MINOR PROJECT

Submitted by:

Gautam Joshi - 18103218 – B7

Vishal Sadhnani - 18103211 - B6

Kshitij Bhargava – 18103193 - B6

Under the supervision of:

Dr Kavita Pandey

PROJECT TITLE

M.A.V.I

(A Medical Assistant For Visually Impaired)

Photo by novia wu on Unsplash

1. The motivation behind the project:

There are many problems faced by Visually impaired people. One of the problems faced by them is to read their own medical reports. A medical assistant is proposed by us for reading medical reports to help Visually Impaired people to extract the required information from reports.

2. Why only Medical reports ?:

We have gone through many research papers and there are already many applications which have been used to help blind persons to solve many of their problems. So we are focussing on medical reports which have not been done before.

3. Problem Statement

A visually impaired individual is one who loses the vision when a part of the eye or the brain that processes images becomes diseased or damaged. Visually Impaired people face a lot of problems like navigation around places, identifying objects, identifying people's emotions or expressions, detecting obstacles and so on. One area that needs to be focused on is where the visually impaired person cannot be able to read medical reports and medical product's name. Those problems can be rectified using existing technology such as ML, IoT, Computer Vision, image processing etc. Therefore, A medical assistant using existing technologies is proposed by us for reading medical reports and medical product's name.

This medical assistant will help those without vision in reading their medical reports.

4. Critical Analysis of Research Papers:

No.	Problem description	Data used	Model used	Data Pre-Processing	Accuracy	references

1	To read text labels in their day to day life which can be executed by a surroundin g object in the camera view	Commo nly used grocery product	OCR and TTS, Cascade-Ada boost model	Text localization, Gaussian background subtractions algorithm	detection of character recognition is 80%	https://iee explore.iee e.org/docu ment/8463 923
2	To read text labels from prescription in their day to day lives	Scannin g of prescript ions	OCR	Edge detection	Accuracy is good	https:/ /www. ijeast. com/p apers/ 265-2 68,Tes ma41 2,IJEAS T.pdf
3	a smart spec for the blind persons which can perform text detection thereby produce a voice output	Text in the form of black and white and as well as a coloured image	OCR and TTS	Grey scaling, adaptive thresholding, hough lines,de-ske wing images, contour detection	Accuracy is good	https:/ /ieeex plore.i eee.or g/abst ract/d ocum ent/83 92145
4.	Reading out texts easily with -full -depth analysis of the textual content -	Any text	OCR, espeak		good	https:/ /ieeex plore.i eee.or g/abst ract/d ocum ent/80 94103

5	To help to read out the report of the medicine The pharmacist will store reports of the medicine in the server. The details will store in the TOMCAT server and server will store the data into the MYSQL database		This proposed framework utilizes module Text Recognition	Content Recognition utilizes some calculation and utilizes two strategy – OCR Pre – preparing and OCR Post – handling..	Text detection followed by recognition using supervised pattern recognition algorithm not only improves accuracy but also increases the speed of the system	https://www.ijert.org/textrecognitionand-m edicine-identification-byvisually-impairedp
6	To help blind persons to read text labels from different patterns with complex backgroun ds found on many everyday products of Hand-held objects		Text localization algorithm which combines rule-based analysis and learning-based classifier training. , A motion-base d algorithm, A portable camera-base d assistive text reading framework	Off-the-shelf OCR is used to perform text recognition. The binarization of the text region is performed by OTS's method. The binarization of the text region is performed by OTS's method	The average precision rate of the proposed system is 81.27 %, The system produces an average recall rate 99.313%,	https://www.ijireeice.com/upload/2016/june-16/IJI REEICE%206 4.p

7	Proposed a mobile application of assisting blind persons by identifying labels on the products.	This project uses a pre-trained model obtained by training COCO dataset on MobileNet network	The YOLO algorithm has been used	OCR is used for detecting text from an image	OCR performs well in case of reading texts with accuracy as high as 70-90%	https://core.ac.uk/reader/288507980

5. Type of project:

This is a research cum development project as there are research papers which uses different methodologies and none of them is considered to be the best, and hence our try would be to achieve a better or a comparable accuracy at the end of the project.

6. Literature Review

Firstly, after doing much research we proposed an idea to develop an application using OCR, image processing and TTS methods. Then we intended to build a flutter application which is a framework used for making cross-platform mobile apps. And the reason to use flutter is that it is cross-platform(that means developers write code once and apply it to all the platforms – iOS, Android, web platform, macOS) and it provides various packages to build an application we need.

7. Implementation Details:

7.1 Technology Used

- **Flutter Framework:**
  - In order to scan and read the medical reports for visually impaired people, we need some platform. So we have used Flutter to build an application where they can use it from their Android or IOS device to scan the medical report.
- **Alan:**
  - Alan provides the complete conversational platform to let you add a voice interface to your app
  - Alan offers easy-to-integrate SDKs for different platforms: Web, iOS, Android, Flutter, Ionic, React Native, Cordova.
- **Flutter TTS:**
  - This **flutter_tts** plugin used to interact with native functionality. Under the hood, it uses `TextToSpeech` for Android, and `AVSpeechSynthesizer` for IOS platform. In this, we are exploring the methods of **flutter_tts** plugin
- **Flutter mobile vision:**
  - Flutter implementation for Google Mobile Vision.
  - Mobile Vision is an API which helps us to find objects in photos and video, using real-time on-device vision technology by Google. This framework has the capability to detect objects in photos and videos.

7.2 System Requirements

a) Hardware Requirements

1. Flutter SDK
   - Disk Space: 1.32 GB (does not include disk space for IDE/tools).
2. VS Code
   - 1.6 GHz or faster processor
   - 1 GB of RAM
3. Android's Emulator
   - 64-bit distribution capable of running 32-bit applications

- 3 GB GAM
- 2 GB hard disk space
- A minimum resolution needs to be 1280 x 800

b) Software Requirements

1. Flutter SDK

   - Operating Systems: Windows 7 SP1 or later (64-bit), x86-64 based
   - Tools: Flutter depends on these tools being available in your environment.

     ➔ Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10)
     ➔ Git for Windows 2.x, with the Use Git from the Windows Command Prompt option.

     (But we used VS Code integrated with Git Bash and Flutter Debugging Tools)

2. VS Code (Code Editor or IDE)

   - Operating System: Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)

3. Android's Emulator

   - Operating System: Windows

7.3 Methodology

In flutter, everything is a widget so we created a widget containing a header with an app's name "M.A.V.I", three buttons with different functionalities, and alan's floating button and a container to show parsed text. The first button is for recognizing blood reports. The second one is for detecting medical products name or brand. The last one is for recognizing normal text reports.

- Blood Report's Button

  When a user clicks this button, a function will be called and the user has to pick an image from the gallery. Then in function image will be prepared and send to OCR API and OCR API will give result object containing our text. Then we extract that

text and apply some logic on it to get a logical solution. The parsed text we extracted will be sent to Flutter_TTS function for this button to speak out the text to the user.

This will pick the image from the gallery.

```
final imagefile = await ImagePicker()

        .getImage(source: ImageSource.gallery, maxWidth:
670, maxHeight: 970);
```

This will prepare the image

```
var bytes =
Io.File(imagefile.path.toString()).readAsBytesSync();

    String img64 = base64Encode(bytes);
```

This will send image to api

```
var url = 'https://api.ocr.space/parse/image';

    var payload = {"base64Image":
"${img64.toString()}"};

    var header = {"apikey": '3abb227f0b88957'};

    var post = await http.post(url, body: payload, headers:
header);
```

This will get back result from api

```
var result = jsonDecode(post.body);
```

- Medical Product's Button

When user click this button or call it to click through alan voice, it will call a function and it will use Flutter_Mobile_Vision to initialize camera and scan a product to recognize a text from it. Flutter_Mobile_Vision will use all OCR, Image processing and ML methods behind the scenes.

- Normal Text Report's Button

  This button's code is the same as the blood report's button except it doesn't have the logic to extract blood report's information. It will recognize text report easily,

- Alan Voice Button

  When a user says a wake-up word "Hey alan" or simply click it, it will wake up to listen. Then it will hear whatever the user's intent was and do the specific job, for example, it will open up the gallery, open the camera to scan, gives the current visual of the page and also some information about the page. To do all this happen, we have to add a package known as alan_voice and add an alan button, then to do the specific job we have to code in alan studio on the server and specify intents, this intent will be handle in our code to perform a certain task.

  Alan's server code looks like this:

```
intent('upload a blood report', p=> {
   p.play({command: 'upload', route: 'takeAnImage'});
   p.play("let's upload a blood report");
});

intent('scan a medical product', p=> {
   p.play({command: 'scan', route: 'scanAnProduct'});
   p.play('scanning...please click somewhere');
});

intent('upload a text report', p=> {
   p.play({command: 'nupload', route: 'grabImage'});
```

```
   p.play("let's upload a text report");
});


intent(`i'm on which screen ?`, p => {
    let screen = p.visual.screen;
    switch (screen) {
        case "gallery":
            p.play(`you are in gallery`);
            break;
    case "scanning page":
            p.play(`you are on scanning page, please click
somewhere.`);
            break;
        default:
            p.play('you are on home page');
    }
});
```

7.4 Project Architecture

Figure 1 is an abstract view and **Figure 2** is an app view

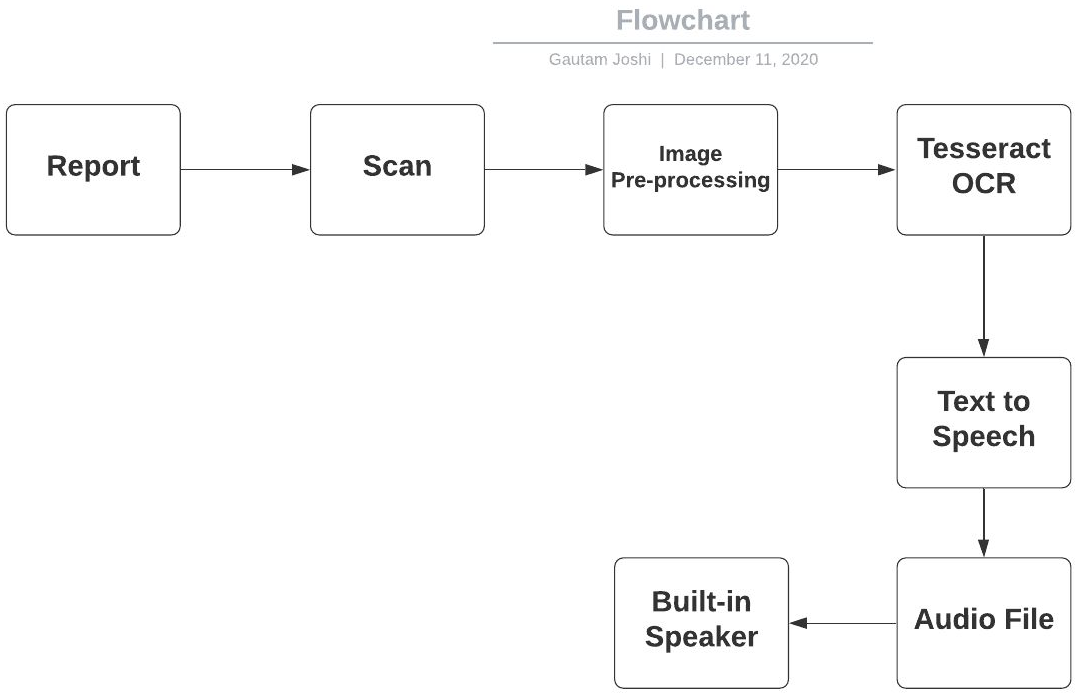


Figure 1

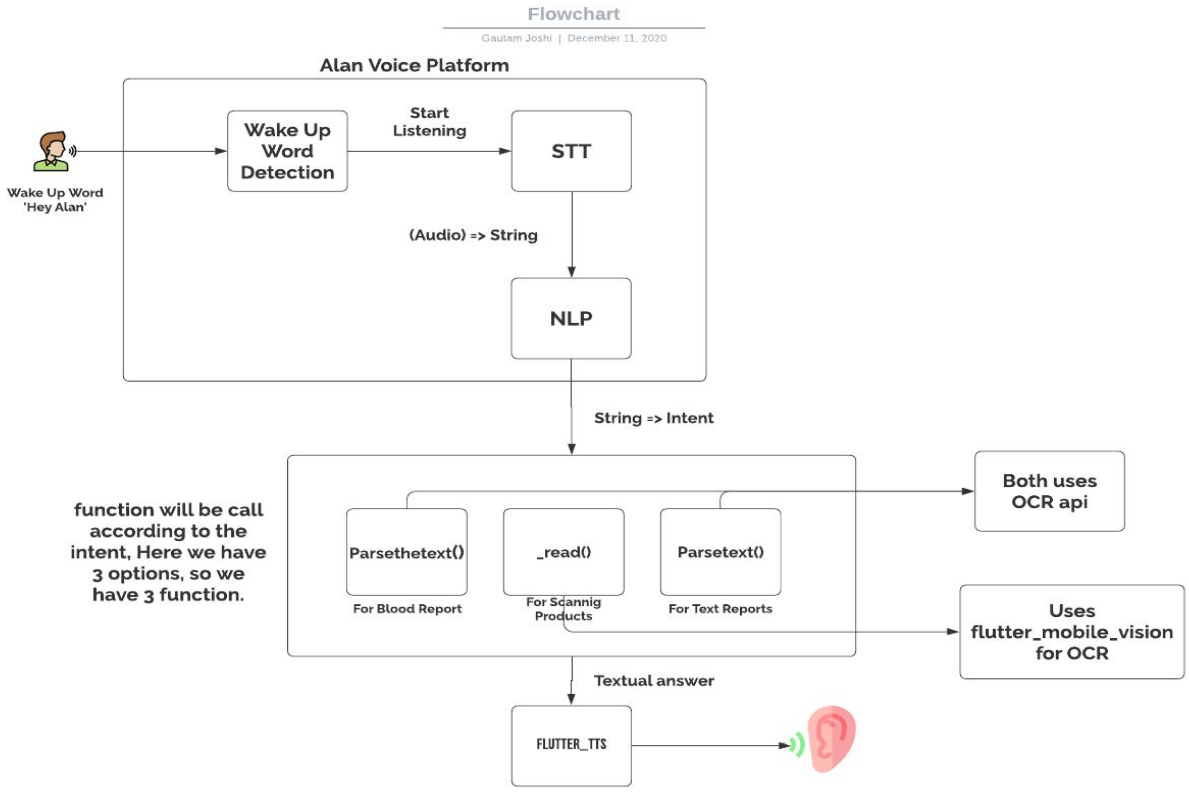


Figure 2

7.5 Result

When the user says “Hey alan”, alan voice platform starts listening to the user’s intent. When the user says to upload a blood report, it will open a gallery and when we click an image, it will parse the text like in **Figure 4** and TTS will speak the parsed text. When the user says to scan a product, it will open a camera and starts scanning in **Figure 3** but when the user clicks the box which is formed around text it will parse the text in **Figure 6** and TTS will speak the parsed text. When the user says to upload a text report, it will recognize text and parsed text like **Figure 5** will be spoken by TTS.

Figure 3

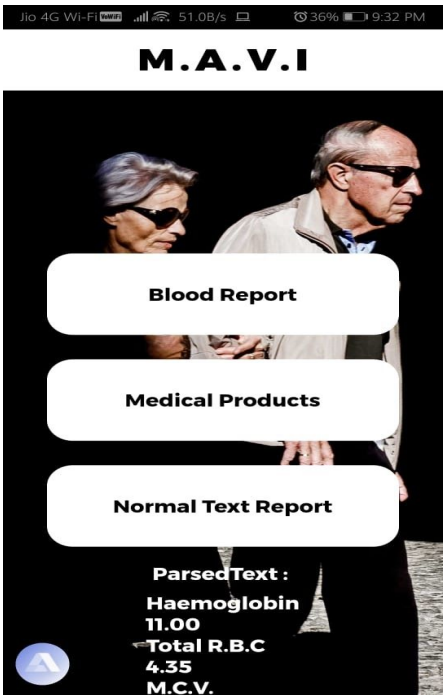


FIgure 4

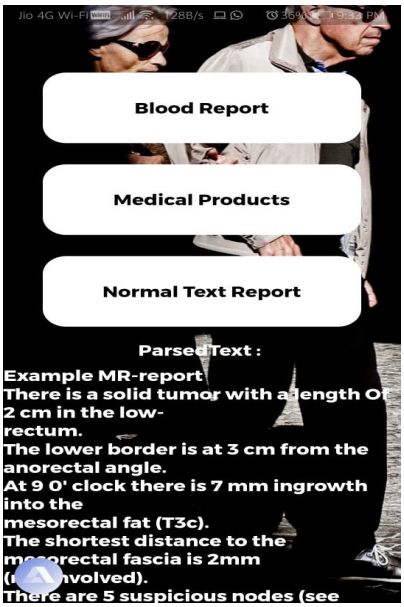


Figure 5

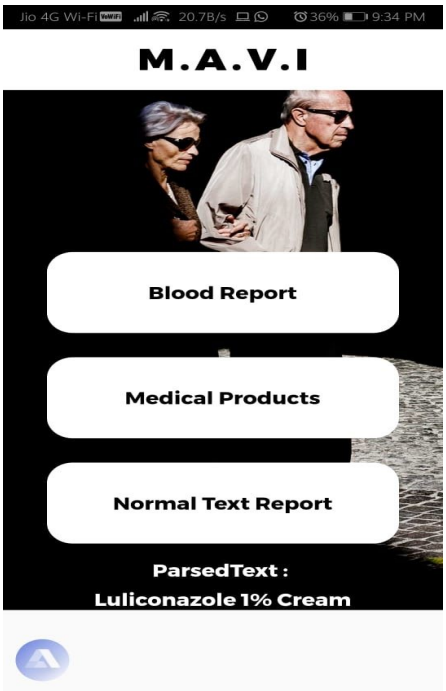


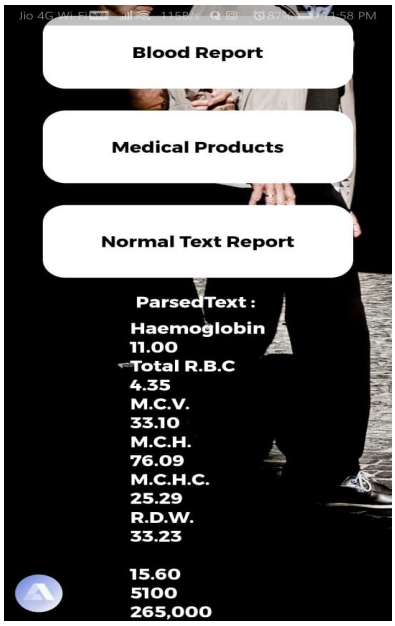
Figure 6

7.6 Testing

We have tested some cases:

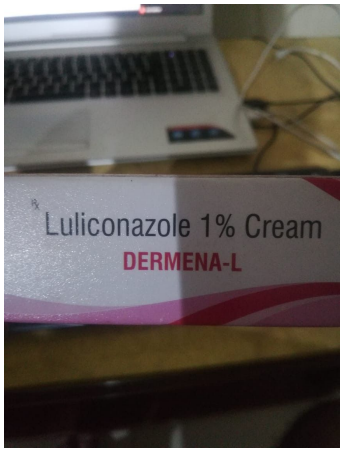
1. For Blood Report

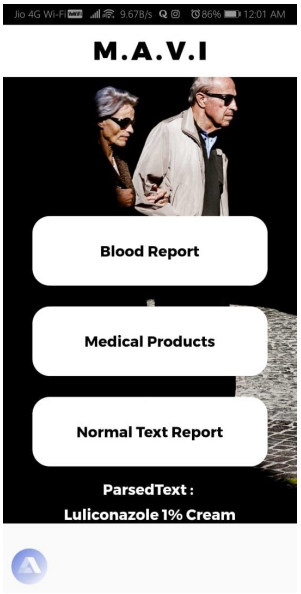
TEST		RESULT	UNIT
BLOOD COUNTS & INDICES			
Haemoglobin	⇩ :	**11.00**	gm%
Total R.B.C.	:	4.35	mill/cmm
P.C.V.	⇩ :	**33.10**	%
M.C.V.	⇩ :	**76.09**	fL
M.C.H.	⇩ :	**25.29**	pg
M.C.H.C.	:	33.23	%
R.D.W.	⇧ :	**15.60**	%
Total W.B.C.	:	5100	/cmm
Platelet Count	:	265,000	/cmm

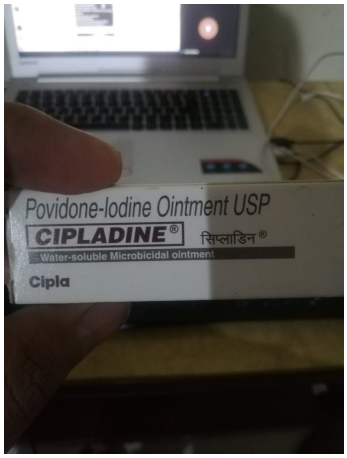


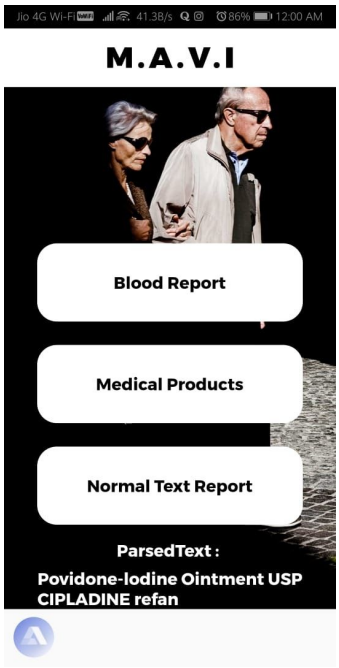
Test Name	Result
COMPLETE BLOOD COUNT (EDTA Blood)	
WBC Count (Impedence)	6840
RBC Count (Impedence)	5.75
Haemoglobin (SLS Method)	17.4
Haematocrit (PCV)	47.2
(RBC Pulse Height Detector Method)	
MCV (Calculated)	82.1
MCH (Calculated)	30.3
MCHC (Calculated)	36.9
Platelet Count (Impedence)	298000
RDW-CV (Calculated)	11.8
DIFFERENTIAL COUNT	
Neutrophils (Flowcytometry)	53.9
Lymphocytes (Flowcytometry)	35.2
Monocytes (Flowcytometry)	9.2
Eosinophils (Flowcytometry)	1.3
Basophils (Flowcytometry)	0.4
IG	0.10

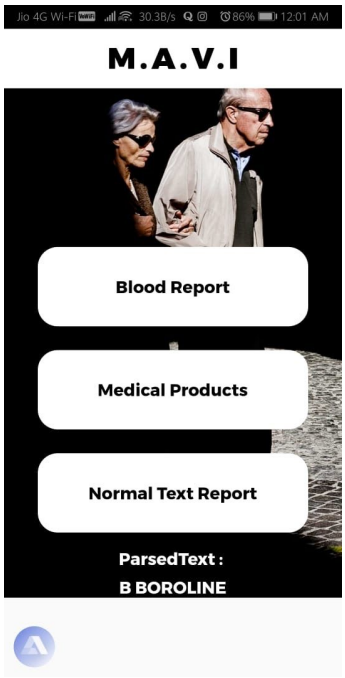
2. For Medical Products



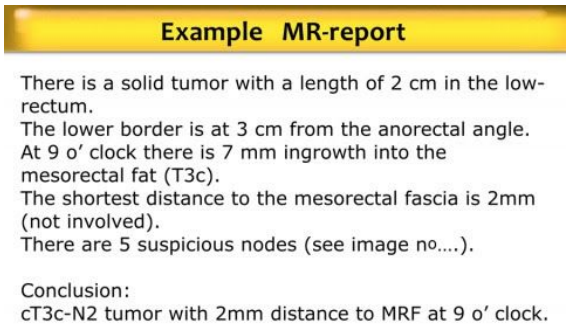


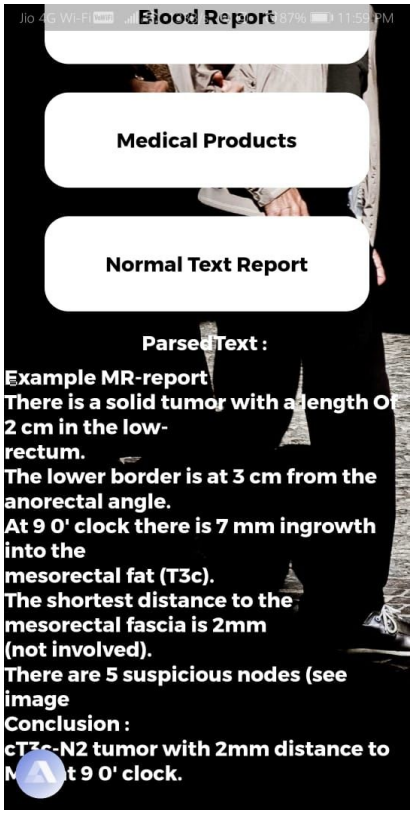






3. For Text Reports

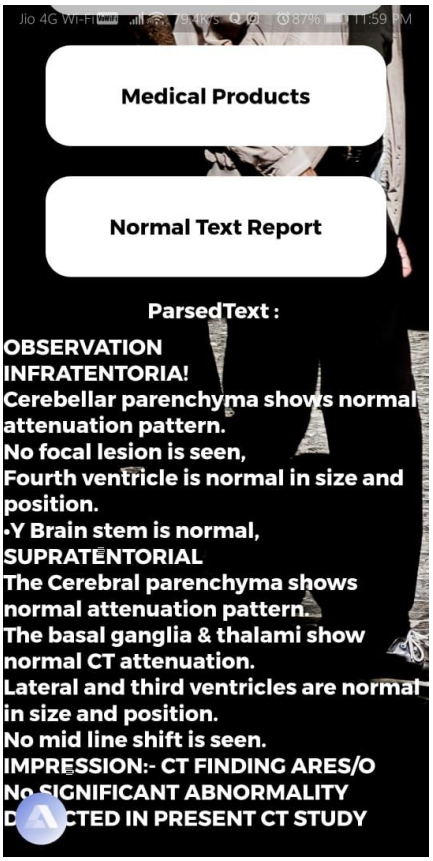




Abdomen CT without contrast adminstration with scanning from liver dome tothe pelvis at 10mm intervals is performed. The result showed:

1.The liver is normally positioned and has normal size and smooth borderd.Its internal structure and attenuation values are normal. The intrahepatic and extrahepatic bil ducts and gallbladder are unremarkable.
2.The spleen is orthotopic and of normal size. .
3.The pancrease is normal in size,position, and internal structure with smooth,lobulated outer contours.The pancreatic duct is unobstructed.
4.Both kidneys show normal size and position. The renal parenchyma show normal width and structure. The renal pelvis and calies show a normal configuration.The urinary tract is unobstructed.
5.The adrenal glands are unremarkable.
6.Major blood vessels appear normal,and there is no evidence of lymphoadenopathy.
7.There are no ascites or pleural effusion.

Imp:Normal picture in upper abdomen CT study



7.7 Future Scope

In future, we have to focus on the live camera to detect text using firebase ML kit, now this is not possible because of some error is thrown every time regarding FlutterJNI. We also have to focus on the user interaction part where the user has the command to every task.

8. References

1. https://flutter.dev/docs
2. https://pub.dev/
3. https://youtu.be/L-c-ZyX-KtY

4. https://www.youtube.com/watch?v=x0uinJvhNxI
5. OCR Using Flutter. Optical character recognition is a... | by Jitss | FlutterWorld
6. https://ocr.space/ocrapi