CURRENCY DETECTOR FOR VISUALLY IMPAIRED

A Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

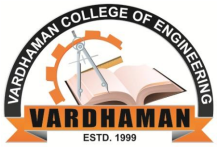
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

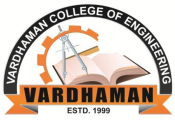
MD Rahman 18881A0595

SUPERVISOR
Dr.S. Hariharan
Professor
Department of CSE



Department of Computer Science and Engineering
VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

May, 2022



VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An Autonomous Institute, Affiliated to JNTUH

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project titled **CURRENCY DETECTOR FOR VISUALLY IMPAIRED** is carried out by

MD Rahman 18881A0595

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** during the year 2021-22.

Signature of the Supervisor
Dr.S. Hariharan
Professor

Signature of the HOD
Dr. Ramesh Karnati
Professor and Head, CSE

Project Viva-voice held on:

Internal Examiner **External Examiner**

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr.S. Hariharan**, Professor and Project Supervisor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. Ramesh Karnati**, the Head of the Department, Department of Computer Science and Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

MD Rahman

Abstract

Visually Impaired are the ones people who've imaginative and prescient impairment or imaginative and prescient loss. Problems faced via visually impaired in performing day by day sports are in outstanding range. They also face a whole lot of difficulties in monetary transactions. They are not able to apprehend the paper currencies because of similarity of paper texture and length between exceptional classes.

This Android Application makes use of text to speech conversion to present voice output of detected foreign money observe value to the person. For forex detection, this software uses Azure custom vision API the usage of ML classification technique to come across foreign money depending pics on mobile digicam.

Present the Currency Detector digicam facilitates visually impaired patients to discover and apprehend money. Using this software blind humans will click picture of the word and Application tells the person by speech how a great deal the foreign money be aware value is.

Keywords: Currency; App; Camera; Android.

Table of Contents

List of Figures

Abbreviations

Abbreviation	**Description**
TF Lite	TensorFlow Lite
MACVA	Microsoft Azure Custom Vision API
MANI	Mobile Aided Note Identifier
AS	Android Studio

CHAPTER 1

Introduction

The World Health Organization anticipated the people with visually impairment problems at global level, based on the various groundwork, it reveals that there are about 285 million humankind who agonize from vision impairment internationally, of which 246 million individuals have visual impairment and 39 million individuals are fully unsighted. In addition to that it predicted that the Eastern Mediterranean Region contains 12.6

One of the major complications fronted by individuals with vision problems is the helplessness to understand the paper currencies because of the indistinguishable paper texture and alike dimension among the special classes. Hence, the objective is to come up with a solution to help people deal with this problem and make blind human beings sense defensive and confident in the transactions involving paper money. There are two ways in Money identification studies area; Scanner-based and Camera based. Scanner-based totally assumes capturing the whole paper. These systems are suitable for the Machinery of money counters. While camera-based systems considers taking pictures of the paper by a camera which additionally captures an element of the paper. Majority of the related works in literature deal with the scanner-based type. For visual impairment use, it's intended to allow customers to capture any a part of the paper through their smart phone and let the gadget analyse it and inform the value of the note.

In this project, camera-primarily based Indian paper currency is trained to be diagnosed the use of very easy android application Utilities what makes the processing time is very brief with acceptable accuracy. The proposed system has the capability to process pictures clicked partly and below divergent lighting conditions.

1.1 Motivation

We discover human beings round us who're visually impaired i.E the humans can not see matters properly and the individuals who are blind making transactions at diverse stores but they're not able to recognize the currency notes they provide to others or they take from others.

Hence there are some instances that they'll receive some different notes rather than giving the right notes wantedly or unwantedly by way of the other people or there are cases that the visually impaired individual may additionally supply other notes instead of giving the proper notes to others. These might also reason loss for the visually impaired humans.

1.2 Problem Statement

Visually Impaired are the ones people who have imaginative and prescient impairment or vision loss. Problems confronted via visually impaired in acting daily sports are in exquisite number. They additionally face a number of problems in financial transactions. They are not able to identify the currency notes because of similar structure and length of the notes. There can be frauds going around him/her with out giving the appropriate foreign money word or the impaired individual can also give greater amount to other with out knowingly. The person might also face trouble in remembering the transactions he/she made.

1.3 Project Objectives

This Android Application makes use of text to speech conversion to present voice output of detected foreign money observe fee to the user. For forex detection, this application uses Azure custom vision API the use of ML class method to discover foreign money based on photos or paper the use of cell digicam.

This Currency Detector app facilitates visually impaired patients to hit upon and understand money. Using this utility blind humans will click photograph

of the word and Application tells the user with the aid of speech how much the forex note cost is.

CHAPTER 2

Literature Survey

A wide range of researchers have contributed to the development of forex reputation strategies in numerous methods. Because of the differences in capabilities between cash and bills (notes),researchers approach the popularity problem differently for every of them. We will move over earlier work finished in foreign money popularity strategies on this phase.

A wide wide variety of researchers have contributed to the improvement of foreign money reputation techniques in numerous methods. Because of the variations in functions among coins and bills (notes), researchers technique the recognition problem otherwise for each of them. We will move over earlier paintings accomplished in foreign money popularity strategies in this phase.

According to research in [1], authors have added an unsupervised set of rules for segmentation of synthetic aperture radar pix that's depend on fuzzy clustering approach to overcome the high time complexity of wealthy-performance clustering algorithms which analyze all pixels for photograph segmentation. Their set of rules selects a subset of key pixels supported with the aid of the rule of nearby extrema and performs segmentation on the ones.

Many of the foreign money reputation systems are proposed. In [5], the author acknowledges and classifies 4 one of a kind styles of currencies via pc vision. The usual Accuracy rate become ninety three.84

However, the above take a look at has no longer solved the problem that the accuracy of the ORB algorithm is decreased/reduced whilst the measurement of the foreign money photograph of an outsized external surroundings trade substantially. So, depend on the authentic ORB picture registration approach, Yanyan Qin et al. [8] combine the primary SIFT technique with the ORB approach and define the SIRB (SIFT and ORB) set of rules. SIRB solved the errors of ORB scale inconsistency whilst coping with the deserves of ORB in matching rapid.

2.1 A. Scanner-based systems

A scientist called Bawane proposed a device that scans whole paper for clarity. They took size as one of the three traits of currency notes same way to color and texture. The technique specified on this paper used for identifying currency notes from various countries. It also handles a foreign currency device which uses the ensemble neural network (ENN). All the neural networks in an ENN are professional through low correlation mastering. The purpose for the use of negative correlation learning is to ability the human beings in an ensemble on numerous components of enter patterns. ENN is getting used to select out new, old and noisy currencies. The Ensemble network is used for the separation of all styles of foreign forex. It reduces the probabilities of errors than a unmarried network.

Yaseri et al. Used Fourier-Mellin rework for invariant rotation, translation and scale of the whole image. The photo is segmented and Markovian traits of each phase had been used to layout a function vectors. These vectors are then fed into SVM classifier for currency popularity.This procedure is concerned with complete paper forex was implemented to both sides of the banknotes, and it is able to end result 98.7

Ahmed and Mirfa developed a gadget for Pakistani paper forex that might apprehend the forex note as it should be. In their paper, they have taken samples place of five great Pakistani paper currency notes (Rs. 10, 20, 50, a hundred, one thousand). They scanned standard 100 currency notes (whole currency paper), 20 from every sample of determined on vicinity for characteristic extraction of these pictures the use of a software program program.

The implemented device uses the specific characteristics of the money for popularity. Experimental results are supplied which display that this scheme can understand presently available 8 notes of Pakistan's Currency (Rs. 10, 20, 50, one hundred, 500, 1000 and many others.) correctly with an average accuracy of 98.57

2.2 B. Camera-based systems

Very few programs helps visible impaired peoples on this problem. Here we are able to check out some of mobile applications and gadgets which can be used for serving them to face by myself in various Responsibilities such as the ability to understand cash in various currencies.

Surya et al. has put forward a cellular efficacy designed with a layer of dismemberment to reduce the waste to a great extent. They used the visible BagofWords (BoW) technique. The technique used works good with noisy images clicked on smart phones. They used the gadget to Indian Rupee paper payments and reported 96.7

Anas devised a non-parametric design where a non-parametric model is procured by the mean of the lined up samples of a currency note. Then, the will coefficient is calculated between an undecided currency note and all models. at last, a keynote rating is used to assign a version to the undecided post. The devised technique is proven to be productively used to capture 3 kinds of Saudi currency notes with fault rate of 10

In , the authors came up with a machine currency with a reputation created using a botonical electronic clock. The applicable shapes are extracted from the image to go along with the Hungarian preflow to the camera using alterable morphometry and doorway shape filters. A two-tier catalog is used for separate patch layers (portrait, face value, brand) and votes are intermixed through a special decision set.

furthermore, a lot of applications are discovered in google play for forex not having any reference on their original health contributions. IDEAL Currency recognizer is an Android cell application that allows visually impaired people to be known of U.S. Greenbacks. It offers lot of advantages to make identify the notes less tangled. It captures frames the use of the tool digital camera and process them in short period to inform the person by using the recognized observe. Texttospeech (TTS) is used to tell the consumer the value of the paper word which provides good knowledge and integration between the consumer and the software. Automatic recognition of digital cameras is one of the

benefits of this software as it can change the brightness of digital cameras to suit common environments for well organized recognization of foreign bank notes. under different lighting conditions.

LookTel currency identifier is a mobile software that works on iOS to assist people detect and calculate their currency notes. It can assist with total blindness by notifying them using the identified amount of notice using TTS. In addition, it can help the visually impaired by presenting the number of messages to be recognized using a large font in the middle of the display.

It can also work with great languages including English, Spanish, French, and Japanese. It has some limitations close to the tool when it comes to taking pictures and works instantly in real time.

One of its advantages is that the user can log in and for the duration of the session the app can rely on the number of diagnosed currencies to come up with a sum when the consultation stops. He also developed to draw using Apple computers and calculators.

Generally, this application has some of beneficial offerings. Although it's far well matched with distinctive languages inclusive of English, French, Arabic and Spanish, it is complex within the interaction with one which can't see due to the fact its interface includes many icons and there is no voice instructions make the system less difficult.

Impatiently it can seize frames the usage of the tool digital camera. By extracting the feature from each note, it checks it with its pre-determined capabilities and based on similarity measures it could find the nearest word it is able to be. But as it makes use of shade capabilities, the brightness is a important hassle. It works in another way below specific brightness situations and mistakenly identify notes. This software has some of functions which includes usage of flash mild and modify camera awareness. These functions can be manually designed and therefore lacing pliability.

In this work, easy machine is put forward and examined in Android structures. The devised gadget considers capturing any a part of the currency note in a selected route. Basic machine mastering utilities are used to identify the foreign money cost the use of template matching.

2.3 Existing System

ROSHNI

The Indian Institute of Technology (IIT), Ropar, in Punjab, has developed an Android app Roshni that facilitates the visually impaired recognize old and new foreign money notes, using image processing and analytics. It utilises a deep getting to know framework, which in addition makes use of the patterns and capabilities embedded at the notes to distinguish and determine the forex denomination. 'MANI' (Mobile Aided Note Identifier)

The RBI Governor Shaktikanta Das released a cellular app in the year 2020 to assist visually-impaired people to recognize the value of forex notes.People can freely down load the app, called 'MANI' (Mobile Aided Note Identifier), from Android Playstore or iOS App Store. The software can test the foreign currency notes the use of the digicam of the mobile telephone. It also gives audio output in Hindi and English.

2.3.1 Limitations of Existing System

- The programs Roshni and MANI does now not provide any records maintainance.
- They do now not provide individual to character transaction quantity and its records.
- User Interface is not that clean to use through the customers.
- Both the programs makes use of buttons in their UI, consumer want to press the button, however it's far hard to press the button by using the user (visually impaired).
- Roshni app does now not locate currency notes correctly and the consumer wishes to show the identical notice to the digicam in extraordinary formats like front, returned, half of folds so one can hear the maximum repeated foreign money fee and that is the result. the opposite hand, MANI app has a higher accuracy in detecting the currency notes, but has the buttons in their UI.

CHAPTER 3

Software and Hardware Specifications

3.1 Software Requirements

3.1.1 Functional Requirements:

Currency Detection.

Date and Time smart History maintainance.

History of user currency detection. of Person to Person transaction forex detection.

3.1.2 Non-Functional Requirements:

Reliability.

Recoverability.

Portability (Android Versions).

Friendly environmental.

3.2 System Specification

3.2.1 Software Specification:

Android version 4.4 (kitkat) or above

3.2.2 Hardware Specification:

Quadcore processor or higher core processor

Minimum 1GB RAM

Minimum 1.3 MP camera

CHAPTER 4

Design

4.1 Use Case Diagram

In the Unified Modeling Language , a use case diagram can summarize the info of your device's end user and their relationship with the system. To create one, you can make use of fixed of reputed symbols and connectors. An productive use case diagram can assist your group speak and constitute:

A use case diagram doesn't involves into a whole lot of element—as an example, don't think it to version the order where the steps are done. Instead, a good use case diagram indicates a excessive-degree evaluate of the interaction between use instances, actors, and structures.

Unified Modeling Language is the modeling toolkit that you may use to construct your use case diagrams. Use instances are presented with a categorised oval form. Stick figures represents actors inside the procedure, and the actor's interaction inside the gadget is constructed with a line between the actor and use case. To indicate the machine boundary, draw a container around the use case.

User will detect the foreign money notes 10, 20, 50, one hundred, 2 hundred, 500, 2000 with the help of cellular camera. Application will offer ways of detection i.E person forex detection and character-character currency detection. And each the user foreign money detection records and individual-individual foreign money detection records are stored, displayed and read out.

The major purpose of a use case diagram is to depict the interacting elements of a device. It accumulates the gadget's requirement, which includes each internal in addition to external influences. It invokes people, use cases, and several matters that invoke the actors and factors liable for the implementation of use case diagrams. It represents how an entity from the external surroundings can interact with part of the system.

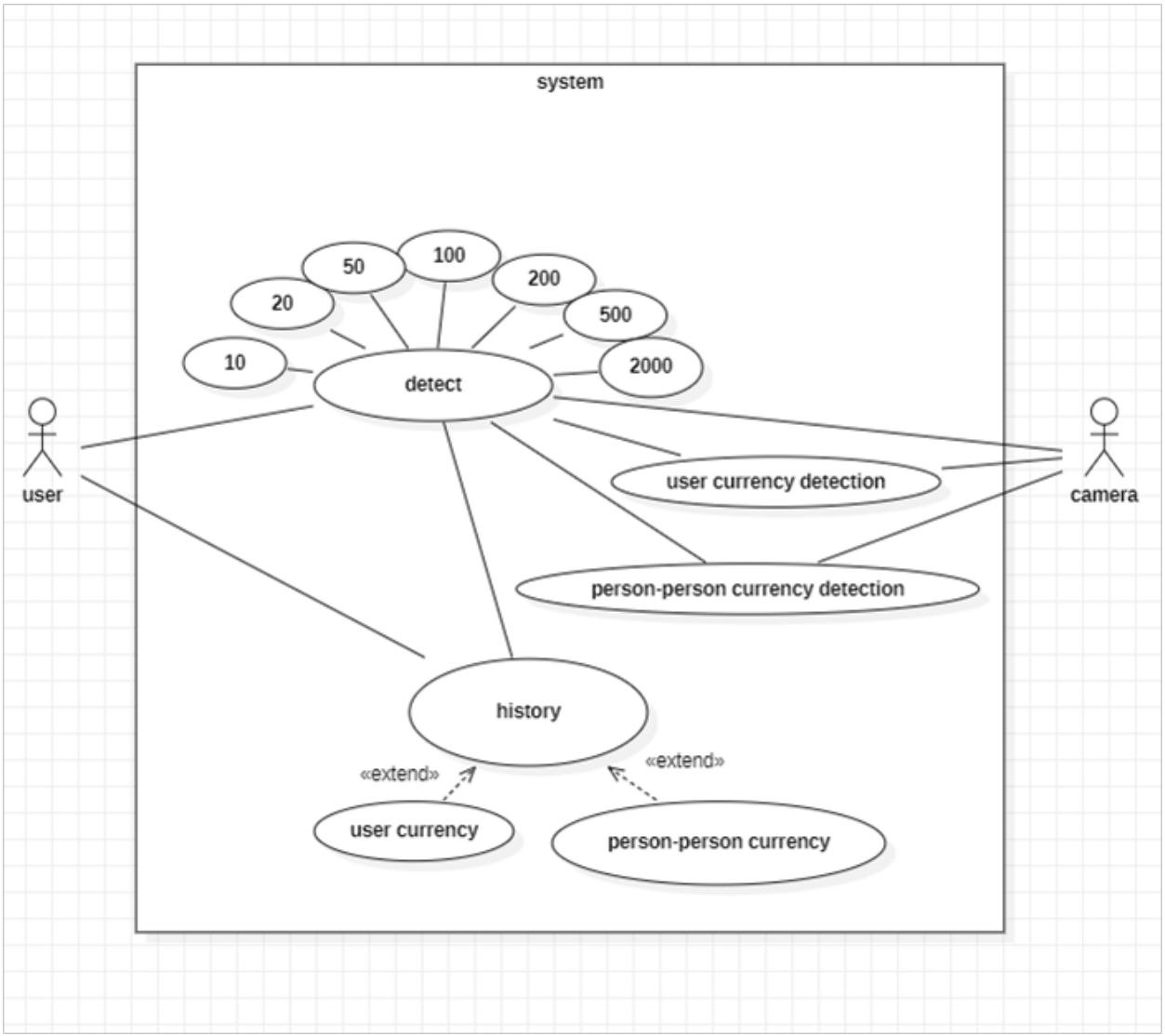


Figure 4.1: UseCase Diagram

4.2 Class Diagram

Class diagram represents the consistent view of an utility. Class diagram is used for constructing executable code of the software application but not best used for visualizing, and documenting extra features of a system however additionally.

Class diagram reports the attributes and operations of a category and limitations imposed on the machine. The class diagrams are extensively used inside the designing of object oriented system because of the feature that they are the only Unified Modeling Language diagrams, which may be connected at once with object-oriented languages.

Class diagram shows a set of lessons, interfaces, associations, collaborations,

and constraints. In this way it is sometimes referred to as a structural diagram.

The reason of class diagram is to design the stable view of an application or software. Class diagrams are the most effective diagrams which can be at once mapped with object-orientated languages and thus extensively used on the time of creation.

UML diagrams, such as activity diagrams and series diagrams, can only show the collection flow of an application, but the splendor diagram is unique. This is the most well-known UML diagram in the programming community.

Class diagrams are the most well-known Unified Modeling Language diagrams used to create software program applications. Learning the process of drawing a class diagram is very important.

There are some places to remember when drawing a class diagram, but here you can view top-class diagrams.

The class diagram is basically a graphical representation of the stable view of the machine, showing a unique aspect of the application. A series of class diagrams make up the entire device.

Class diagrams are stable diagrams and are commonly used to version stable views of devices. Stable views tells the vocabulary of the system.

Class diagrams are also considered as inspiration for aspect and layout diagrams. Class diagrams are not optimally used to visualize static views of the system, but are often used to assemble executable code for device pre-engineering and counter-engineering.

In general, UML diagrams cannot be easily mapped to object-oriented programming languages, with the exception of magnificence diagrams.

Class diagram genuinely indicates the mapping with item-oriented languages together with Java, C++, and so on. From realistic revel in, class diagram is generally used for creation reason.

The class diagram actually shows the mapping to object-oriented languages in addition to Java, C ++, and so on. For real time reasons, class diagrams are commonly used for construction.

Camera class covers the method cameraOpen(), cameraPreview(), cameraClosed(), cameraResume(), takePicture(), updatePreview() which might be digicam particular methods to manipulate the camera. User forex detection

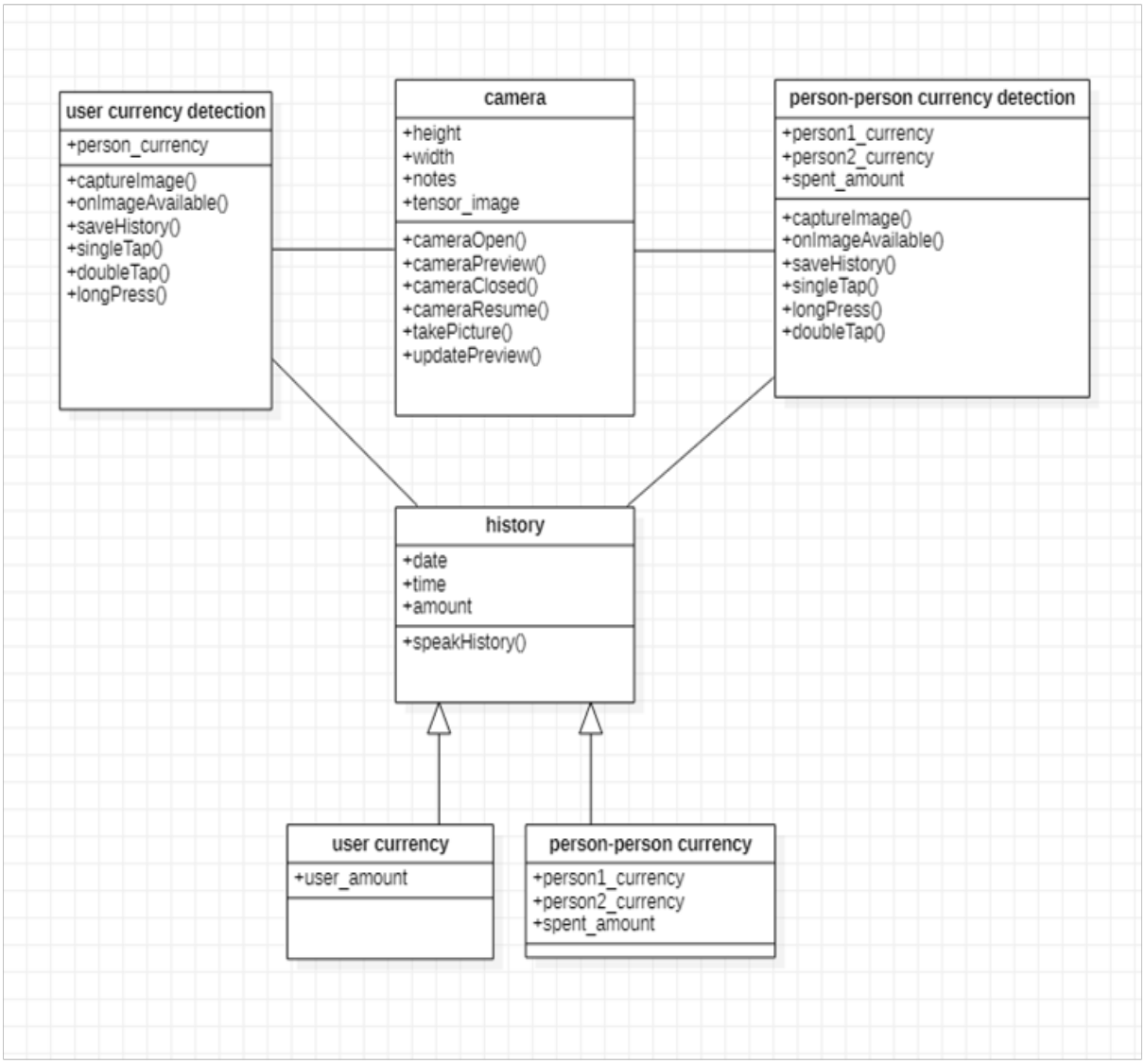


Figure 4.2: Class Diagram

magnificence and character-character currency detection magnificence have a minor difference in their implementations. Person-person currency detection magnificence can have humans forex calculation whereas person currency detection elegance will most effective calculate the user foreign money. History sub classes also are identical because the detection training.

4.3 Activity Diagram

Generally activity diagrams are to show slide operations on your device and review the steps involved in running a use case. Therefore, you basically use entertainment diagrams to visually represent your workflow. The entertainment map focuses on the wind conditions and the sequence of situations in which

they occur. Explain or explain why you use entertainment charts on specific occasions. UML basically models three types of diagrams: shape diagrams, interaction diagrams, and motion diagrams.

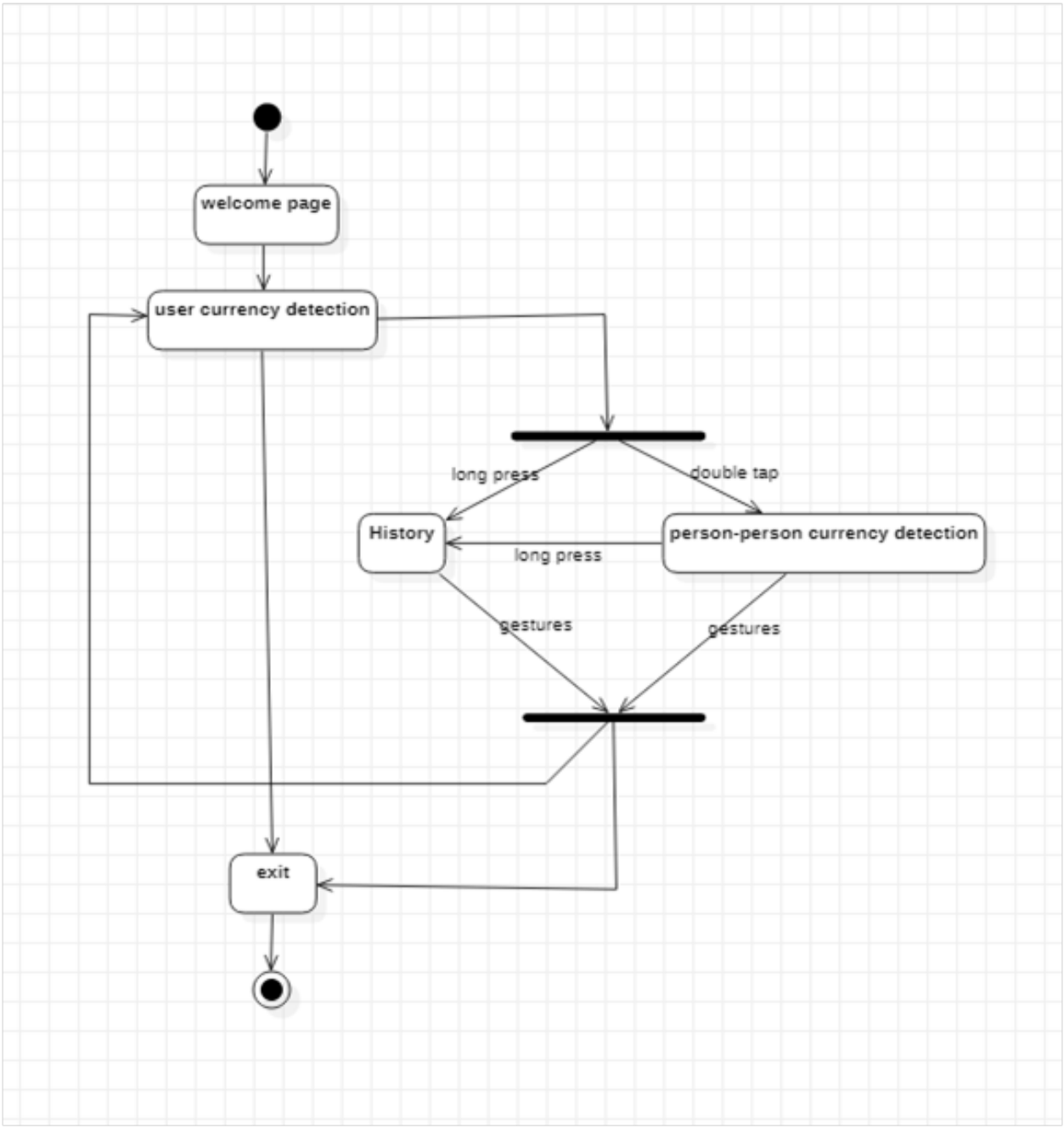


Figure 4.3: Activity Diagram

The entertainment map is an action map. That is, it shows the behavior of the device. The interest diagram shows the management flow from the start point to the end element and shows the various decision paths that exist at the same time as the hobby stops. Interest charts can be used to represent

any sequential and parallel processing activity. These are primarily used in business and method modeling related to mapping the dynamic aspects of a machine. Interest rate charts are very similar to flowcharts. Now let's see if there is anything special about entertainment diagrams or flowcharts.

At first the application will open with a welcome web page after which consumer foreign money detection page could be open wherein the user foreign money notes are to detected and double taping here will flow to the person-man or woman forex detection page wherein forex of both humans worried in transaction are to be detected. Long press from those two pages will take you to the history web page. You can go out the App from any page the use of a specific gesture.

4.4 Architecture Diagram

An architecture diagram is a optic depiction of all the elements that make up element, or all, of a device. Above all, it allows the engineers, designers, stakeholders — and anybody else concerned within the assignment — apprehend a machine or app's layout.

Think of it as being a chunk like a blueprint to a building: You can see the component as an entire, in addition to special types of interior perspectives, and things like pipes, partitions, floorplans, and so forth.

Diagrams make it less difficult to soak up records. They also assist with comprehension and bear in mind. Here are some greater advantages:

This diagram offers a top-degree view of a software's structure. To complex, it normally consists of numerous additives that engage with each other and how the software interacts with outside databases and servers. It's beneficial for explaining software to clients and stakeholders; and assessing the impact of adding new capabilities or upgrading, changing, or merging present programs.

These diagrams visualize community limitations, at the side of the software program, nodes, and processors that make up the gadget. They also can help you understand how extraordinary components talk with every different. Not only that, however additionally they come up with a top level view of the bodily hardware in the machine.

Use a deployment architecture diagram to assist plan strategic applications and carrier upgrades so your servers can handle extra requests because the business grows.

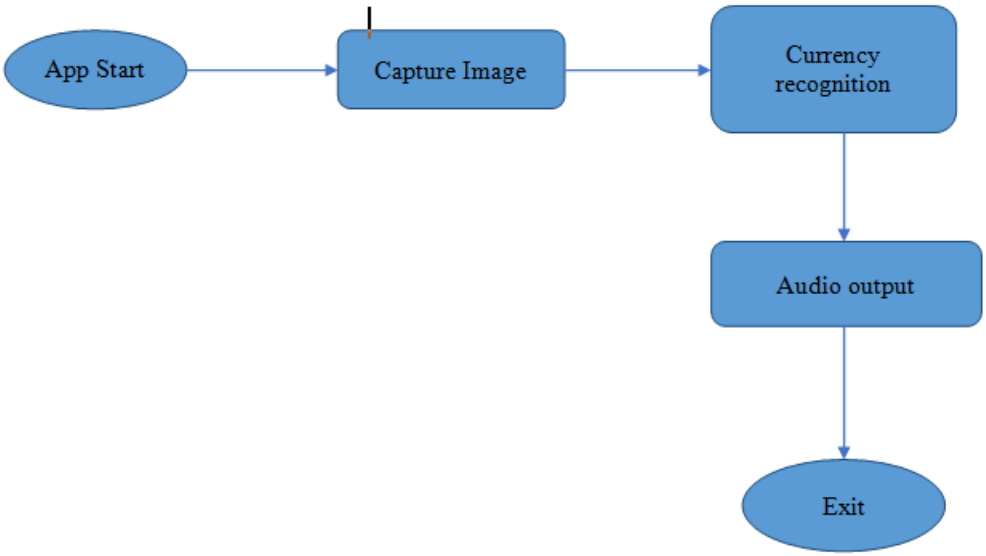


Figure 4.4: Architecture Diagram

4.5 Technology Description

Figure 4.5: Android Studio

Android Studio provides many extraordinary capabilities that beautify productiveness while building Android apps, together with a blended surroundings wherein it is easy to develop for all Android gadgets, practice Changes to push

code and aid adjustments to the running app with out restarting the app, a flexible Gradle-primarily based construct machine, a fast and feature-wealthy emulator, GitHub and Code template integration to can help you increase common app capabilities and import pattern code, big trying out tools and frameworks, C++ and NDK help, and many extra.

Based on IntelliJ IDEA, this IDE provides quick code completion and immediate workflow evaluation. There are certain features of Android Studio, such as code push for customization and an incredible code editor for streamlined coding output.

With Android Studio, developers can push code and insert customizations instantly by allowing for easy customization without having to restart the app completely. This guarantees great flexibility in adjusting small apps, even if the app is still walking.

Customize your laptop's latest imaginative predictive photo ratings and integrate accurate domain names with Custom Vision, part of Azure Cognitive Services. Create smooth shopper reports, optimize production techniques, and speed up virtual advertising campaigns. System mastering know-how is not required.

Android Studio's intuitive code editor is a key feature that offers one of the main benefits of Android Studio, which comprises quick coding. At the same time, it provides state-of-the-art refactoring, completion code, and code analysis.

Android Studio has an emulator that allows you to launch your entire app faster than the actual tools. The emulator allows you to test your app on multiple devices such as tablets, phones, Android Wear, Android TV, and simulates many distinctive hardware features such as GPS, motion sensors, accelerometers, and more.

Android Studio developers can ensure that the overall quality of coding is maintained while working on missions at the same time.During the app challenge it also provides excellent coding. Android Studio guarantees optimal output for your app project only with all the satisfactory programming practices under your control.

Android Studio provides many review tools and frameworks for reviewing

apps using various useful UI review tools. The benefits of Android Studio are also evidenced by excellent review devices and frameworks of all kinds for all purposes. Tests run on these devices can be run using the Firebase test lab in a real gadget, emulator, or stable integrated environment.

Android Studio comes with a Firebase wizard that allows you to connect any app to your Firebase server, except for adding many important services such as app notifications, authentication, and analytics,. Android Studio makes it easy to combine your app with the Google Cloud platform. The Android App Improvement Agency in India, which builds enterprise apps, considers this cloud help to be very important.

Android Studio to work with XML files provides a visual drag-and-drop editor device . This makes it easy to create entirely different formats of app. Built in sync with the ConstraintLayout API, Android Studio's Layout Editor allows you to develop layouts that can adapt to your display size. This maintains an optimized layout approach based on the requirements.

Android Studio provides excellent version control devices such as Subversion and GitHub to give you more control over your team and make changes at any time. Robust repositories of accessible devices and good installed coding techniques improve collaboration and teamwork .

Android Studio provides a powerful and integrated platform for creating apps for all types of Android devices. You can also use this device to fix problems and provide the device with optimized enjoyment. Android offers the largest tool ecosystem, and thanks to Android Studio, it pushes the limits of tool-optimized improvement attributes.

Android Studio comes with some sample tasks and your own code templates to guide developers with a proven strategies for various app tasks. You can also use the sample code to search and search the Internet while using the sample code. In addition, Android Studio allows you to integrate some of the most useful apps on your system, such as GitHub.

Android Studio makes it even easier to display your app in search results by displaying a URL form optimized for your app code. App makers can also upload these URL attributes directly to the AndroidManifest.Xml document which will improve GoogleAppIndexing.

Android Studio provides a dedicated GPU debugger that uses rigorous images to develop OpenGL ES games while allowing game developers to make in-depth body-specific assessments. This device can solve all kinds of video game image rendering issues. It also improves the performance of all great games.

The Android Studio translation editor device provides a single, powerful and integrated view of each translated resource. This makes it easy to add or modify translations. The same tools can also assist you find missing translations. The device additionally offers a separate hyperlink to the service for translations of ordering.

From the beginning, Android Studio has been the go-to IDE for app developers to perform all their app tasks on Android. Android Studio has evolved to further help Android app developers develop, test, and deploy highly technical apps.

4.6 Microsoft Azure Custom Vision API

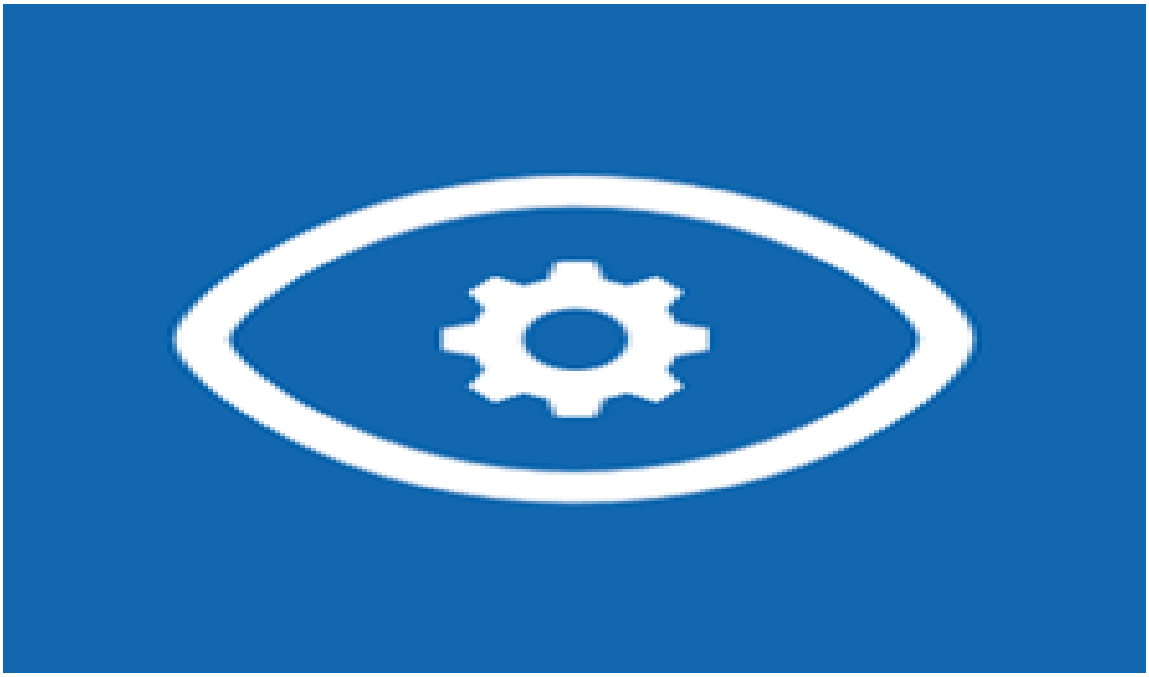


Figure 4.6: Microsoft Azure Custom Vision API

Customise and embed trendy computer vision photo evaluation for domains consisting the Custom Vision, Cognitive Services part Azure. create

friction less client stories, optimize production methods, boost up digital marketing campaigns, and greater. No system gaining knowledge of understanding required.

The primary concept behind Custom Vision is to take a pre-built picture reputation model supplied by using Azure, and customize it to your needs with the aid of providing a set of snap shots with which to replace it. All version schooling and prediction is performed in the cloud, so that you don't need a powerful gadget. Similarly, because you are beginning with a model that has already been trained, you don't want a completely massive dataset or long schooling instances to acquire excellent predictions (preferably). This vignette walks you through the technique of making and deploying a Custom Vision predictive provider.

You can create the Custom Vision sources the usage of the Azure RMR framework for interacting with Resource Manager. Note that Custom Vision requires as a minimum two sources to be created: one for schooling, and one for prediction. The to be had provider ranges for Custom Vision are F0 (unfastened, limited to 2 initiatives for training and 10k transactions/month for prediction).

Custom Vision is organised hierarchically. At the top stage, we've a mission, which represents the data and model for a selected challenge. Within a project, we have one or more iterations of the version, built on exceptional sets of education snap shots. Each generation in a project is impartial: you could create (teach) an generation, set up it, and delete it with out affecting different iterations.

Cognitive services offer several features depending on the requirement. Popular versions of computer vision videos and images are a good place to start if you don't need to create your own model.

Start schooling your pc imaginative and prescient model by using definitely importing and labeling some photographs.Models test these themselves and usually increases accuracy via loop comments when uploading photos.

The Custom Vision carrier makes use of a machine gaining knowledge of set of rules to research photos. You, the developer, put up businesses of photographs which have and don't have the characteristics in query. You label

the pix your self at the time of submission. Then the set of rules trains to this statistics and calculates its very own accuracy by using testing itself on the ones same images. Once you've trained the algorithm, you may check, retrain, and ultimately use it for your image recognition app to categorise snap shots. You can also export the version itself for offline use.

The Custom Vision service is optimized to quickly detect large differences between images, so you can start prototyping versions with a small chunks of data. 50 photographs in line with label are usually an amazing begin. However, the service isn't greatest for detecting diffused differences in pics (as an instance, detecting minor cracks or dents in nice warranty eventualities).

Additionally, you may choose from numerous versions of the Custom Vision algorithm which are optimized for photos with certain concern cloth—for instance, landmarks or retail items. See Select a domain for greater statistics.

This Service is to be had as a set of various native SDKs as well as through an internet-primarily depending interface at portal. You can create, check, and train a version thru either interface or use both collectively.

Azure Custom Vision is an picture recognition provider that lets you construct, install, and enhance your personal image identifier fashions. An picture identifier applies labels to pix, in step with their detected visible traits. Each label represents a classifications or gadgets. It is not like the Computer Vision service, this service allows you to specify your own labels and teach custom fashions to apply on them.

A consumer-friendly interface walks you thru growing and deploying custom laptop models of imaginative and prescient.

4.7 TensorFlow Lite

TensorFlow Lite is basically a set of tools that enable on-device machines to gain that knowledge by allowing developers to run models on embedded, mobile, and IoT devices. The main traits of TensorFlow Lite are setup for research on your device, with importance on connectivity, length,latency, privacy, and power consumption. The framework is created to offer knowledge for a various platforms, including microcontrollers Android and iOS gadgets,

Figure 4.7: TensorFlow Lite

and embedded Linux. Tensor Flow also contains built-in assistance for a lot of languages.In addition to Cpp, Java, Swift and Python, with hardware rise and model optimization for whole best performance. Many structures offer end-to-end examples of daily gadget learning works for example object recognition, classification of images, estimation of pose,answering questions, and various text types.

A TensorFlow Lite model is presented in a green transportable layout sometimes reffered to as FlatBuffers. This provides numerous advantages over TensorFlow's protocol buffer version format with addition of minimized length and quicker inference that allows TensorFlow Lite to execute correctly on devices with limited compute and sources.

A TensorFlow Lite model can also encompass metadata which contains human-readable version description and system-readable records for automated generation of pre- and publish-processing pipelines for the period of on-device inference. Referring to Add metadata can give us greater details.

TensorFlow Lite is a robust device that transforms and setups TensorFlow models so that they can run on mobile and facet devices. Presently it

is running on over 4 billion devices. With TensorFlow 2.X, you can use tf.Keras to train your version, just convert the version to .Tflite and install it. Alternatively, you can download the pre-trained TensorFlow Lite model from the Zoo version.

Preetly supported on Android and iOS via the Cpp API, including the Java wrapper for Android developers. Additionally on Android devices that support it, the interpreter can also use the Android Neural Networks API for hardware rise. To go in other way, the interpreter runs using the CPU by default. Currently designed for educational purposes only. Instead, tell the version on a more powerful device, then convert that version to .TFLITE format and load miles from there into the mobile interpreter. **How does Tensorflow Lite(TF Lite) work?**

Select and Train a Model

Convert the Model using Converter

Tensorflow Lite Converter converts a Tensorflow model to Tensorflow Lite flat buffer file(.tflite).

CHAPTER 5

Implementation and Testing

This Android Application uses textual content to speech conversion to give voice output of detected foreign money be aware fee to the user. For forex detection, this software makes use of Azure custom vision API the usage of ML class technique to come across currency primarily based on pictures or paper using cellular camera.

This Currency Detector app helps visually impaired sufferers to locate and understand money. Using this application blind people will click on photo of the be aware and Application tells the person via speech how a great deal the currency word price is.

5.1 Implementation:

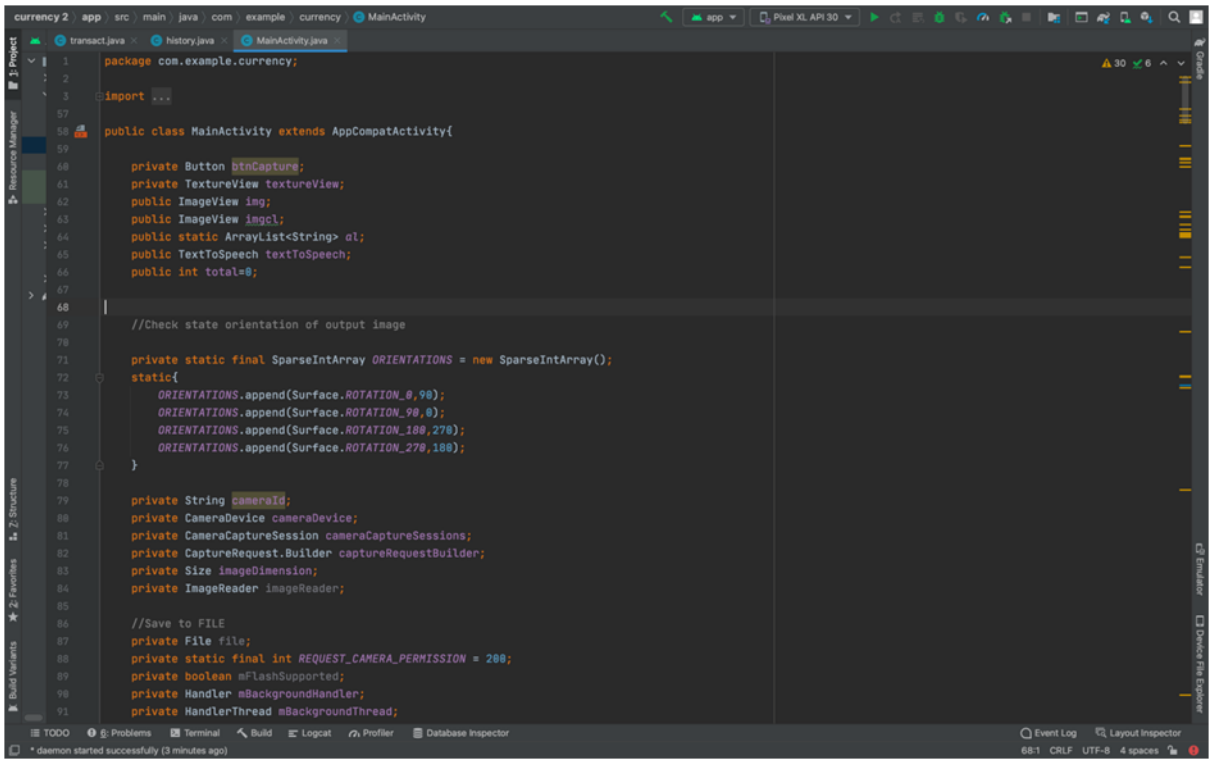


Figure 5.1: importing libraries

Importing the required libraries and declaring the required views and variables.

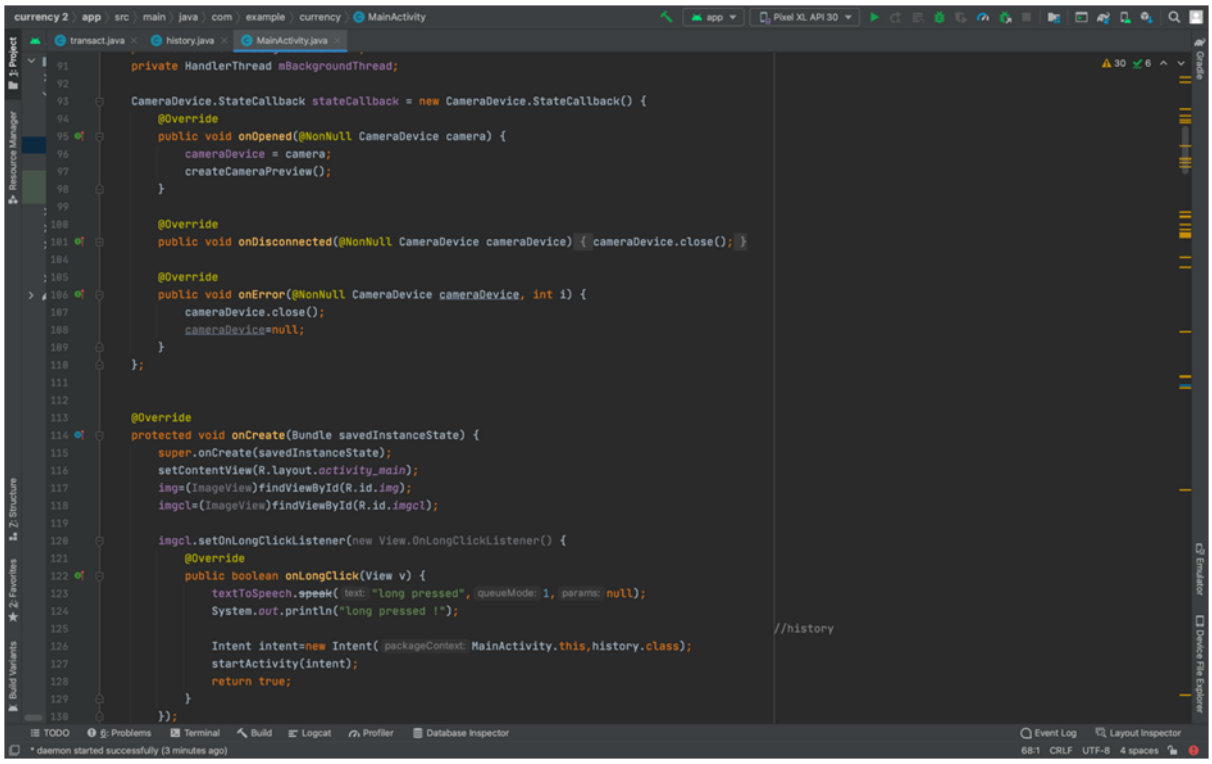


Figure 5.2: importing libraries

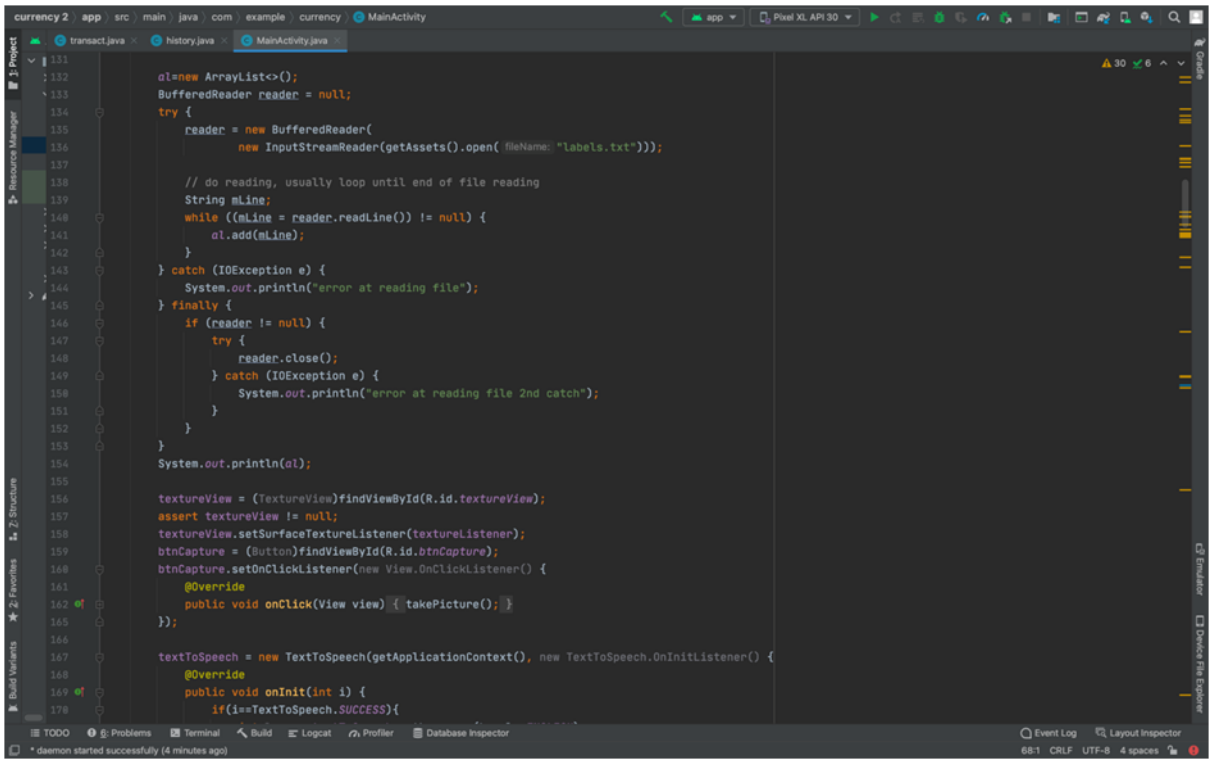
onCreate() function will be invoked when ever the activity is started. onOpened(), onDisconnected(), onError() are implemented to handle the camera actions.

5.2 Testing:

Every screenshot shows a toast message in it about its instructions/detected currency value.

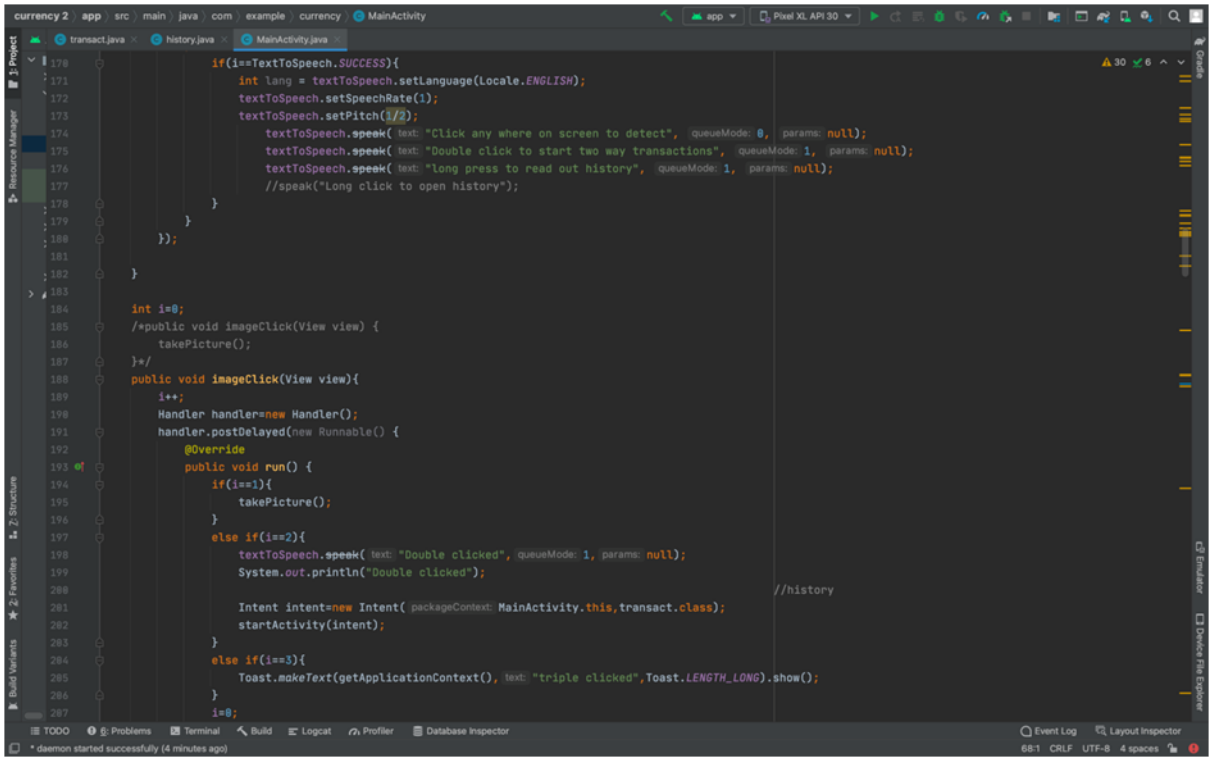
Instructions after starting the Application

5.2.1 Currency detection :



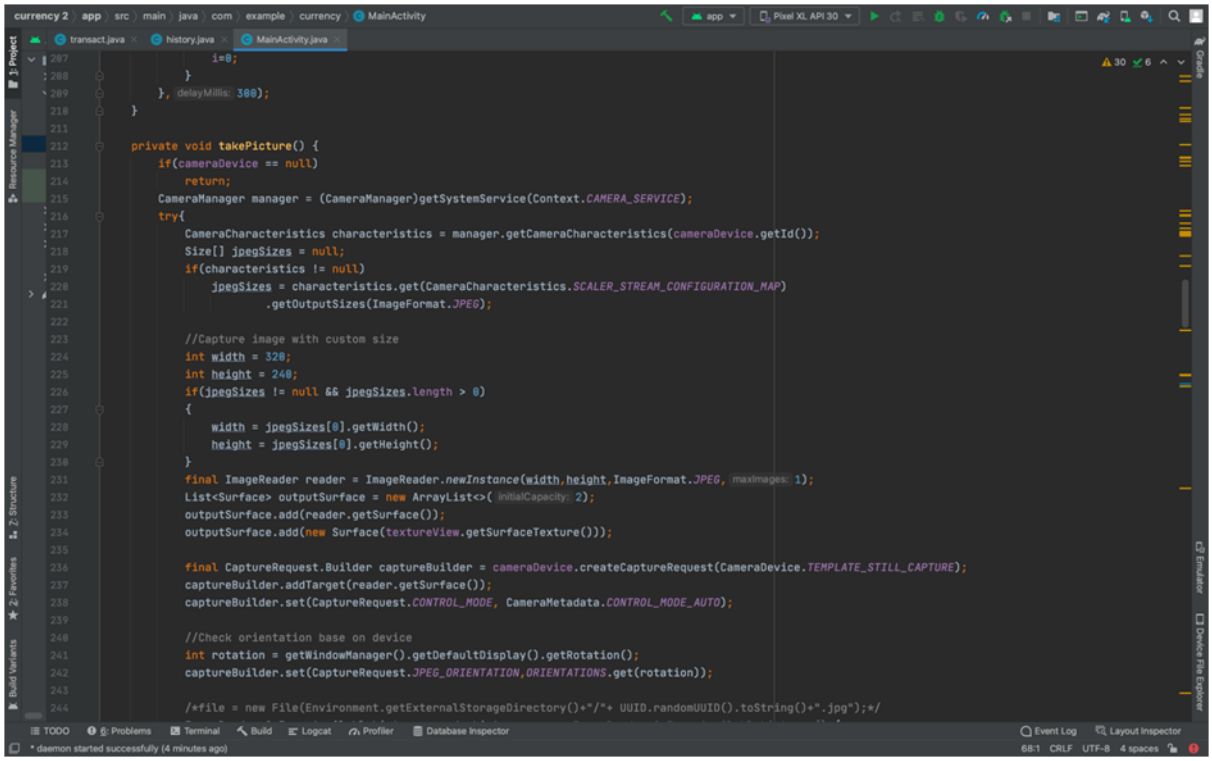
```
al=new ArrayList<>();
BufferedReader reader = null;
try {
    reader = new BufferedReader(
            new InputStreamReader(getAssets().open( fileName: "labels.txt")));

    // do reading, usually loop until end of file reading
    String mLine;
    while ((mLine = reader.readLine()) != null) {
        al.add(mLine);
    }
} catch (IOException e) {
    System.out.println("error at reading file");
} finally {
    if (reader != null) {
        try {
            reader.close();
        } catch (IOException e) {
            System.out.println("error at reading file 2nd catch");
        }
    }
}
System.out.println(al);

textureView = (TextureView)findViewById(R.id.textureView);
assert textureView != null;
textureView.setSurfaceTextureListener(textureListener);
btnCapture = (Button)findViewById(R.id.btnCapture);
btnCapture.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { takePicture(); }
});

textToSpeech = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int i) {
        if(i==TextToSpeech.SUCCESS){
```

Figure 5.3: ArrayList is used to store the labels, Texture view is initialized, tapped upon which takePicture() will be called. Text to speech is initialised here.



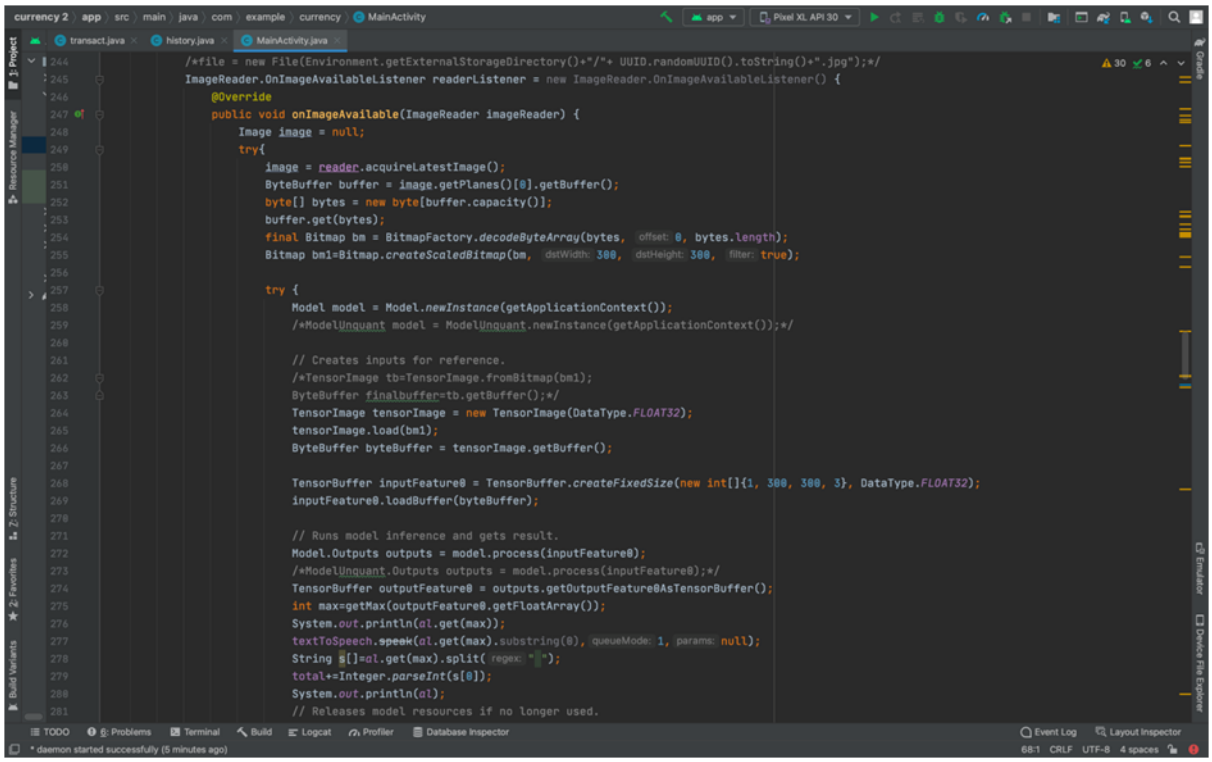
```
            if(i==TextToSpeech.SUCCESS){
                int lang = textToSpeech.setLanguage(Locale.ENGLISH);
                textToSpeech.setSpeechRate(1);
                textToSpeech.setPitch(1/2);
                    textToSpeech.speak( text: "Click any where on screen to detect",  queueMode: 0,  params: null);
                    textToSpeech.speak( text: "Double click to start two way transactions",  queueMode: 1,  params: null);
                    textToSpeech.speak( text: "long press to read out history",  queueMode: 1,  params: null);
                    //speak("Long click to open history");
            }
        }
    });

}

int i=0;
/*public void imageClick(View view) {
    takePicture();
}*/
public void imageClick(View view){
    i++;
    Handler handler=new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            if(i==1){
                takePicture();
            }
            else if(i==2){
                textToSpeech.speak( text: "Double clicked", queueMode: 1, params: null);
                System.out.println("Double clicked");
                                                                    //history
                Intent intent=new Intent( packageContext: MainActivity.this,transact.class);
                startActivity(intent);
            }
            else if(i==3){
                Toast.makeText(getApplicationContext(), text: "triple clicked",Toast.LENGTH_LONG).show();
            }
            i=0;
```

Figure 5.4: imageClick() is onClick method of the image view.



```
                i=0;
            }
        }, delayMillis: 300);
    }

    private void takePicture() {
        if(cameraDevice == null)
            return;
        CameraManager manager = (CameraManager)getSystemService(Context.CAMERA_SERVICE);
        try{
            CameraCharacteristics characteristics = manager.getCameraCharacteristics(cameraDevice.getId());
            Size[] jpegSizes = null;
            if(characteristics != null)
                jpegSizes = characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP)
                        .getOutputSizes(ImageFormat.JPEG);

            //Capture image with custom size
            int width = 320;
            int height = 240;
            if(jpegSizes != null && jpegSizes.length > 0)
            {
                width = jpegSizes[0].getWidth();
                height = jpegSizes[0].getHeight();
            }
            final ImageReader reader = ImageReader.newInstance(width,height,ImageFormat.JPEG, maxImages: 1);
            List<Surface> outputSurface = new ArrayList<>( initialCapacity: 2);
            outputSurface.add(reader.getSurface());
            outputSurface.add(new Surface(textureView.getSurfaceTexture()));

            final CaptureRequest.Builder captureBuilder = cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_STILL_CAPTURE);
            captureBuilder.addTarget(reader.getSurface());
            captureBuilder.set(CaptureRequest.CONTROL_MODE, CameraMetadata.CONTROL_MODE_AUTO);

            //Check orientation base on device
            int rotation = getWindowManager().getDefaultDisplay().getRotation();
            captureBuilder.set(CaptureRequest.JPEG_ORIENTATION,ORIENTATIONS.get(rotation));

            /*file = new File(Environment.getExternalStorageDirectory()+"/"+ UUID.randomUUID().toString()+".jpg");*/
```

Figure 5.5: TakePicture() will capture the image and resizes it.



```
            /*file = new File(Environment.getExternalStorageDirectory()+"/"+ UUID.randomUUID().toString()+".jpg");*/
            ImageReader.OnImageAvailableListener readerListener = new ImageReader.OnImageAvailableListener() {
                @Override
                public void onImageAvailable(ImageReader imageReader) {
                    Image image = null;
                    try{
                        image = reader.acquireLatestImage();
                        ByteBuffer buffer = image.getPlanes()[0].getBuffer();
                        byte[] bytes = new byte[buffer.capacity()];
                        buffer.get(bytes);
                        final Bitmap bm = BitmapFactory.decodeByteArray(bytes, offset: 0, bytes.length);
                        Bitmap bm1=Bitmap.createScaledBitmap(bm, dstWidth: 300, dstHeight: 300, filter: true);

                        try {
                            Model model = Model.newInstance(getApplicationContext());
                            /*ModelUnquant model = ModelUnquant.newInstance(getApplicationContext());*/

                            // Creates inputs for reference.
                            /*TensorImage tb=TensorImage.fromBitmap(bm1);
                            ByteBuffer finalbuffer=tb.getBuffer();*/
                            TensorImage tensorImage = new TensorImage(DataType.FLOAT32);
                            tensorImage.load(bm1);
                            ByteBuffer byteBuffer = tensorImage.getBuffer();

                            TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 300, 300, 3}, DataType.FLOAT32);
                            inputFeature0.loadBuffer(byteBuffer);

                            // Runs model inference and gets result.
                            Model.Outputs outputs = model.process(inputFeature0);
                            /*ModelUnquant.Outputs outputs = model.process(inputFeature0);*/
                            TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();
                            int max=getMax(outputFeature0.getFloatArray());
                            System.out.println(al.get(max));
                            textToSpeech.speak(al.get(max).substring(0), queueMode: 1, params: null);
                            String s[]=al.get(max).split( regex: " ");
                            total+=Integer.parseInt(s[0]);
                            System.out.println(al);
                            // Releases model resources if no longer used.
```

Figure 5.6: onImageAvailable() method will be executed after capturing the picture.

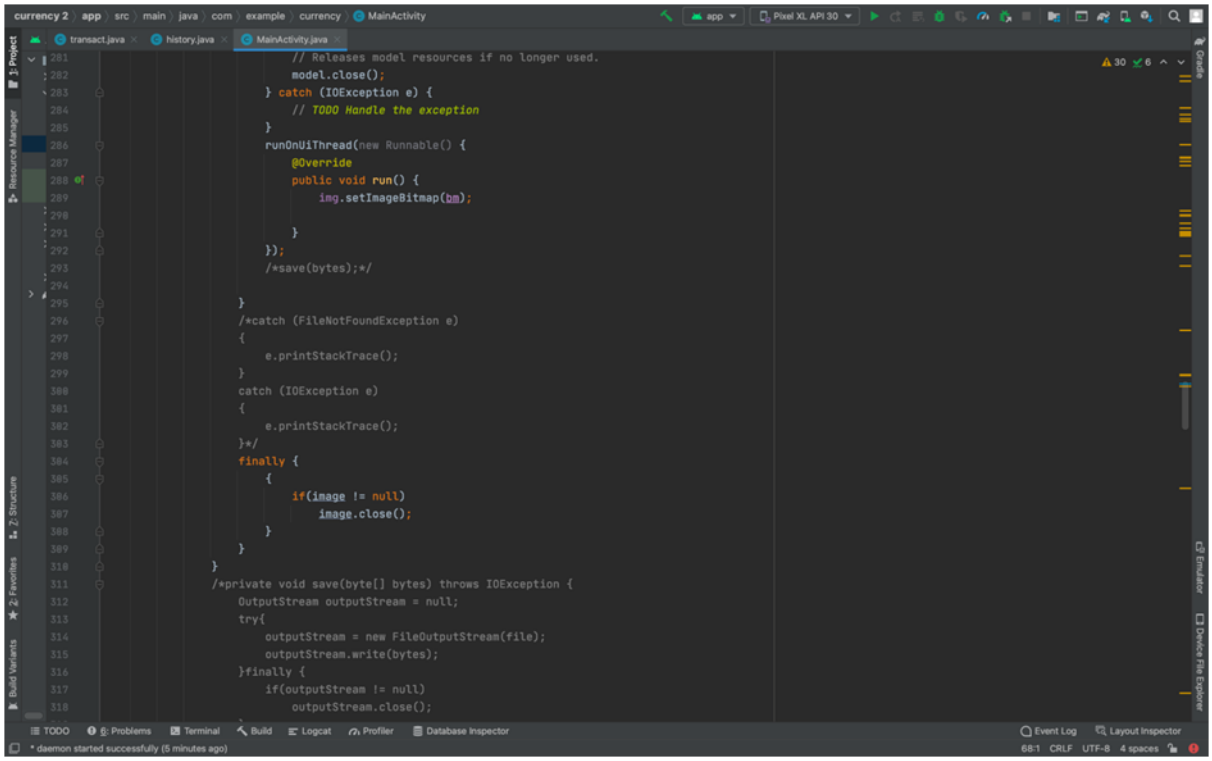


Figure 5.7: coding

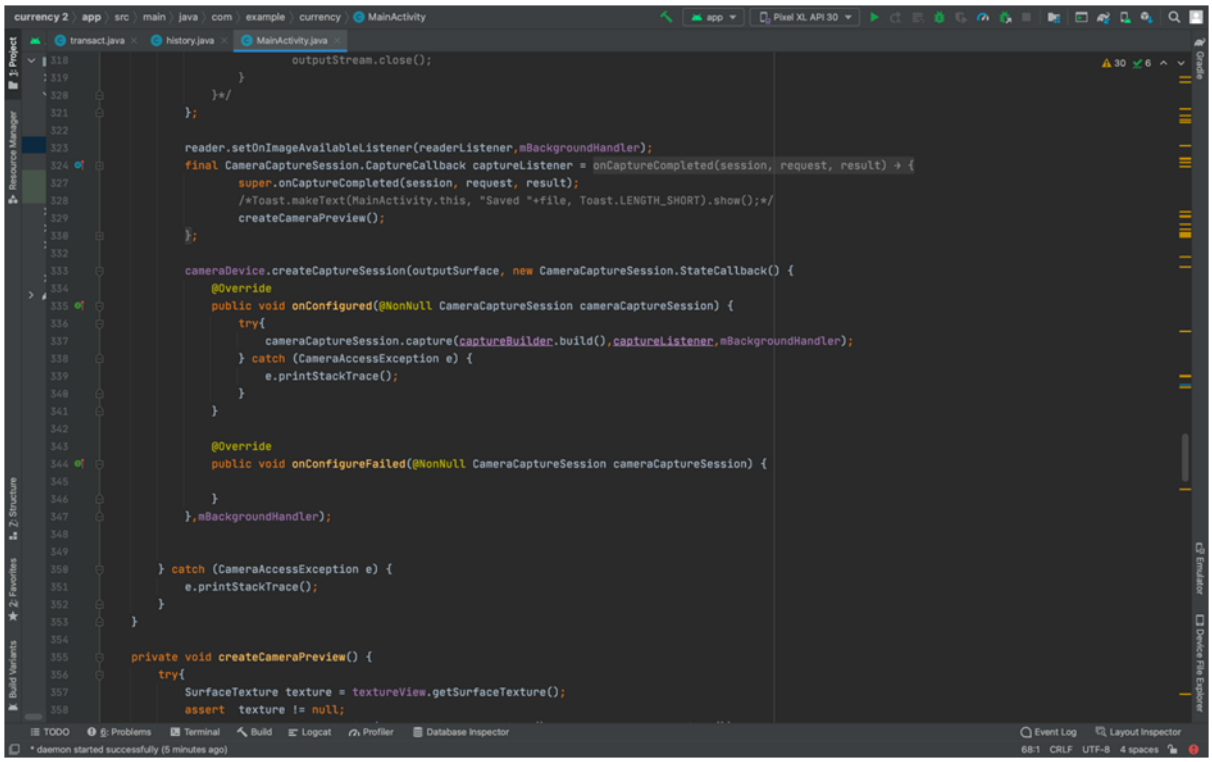
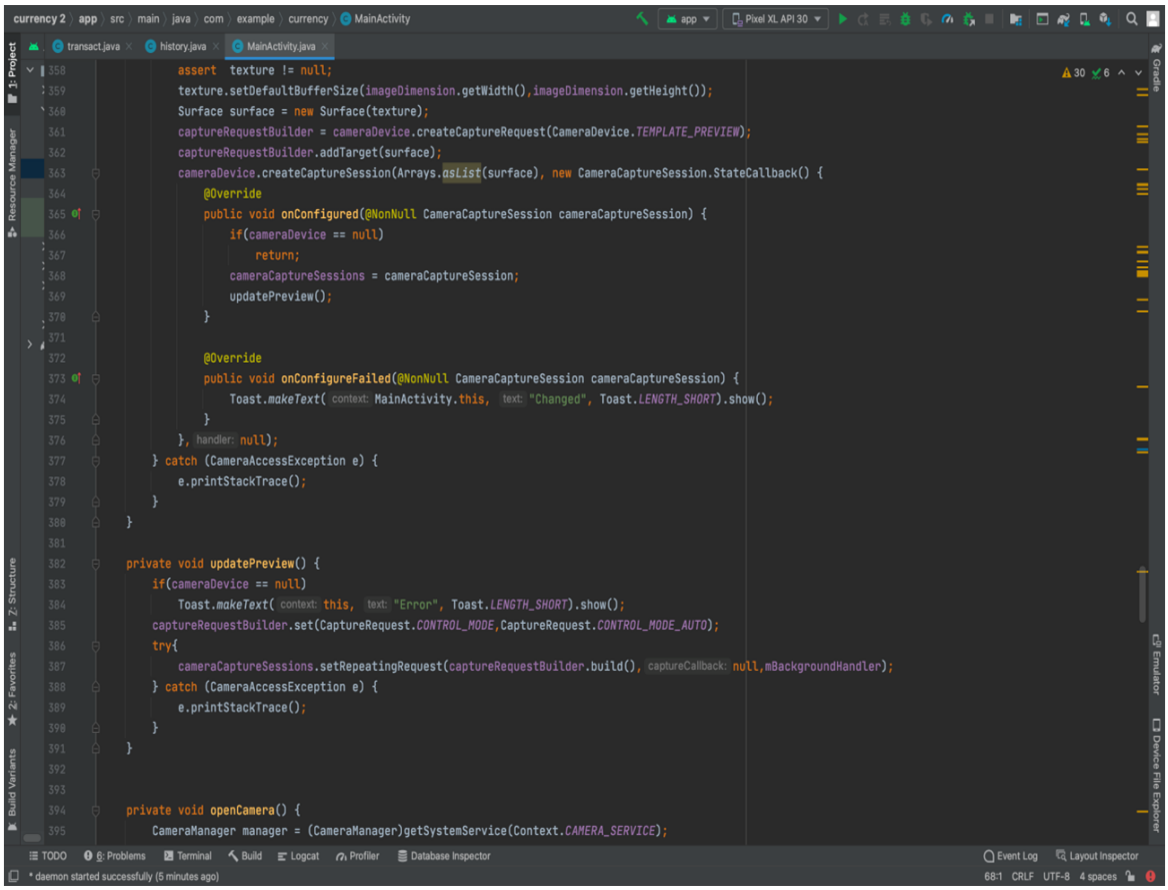
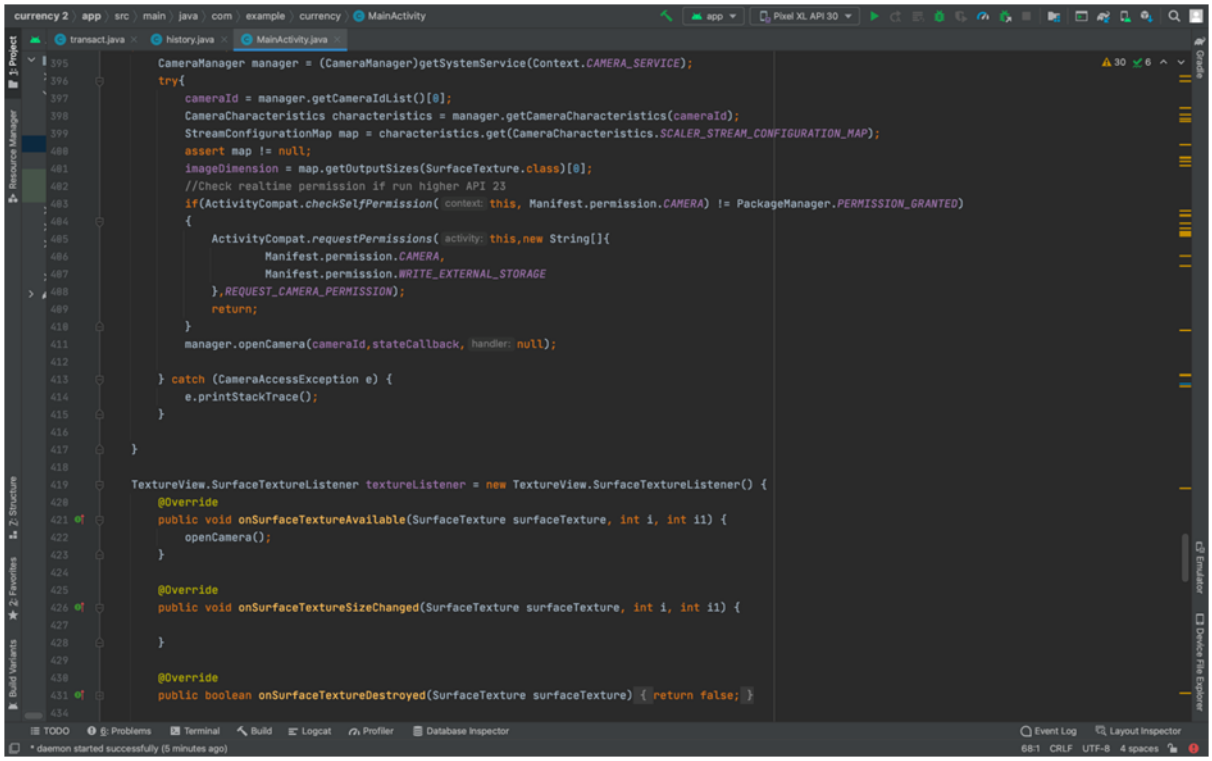


Figure 5.8: createCaptureSession() method is provided by cameraDevice object which will provide session to use camera.



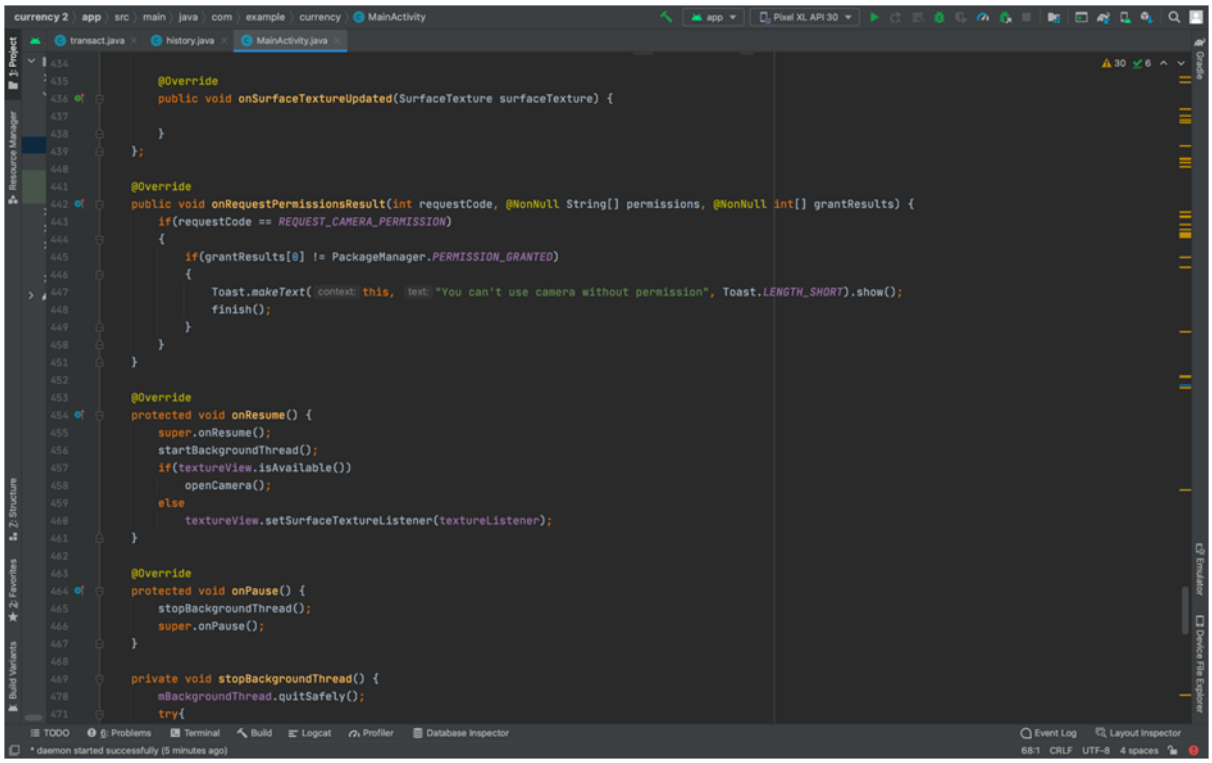
```
            assert  texture != null;
            texture.setDefaultBufferSize(imageDimension.getWidth(),imageDimension.getHeight());
            Surface surface = new Surface(texture);
            captureRequestBuilder = cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_PREVIEW);
            captureRequestBuilder.addTarget(surface);
            cameraDevice.createCaptureSession(Arrays.asList(surface), new CameraCaptureSession.StateCallback() {
                @Override
                public void onConfigured(@NonNull CameraCaptureSession cameraCaptureSession) {
                    if(cameraDevice == null)
                        return;
                    cameraCaptureSessions = cameraCaptureSession;
                    updatePreview();
                }

                @Override
                public void onConfigureFailed(@NonNull CameraCaptureSession cameraCaptureSession) {
                    Toast.makeText( context: MainActivity.this,  text: "Changed", Toast.LENGTH_SHORT).show();
                }
            }, handler: null);
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }
    }

    private void updatePreview() {
        if(cameraDevice == null)
            Toast.makeText( context: this,  text: "Error", Toast.LENGTH_SHORT).show();
        captureRequestBuilder.set(CaptureRequest.CONTROL_MODE,CaptureRequest.CONTROL_MODE_AUTO);
        try{
            cameraCaptureSessions.setRepeatingRequest(captureRequestBuilder.build(), captureCallback: null,mBackgroundHandler);
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }
    }


    private void openCamera() {
        CameraManager manager = (CameraManager)getSystemService(Context.CAMERA_SERVICE);
```

Figure 5.9: updatePreview() will update the preview of camera after every takePicture().



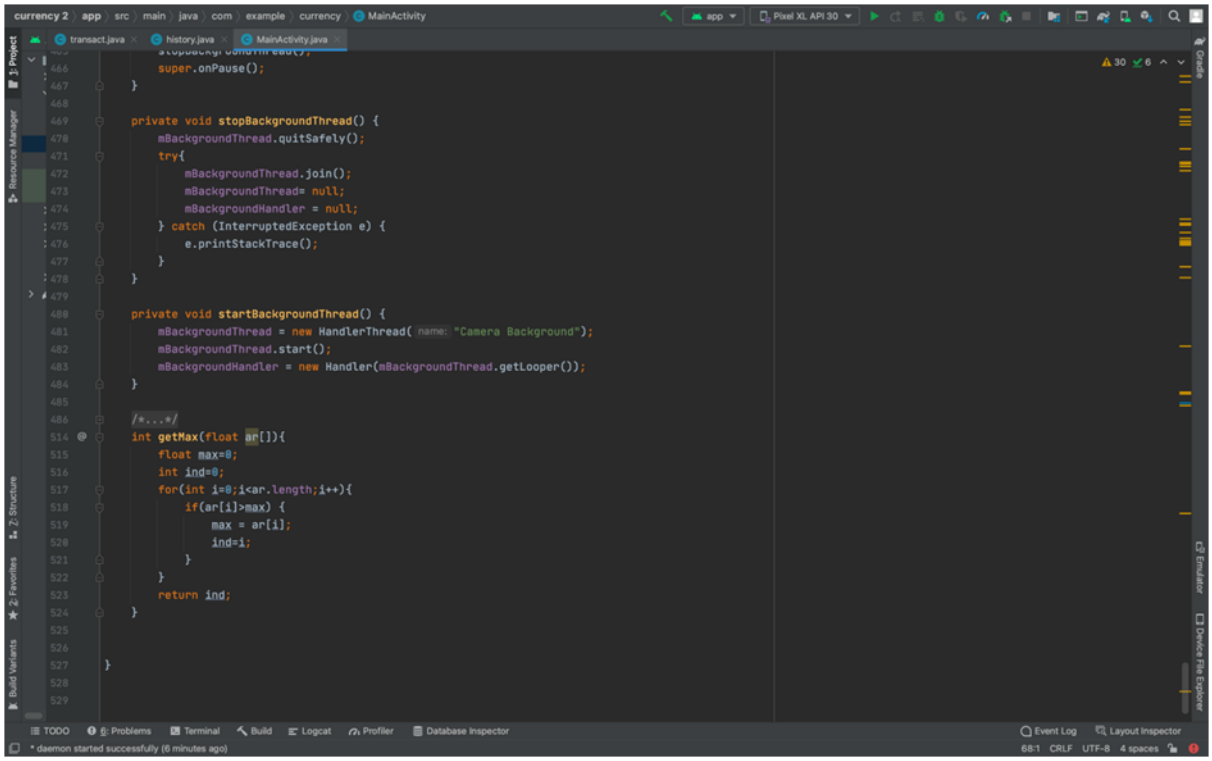
```
        CameraManager manager = (CameraManager)getSystemService(Context.CAMERA_SERVICE);
        try{
            cameraId = manager.getCameraIdList()[0];
            CameraCharacteristics characteristics = manager.getCameraCharacteristics(cameraId);
            StreamConfigurationMap map = characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP);
            assert map != null;
            imageDimension = map.getOutputSizes(SurfaceTexture.class)[0];
            //Check realtime permission if run higher API 23
            if(ActivityCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED)
            {
                ActivityCompat.requestPermissions( activity: this,new String[]{
                        Manifest.permission.CAMERA,
                        Manifest.permission.WRITE_EXTERNAL_STORAGE
                },REQUEST_CAMERA_PERMISSION);
                return;
            }
            manager.openCamera(cameraId,stateCallback, handler: null);

        } catch (CameraAccessException e) {
            e.printStackTrace();
        }

    }

    TextureView.SurfaceTextureListener textureListener = new TextureView.SurfaceTextureListener() {
        @Override
        public void onSurfaceTextureAvailable(SurfaceTexture surfaceTexture, int i, int i1) {
            openCamera();
        }

        @Override
        public void onSurfaceTextureSizeChanged(SurfaceTexture surfaceTexture, int i, int i1) {

        }

        @Override
        public boolean onSurfaceTextureDestroyed(SurfaceTexture surfaceTexture) { return false; }
```

Figure 5.10: coding2



```
@Override
public void onSurfaceTextureUpdated(SurfaceTexture surfaceTexture) {

    }
};

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    if(requestCode == REQUEST_CAMERA_PERMISSION)
    {
        if(grantResults[0] != PackageManager.PERMISSION_GRANTED)
        {
            Toast.makeText( context: this,  text: "You can't use camera without permission", Toast.LENGTH_SHORT).show();
            finish();
        }
    }
}

@Override
protected void onResume() {
    super.onResume();
    startBackgroundThread();
    if(textureView.isAvailable())
        openCamera();
    else
        textureView.setSurfaceTextureListener(textureListener);
}

@Override
protected void onPause() {
    stopBackgroundThread();
    super.onPause();
}

private void stopBackgroundThread() {
    mBackgroundThread.quitSafely();
    try{
```

Figure 5.11: onRequestPermissionsResult() will describe the permission of the user to give access to the camera.



```
    super.onPause();
}

private void stopBackgroundThread() {
    mBackgroundThread.quitSafely();
    try{
        mBackgroundThread.join();
        mBackgroundThread= null;
        mBackgroundHandler = null;
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

private void startBackgroundThread() {
    mBackgroundThread = new HandlerThread( name: "Camera Background");
    mBackgroundThread.start();
    mBackgroundHandler = new Handler(mBackgroundThread.getLooper());
}

/*...*/
int getMax(float ar[]){
    float max=0;
    int ind=0;
    for(int i=0;i<ar.length;i++){
        if(ar[i]>max) {
            max = ar[i];
            ind=i;
        }
    }
    return ind;
}


}
```

Figure 5.12: getMax() is user defined method used to retrieve the label which has more accuracy.

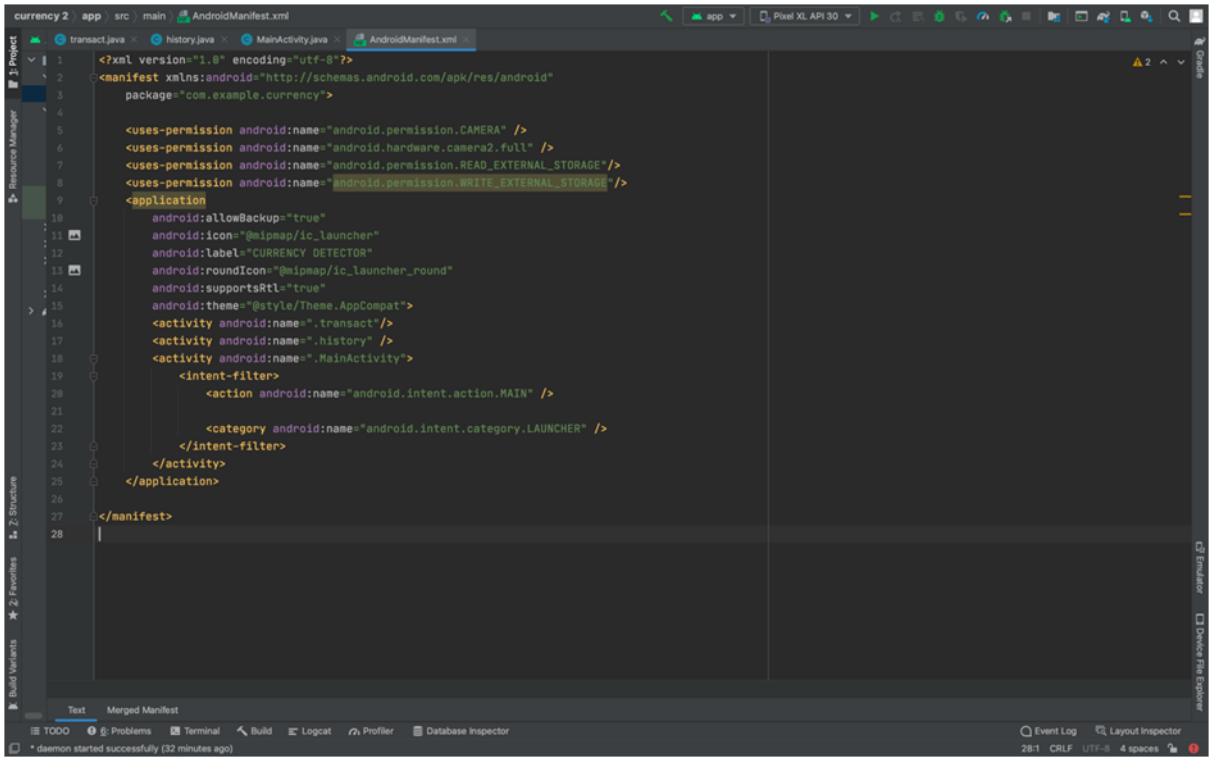


Figure 5.13: AndroidManifest file will give the overall details of the applications like permissions, activities, etc.

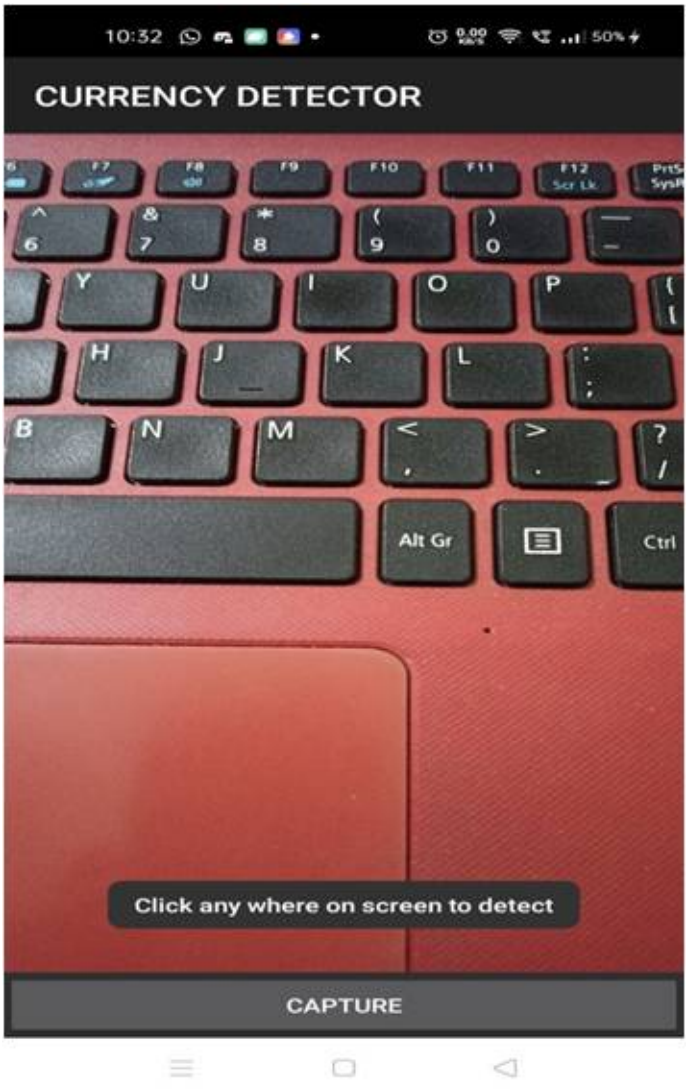


Figure 5.14: instructions

Figure 5.15: instructions

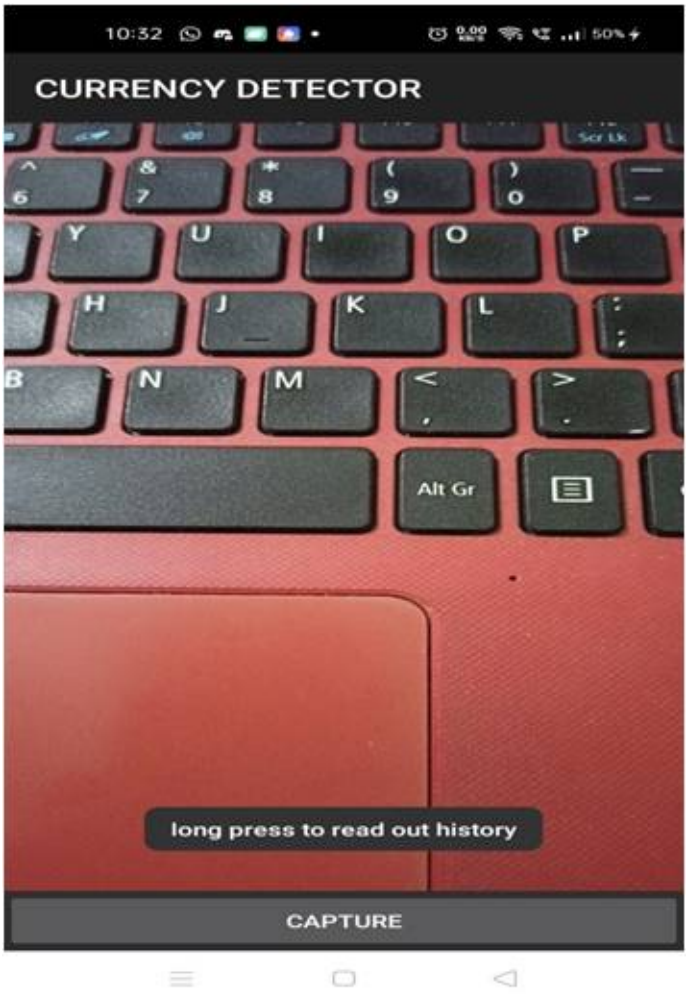


Figure 5.16: instructions

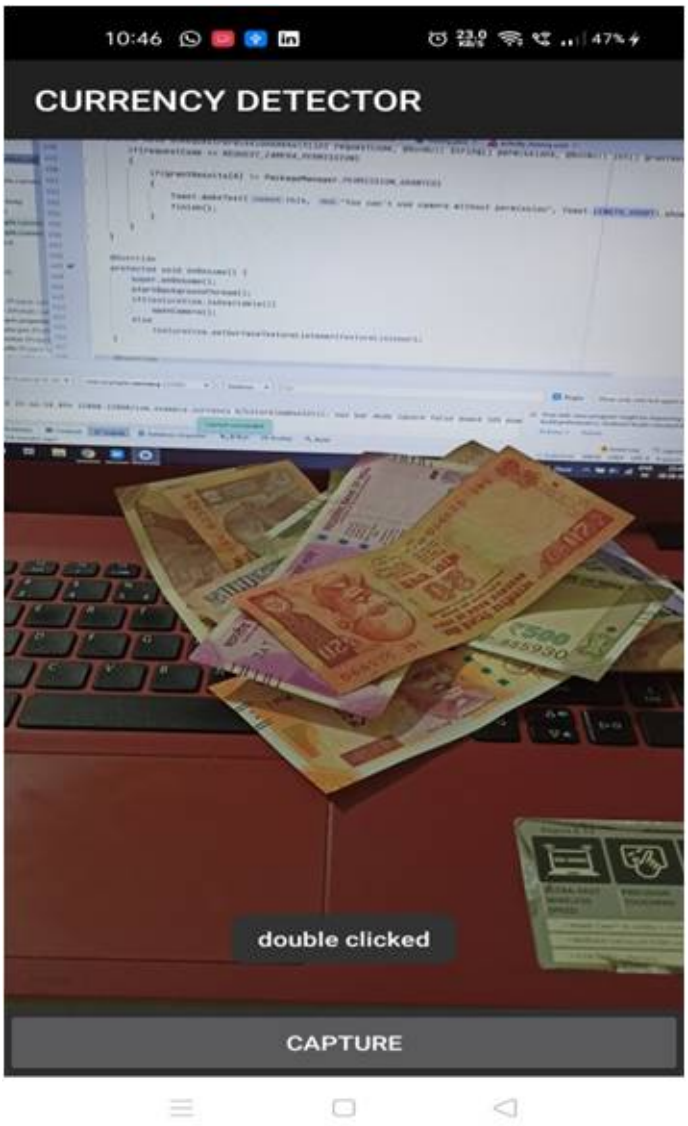


Figure 5.17: Double Clicked

Figure 5.18: 10 rupees new

Figure 5.19: 20 rupees new

Figure 5.20: detecting the currency

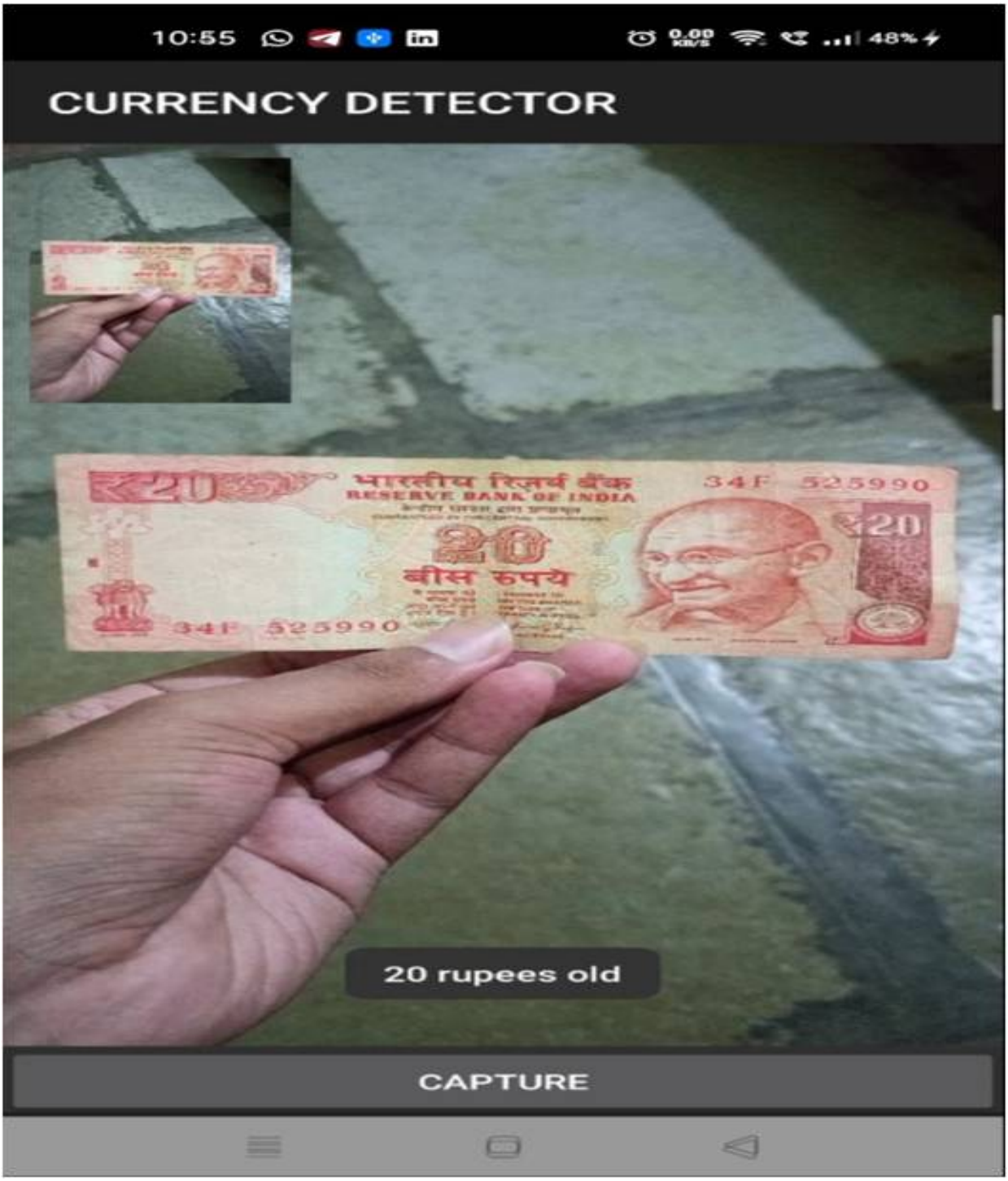


Figure 5.21: 20 rupees old

Figure 5.22: 50 rupees new

Figure 5.23: 50 rupees old

Figure 5.24: 100 rupees new

Figure 5.25: 100 rupees old

Figure 5.26: 200 rupees

Figure 5.27: 500 rupees

Figure 5.28: 2000 rupees

CHAPTER 6

Conclusions and Future Scope

6.1 Conclusion

This Application detects all the Indian Currency notes that are currently in circulation.

Application gives Instruction at first and then proceeds for detection.

After the detection, Application speaks out the detected currency value.

Gestures like swipes, double tap, triple tap, long press are used for navigating from one page to other page of the Application.

User currency detection and person-person currency detection are supported by this Application.

Application uses Azure Custom Vision API for accurate Image detection.

In this challenge, this system is beneficial for visually impaired human beings. By the use of this approach, we get extra accurate effects. The important aim for the advent of the forex detection model become for assisting the visually impaired humans and giving them a threat of shielding themselves from scammers who try to take earnings out of their incapacity. As scamming and dishonest are a punishable offense and those should be discouraged not to do it, seeking to scam someone who's abled is just inhuman and makes us comprehend to what extent humanity has fallen.

6.2 Future Scope

- Displaying History in a separate page(Activity).
- Storing the History in internal storage of the device for the user's need.
- Future improvements will also consist of a neural network only for checking if what is determined is an authentic foreign money or a

counterfeit currency by comparing the diverse symbols and protection markings furnished within the foreign money.

CHAPTER 7

References

7.1 References

1 .https://www.tensorflow.org/lite/guide/android.

2 .https://azure.microsoft.com/en-in/services/cognitive-services/custom-vision-service/.

3 .https://developer.android.com/reference/android/speech/tts/TextToSpeech.

4 .Snehal Saraf, Vrushali Sindhikar, Ankita Sonawane, Shamali Thakare, "Currency Recognition system for Visually Impaired", IJARIIE-ISSN(O)-2395-4396,2017.

5 .Barani.S, "Currency Identifier for Indian Denominations to Aid Visually Impaired", International Conference on Circuit, Power and Computing Technologies[ICCPCT],2015.

6 .Swati Sagar, Shaheen Mondal, Apoorva Seth, Roopvati Shah, Akansha Deshpande, "An Android Application forIndian Currency Recognition and Authentication for blind", IJCSN International Journal of Computer Science and Network, April 2016.

7 . Md.Ferdousur Sarker, Md Israfil Mahmud Raju, "Real Time Bangladeshi Currency Detection system for visually impaired person", December 2019.