Group No.: <As per attendance>

Project Title: Optical Character Recognition(OCR)

Member 1 Member 2

Name: Rutuja Badbe Name: Swarali Purandare

Div. & Roll no.: L 10 Div. & Roll no.:B 56

GR No.: 1710624 GR No.:11810489

E-mail ID:rutuja.badbe17@vit.edu E-mail ID:swarali.purandare18@vit.edu

**Introduction:**

What is Optical Character Recognition (OCR)?

Optical Character Recognition (OCR for short) is a technique that converts digital images of text into machine-readable data.

There are two methods to perform OCR: matrix matching and feature detection.

Matrix matching is the simpler of the two;

it takes an image and compares it to an existing library of character matrices or templates to generate a match.

Feature detection is more complex as it looks for general features like diagonal lines, curvatures, intersections, etc.

and compares it to other features on the image within a certain distance.

How does it work?

The tool first performs a layout analysis on the image to segment the location of the text. After the general location is detected,

the OCR module then performs a text recognition analysis on the specified location to generate the text.

Finally, errors are corrected at a post-processing step by feeding it through a language model or dictionary.

All of this is done through a convolutional neural network.

What is Google Vision API?

Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API.

It quickly classifies images into thousands of categories, detects individual objects and faces within images,

and finds and reads printed words contained within images.

**Working Methodology:**

1. Add dependency to include the play-services-vision dependency. implementation 'com.google.android.gms:play-services-vision:18.0.0'
2. In createCameraSource function we declare object of TextRecognizer .TextRecognizer detector object processes images and determines what text appears within them. TextRecognizer can be used to detect text in all types of images.
3. Check if TextRecognizer is operational.
4. Create a CameraSource, which is a camera manager pre-configured for Vision processing.

Set the resolution high and turn autofocus on, because that's a good match for recognizing small text.

setRequestedPreviewSize(1280, 1024) setRequestedFps(2.0f)

setAutoFocusEnabled(true)

1. Implement surfaceChanged and surfaceDestroyed() .
2. Create a Processor which will handle detections as often as they become available Detector.Processor<TextBlock>
3. Override receiveDetections to detect and store it into StringBuilder using toString() print the text on the screen.

**Insights:**

1.In tesseract code of OCR the dependencies we used were in deprecated APIs. Which caused a lot of errors.

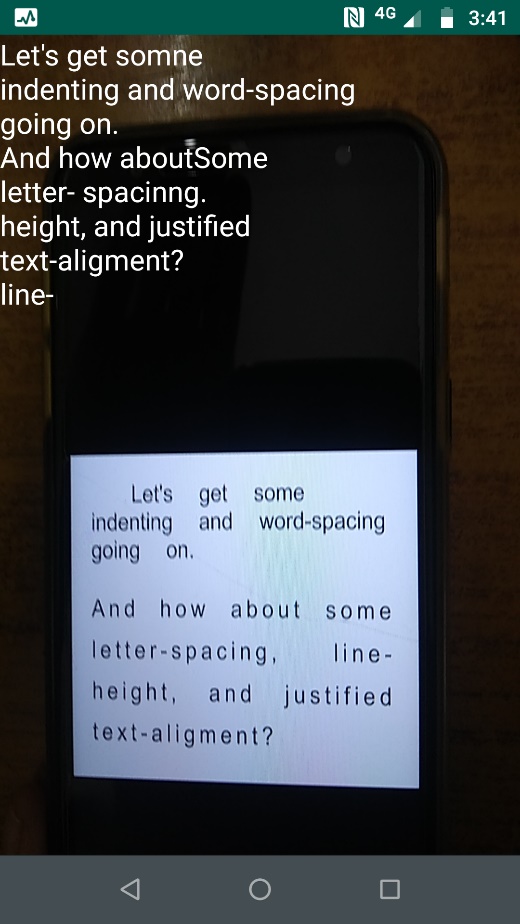
Eg-butterknife

2.Android device showed as not connected because the USB Debugging was Off.

3.Had to Migrate to androidx to use latest version of dependencies in tesseract.

4.

**Result:**



**References:**

- https://youtu.be/xoTKpstv9f0

- https://youtu.be/WGvAhUrqIeY

- https://codelabs.developers.google.com/codelabs/mobile-vision-ocr/#4