**TRF Summer internship and training course 2019 Task Report**

TASK 3

Group No.: 14

Project Title: Android Application for plotting V-I characteristic curve using OCR

Member 1 Member 2

Name: Rutuja A Badbe. Name: Swarali A Purandare.

Div. & Roll no.: L-11 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

1. INTRODUCTION

Objective:

* To detect the Voltage and current value from Battery using OCR and plot the Current Vs Voltage graph.
* To learn how to plot graph using data collected by OCR.
* Graph is created using GraphView Library of Android.

SOFTWARE USED:

**Tesseract** is open-source, free and can run locally. But it also requires training and tweaks, and in general its recognition rate is worse **than** that of the cloud **ocr** services from **Google**, Microsoft and **OCR**.space.In this project we have used google vision.

2. WORKING METHODOLOGY

1. Add dependency to include the play-services-vision dependency. “implementation 'com.google.android.gms:play-services-vision:18.0.0' “ and

Add dependency to include the Graph View –

“implementation 'com.jjoe64:graphview:4.2.2'”

1. 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.
2. Check if TextRecognizer is operational. The output of the TextRecognizer can be retrieved by using SparseArray and StringBuilder.
3. 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)

### Implement surfaceChanged() and surfaceDestroyed() .

### surfaceChanged:This is called immediately after any structural changes (format or size) have been made to the surface.

surfaceCreated:This is called immediately after the surface is first created. Implementations of this should start up whatever rendering code they desire.

onResume()->onSurfaceCreated()->onSurfaceChanged()

onPause()->onSurfaceDestroyed()

1. Create a Processor which will handle detections as often as they become available Detector.Processor<TextBlock>. We have used TextBlock to retrieve the paragraph from the image using OCR.
2. Override receiveDetections() to detect and store it into StringBuilder using toString() print the text on the screen.
3. Store the detected data in string. If string has numbes it will parse the string into integer.Create a class DataSample to store values of current and voltage .This c;ass is used to initialise datapoint variable.
4. Pass the integers to datapoint series using get and set functions in DataSample.java.
5. Add series to graph view for ploting. mGraph.addSeries(series);

3. INSIGHTS

* **NumberformatException-**

OCR detects numbers as well as other characters. But to plot a graph we only needed numbers. If the characters other than number get detected, Application used to crash down.

Therefore, we add try and catch statement with numberformat Exception to get rid of this problem.

* **ArrayIndexOutofBounds –**

The ArrayIndexOutOfBoundsException, also known as java.lang.ArrayIndexOutOfBoundsExcepiton is one of the most common errors in Java program. It occurs when Java program tries to access an invalid index e.g. an index which is not positive or greater than the length of an array.

* **NullPointerException-**

In Java, a special null value can be assigned to an object reference. NullPointerException is thrown when program attempts to use an object reference that has the null value.

These can be:

Invoking a method from a null object.

Accessing or modifying a null object’s field.

Taking the length of null, as if it were an array.

Accessing or modifying the slots of null object, as if it were an array.

Throwing null, as if it were a Throwable value.

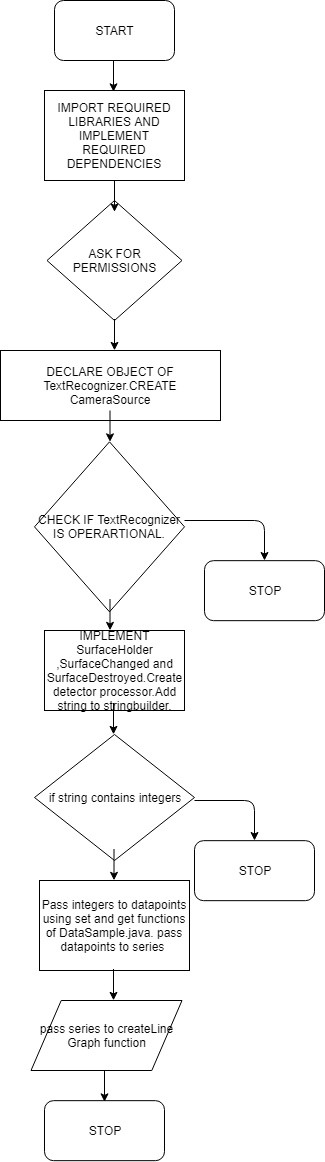
When you try to synchronize over a null object.

* The java.lang.Throwable class is the superclass of all errors and exceptions in the Java language. Only objects that are instances of this class (or one of its subclasses) are thrown by the Java Virtual Machine or can be thrown by the Java throw statement.
* CreateLineGraph() has input type only of Datapoint. Therefore, we needed to store current and voltage value in datapoint.
* OCR gets confused between similar alphabets and digits.

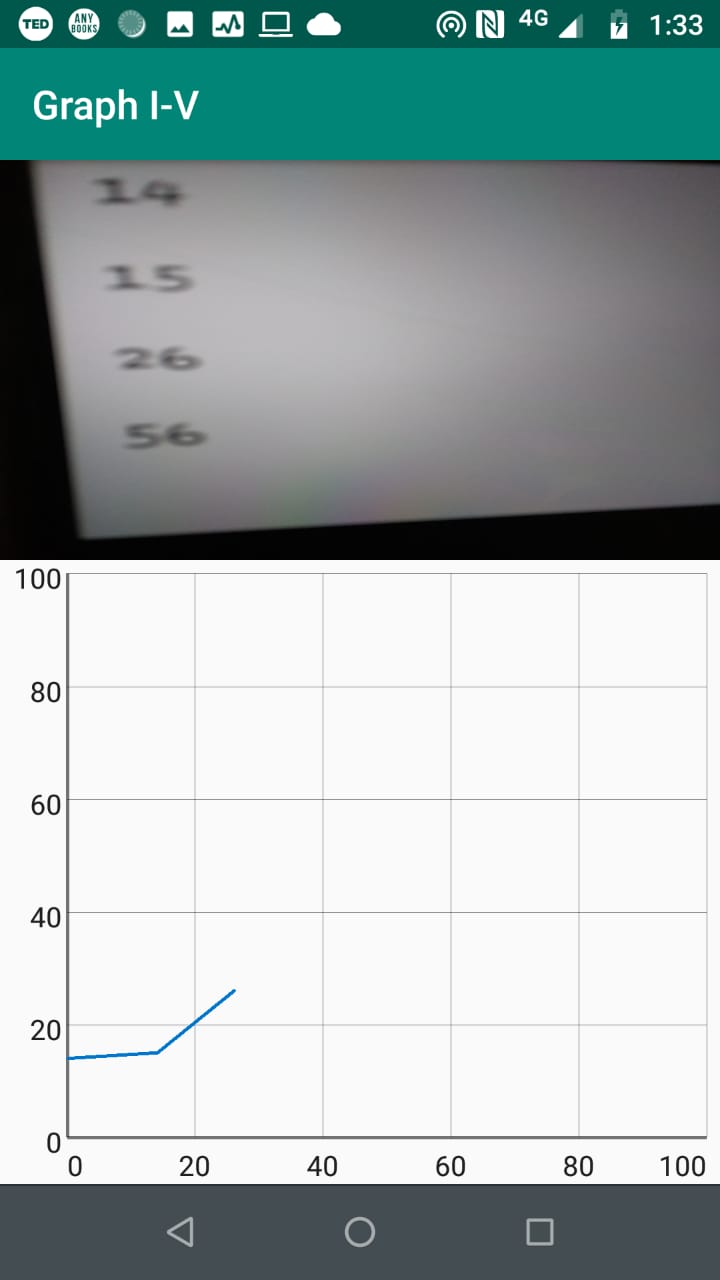
For e.g. It detects ‘5’ as ‘S’ OR ‘8’ as ‘&’

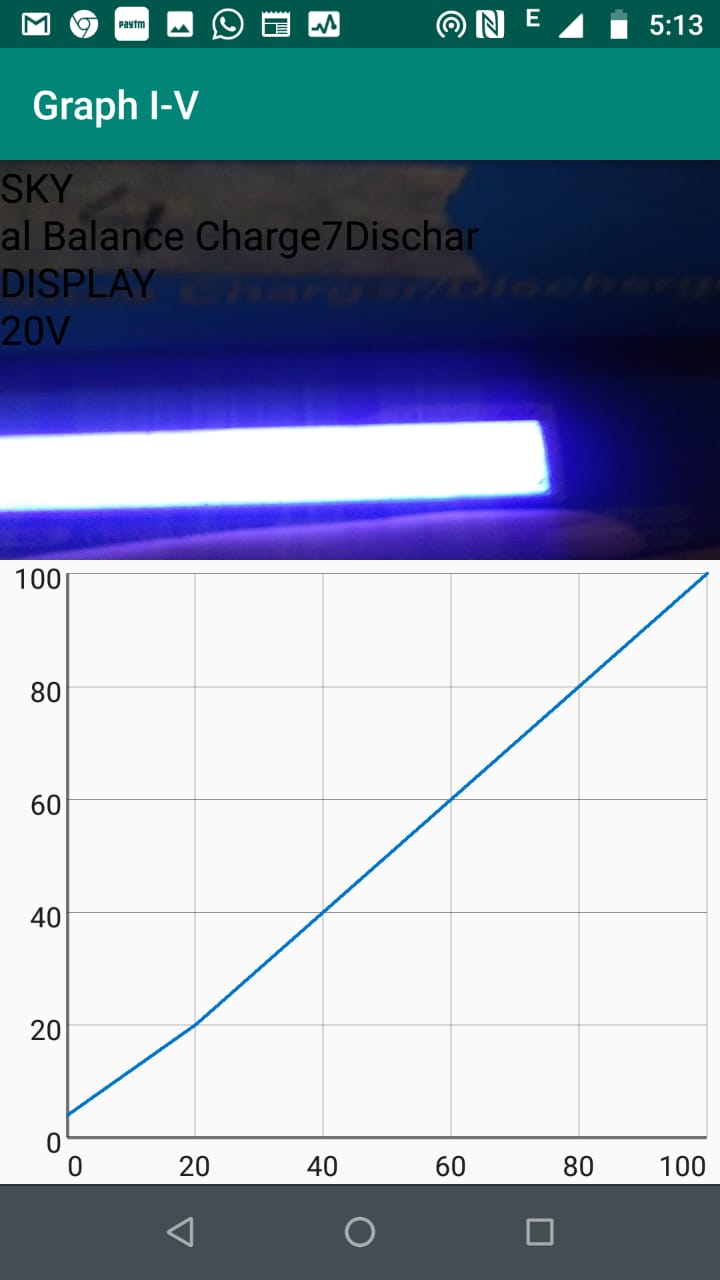
* OCR doesn’t detect decimal point in the decimal numbers.

4. Algorithm(Flowchart):



RESULT:





6. REFERENCES

* www.stackoverflow.com
* <https://github.com/jjoe64/GraphView>
* <https://www.codeproject.com/Articles/840623/Android-Character-Recognition>
* <https://www.ssaurel.com/blog/create-a-real-time-line-graph-in-android-with-graphview/>