
Implement OCR algorithms or utilize OCR libraries to recognize and extract text from input images or documents.

Module Testing Strategies:
Conduct testing using different input images or documents to evaluate the accuracy and reliability of the text recognition and extraction.

Module Deployment:
Deploy the OCR module as part of the overall system, ensuring its proper integration and functionality.

Frontend Development:

To design and implement the user interface (UI) for the system, allowing users to interact with the system effectively.

Project's Modules - Individual Contribution:

Responsible for designing and implementing the user interface (UI) components to provide a seamless user experience.

Module Interfaces:
Interfaces with the backend functionality and provides a user-friendly interface for users to interact with the system.

Module Dependencies:
Depends on the backend functionality and may require integration with the OCR module and other system components.
Frameworks (apart from the frameworks used in other modules):

- Flask
- PDFkit
- request

Module Design:
Design UI components, user input forms, and ensure seamless integration with the backend functionality.

Module Implementation:
Code and implement the UI components and integrate them with the backend functionality.
Setting up the python virtual environment (env1):

- Installing all the libraries and frameworks required by the modules
- Setting up the Flask application environment
- Directory for images: static
- Directory for html templates: templates
- Directory to store temporary data that gets created during report generation

Module Testing Strategies:
Perform user testing to ensure the UI components are intuitive, responsive, and provide a seamless user experience.

Module Deployment:
Integrate and deploy the frontend components, ensuring their proper interaction with the backend functionality and other system modules.
Locally run.

Hardware & Software requirements:

Software: -

- Jupyter NB
- Python
- Anaconda
- Text editor

Web browser (internet connection)

Hardware: -

Processor: Intel core i5 8th gen and above
RAM: 8GB and above
Storage: 80GB and above
Display: CRT/ LCD/ LED
Mouse: Compatible mouse
Keyboard: normal or multimedia