A Mini- Project Report

On

“Android Project”

Submitted to the

Pune Institute of Computer Technology, Pune

In fulfillment for the award of the Degree of

Bachelor of Engineering

In

Information Technology

by

Tanvi Bhaskarwar	71829283K
Saniya Shah	71829185K

Under the guidance of

Prof. A. C. Karve

Department Of Information Technology

Pune Institute of Computer Technology

Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

2020-2021

CERTIFICATE

This is to certify that the project report entitled

“Assistant for Visually Impaired People”

Submitted by

Tanvi Bhaskarwar	71829283K
Saniya Shah	71829185K

is a bonafide work carried out by them under the supervision of Prof. A. C. Karve and it is approved.
for the fulfillment of the requirement of **Computer Laboratory - X** for the award of the Degree of Bachelor of Engineering (Information Technology).

Prof. A C Karve
Lab Teacher
Department of Information Technology

Dr A. M. Bagade
Head of Department
Department of Information Technology

Place:

Date:

ACKNOWLEDGEMENT

We thank everyone who has helped and provided valuable suggestions for successfully creating a wonderful project.

We are very grateful to our guide, Prof. A C Karve, Head of Department Dr. A. M. Bagade and our principal Dr R Sreemathy. They have been very supportive and have ensured that all facilities remained available for smooth progress of the project.

We would like to thank our professor and Prof. A C Karve for providing very valuable and timely suggestions and help.

Tanvi Bhaskarwar
Saniya Shah

ABSTRACT

Strolling securely and unhesitatingly with no human help with urban or obscure situations is a troublesome errand for visually impaired individuals. These individuals face a few issues throughout their life, one of these issues, that is the most indispensable one, is recognising objects when they are strolling. While moving starting with one spot then onto the next, they need assistance of others around.This project is for the most part centered around giving a kind of visual guide to these visually impaired people. Herein, we propose a framework wherein an Android cell phone is utilized to help a visually impaired client in object identification and route navigation. Consequently, this project utilizes an Android cell phone that utilizes its camera to recognize hindrances/objects in the environment and gives a sound output.

LIST OF FIGURES

LIST OF TOPICS

INTRODUCTION

The application developed can detect the objects in the user's surroundings. It can alert the user of the obstacles in his pathway and this way helps the user to navigate from one place to another saving him from tripping anywhere. This project makes the use of smartphones, a common device available to anyone and uses technology to make an application which can help the blind user detect objects in his surroundings and help him in navigating from one place to another. The output of the system is in audio form that can be easily understandable for a blind user.

This project takes a “speech” input and converts it into text to be used as input for object detection and the output is again converted to audio.Hence this project allows these individuals to easily track their route and be able to hear where objects lie in their path.This project hence makes use of Machine Learning to detect objects.

SCOPE AND OBJECTIVE

- **Scope :**

  Scope of this project is blind people should be able to know the things around them and able to survive with freedom and without any help.The main scope is blind people can use it for detection of objects and obstacles around them.

- **Objective :**

  This project is based on creating an application for the set of humans, who may not be in the position of using mobile phones for object detection, with the required comfort. In other words, object detection can be completely voice based. The application listens to your voice commands and then responds by converting your text to speech and vice versa. This application's capability to convert from speech to text and vice versa, a trendy feature of the Android platform for many years which can be effectively used by the blind to use. The main objective of this android application is to provide object detection system for the blinds in an effective way by exploring the powers of the platform to the maximum. The application provides better user interface and interaction is completely through voice. The application is expected to be effective with a convenient and comparatively simple user interface. The project attracts wider scope as Android is a commonly used platform today and is still facing a shortage of apps for common day-to-day facilities.

PROJECT FLOW

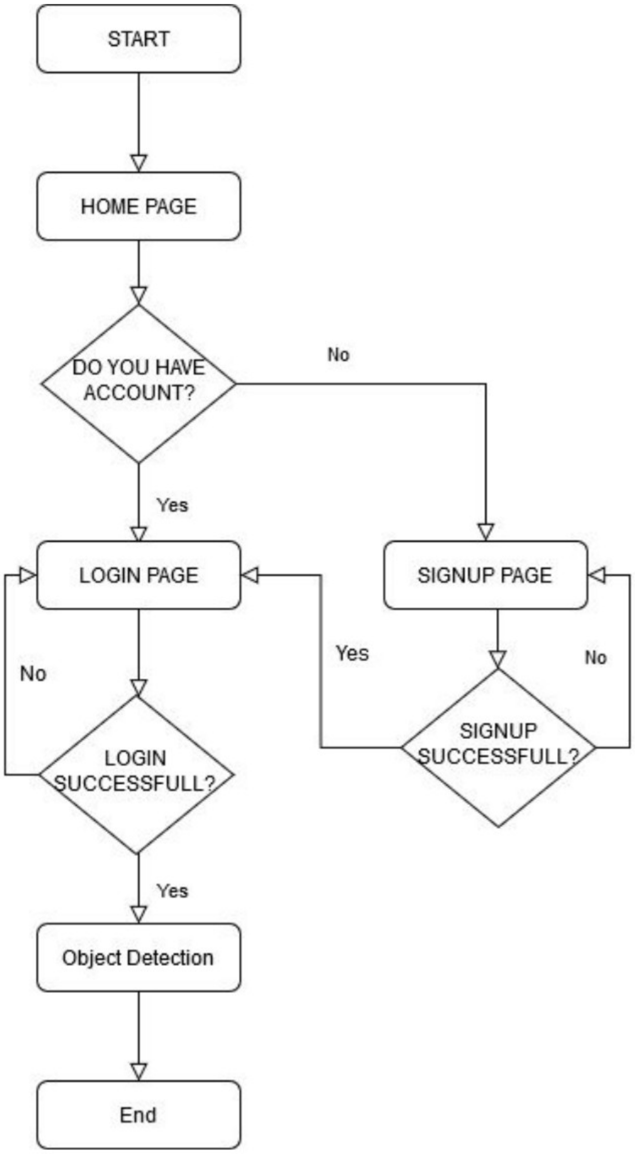
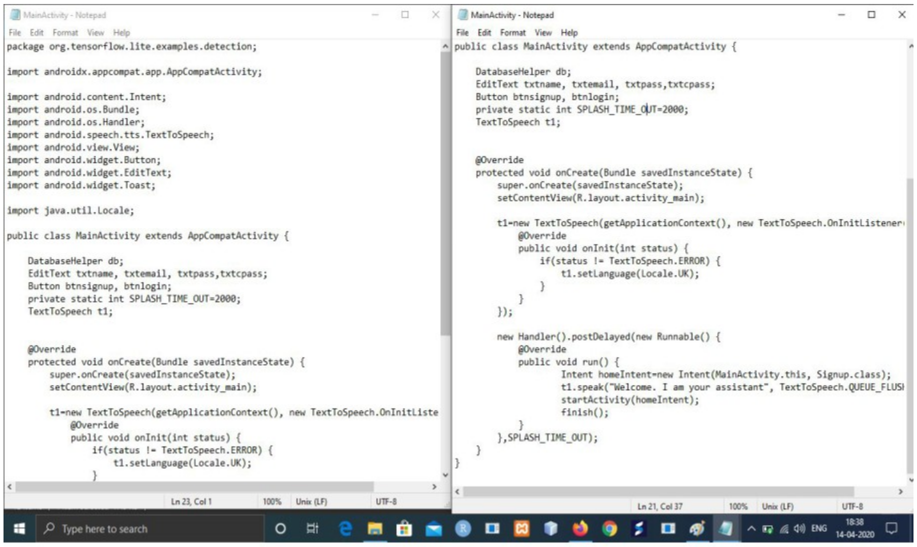


Fig 1 : Project Flow Diagram

CODE AND SNAPSHOT

CODE:

- **<u>MAINACTIVITY.JAVA :</u>**

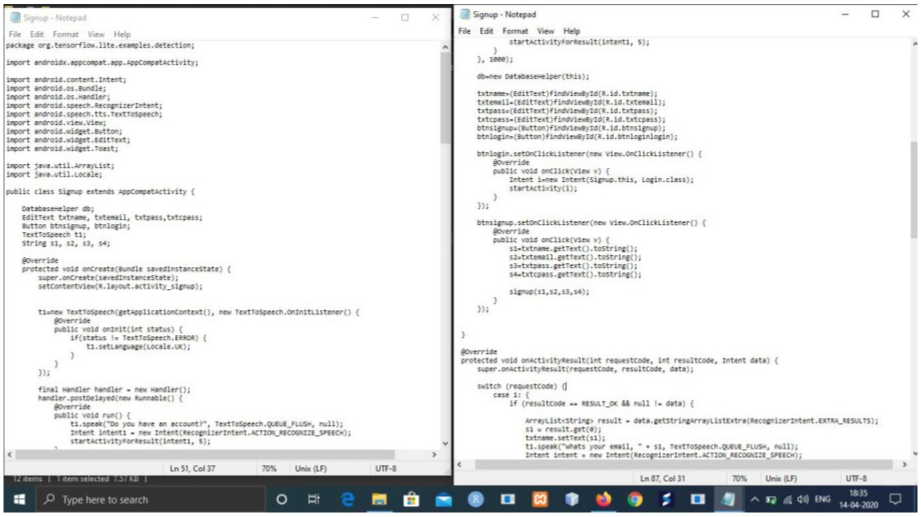


```
package org.tensorflow.lite.examples.detection;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    DatabaseHelper db;
    EditText txtname, txtemail, txtpass,txtcpass;
    Button btnsignup, btnlogin;
    private static int SPLASH_TIME_OUT=2000;
    TextToSpeech t1;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListe
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
```

```
public class MainActivity extends AppCompatActivity {

    DatabaseHelper db;
    EditText txtname, txtemail, txtpass,txtcpass;
    Button btnsignup, btnlogin;
    private static int SPLASH_TIME_OUT=2000;
    TextToSpeech t1;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                    Intent homeIntent=new Intent(MainActivity.this, Signup.class);
                    t1.speak("Welcome. I am your assistant", TextToSpeech.QUEUE_FLUSH
                    startActivity(homeIntent);
                    finish();
            }
        },SPLASH_TIME_OUT);
    }
}
```

Fig 2 : MainActivity.java

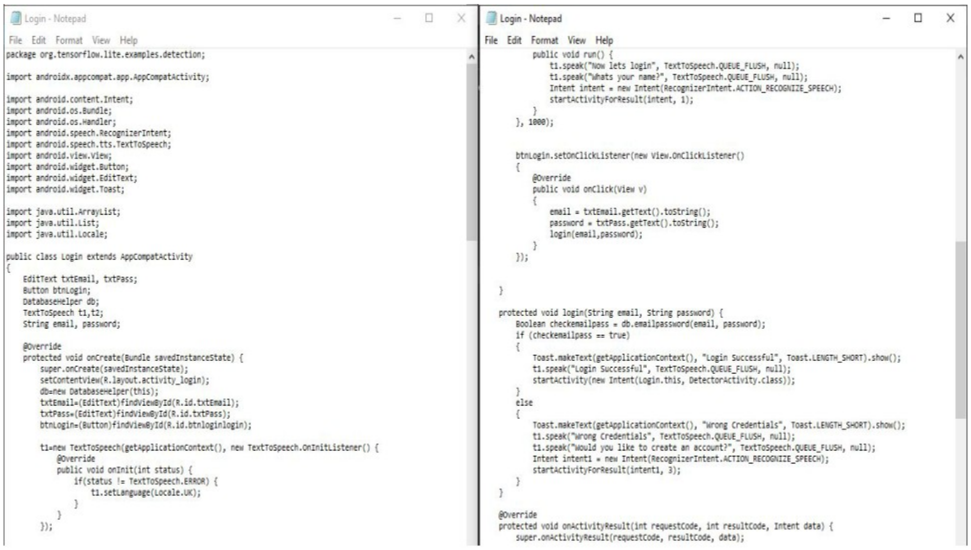
- **<u>SIGNUP.JAVA :</u>**



```
package org.tensorflow.lite.examples.detection;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Locale;

public class Signup extends AppCompatActivity {

    DatabaseHelper db;
    EditText txtname, txtemail, txtpass,txtcpass;
    Button btnsignup, btnlogin;
    TextToSpeech t1;
    String s1, s2, s3, s4;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);


        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });

        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                t1.speak("Do you have an account?", TextToSpeech.QUEUE_FLUSH, null);
                Intent intent1 = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                startActivityForResult(intent1, 5);
```

```
                startActivityForResult(intent1, 5);
            }
        }, 1000);

        db=new DatabaseHelper(this);

        txtname=(EditText)findViewById(R.id.txtname);
        txtemail=(EditText)findViewById(R.id.txtemail);
        txtpass=(EditText)findViewById(R.id.txtpass);
        txtcpass=(EditText)findViewById(R.id.txtcpass);
        btnsignup=(Button)findViewById(R.id.btnsignup);
        btnlogin=(Button)findViewById(R.id.btnloginlogin);

        btnlogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i=new Intent(Signup.this, Login.class);
                startActivity(i);
            }
        });

        btnsignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                s1=txtname.getText().toString();
                s2=txtemail.getText().toString();
                s3=txtpass.getText().toString();
                s4=txtcpass.getText().toString();

                signup(s1,s2,s3,s4);
            }
        });


    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case 1: {
                if (resultCode == RESULT_OK && null != data) {

                    ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    s1 = result.get(0);
                    txtname.setText(s1);
                    t1.speak("whats your email, " + s1, TextToSpeech.QUEUE_FLUSH, null);
                    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
```

Fig 3 : SignUp.java_1

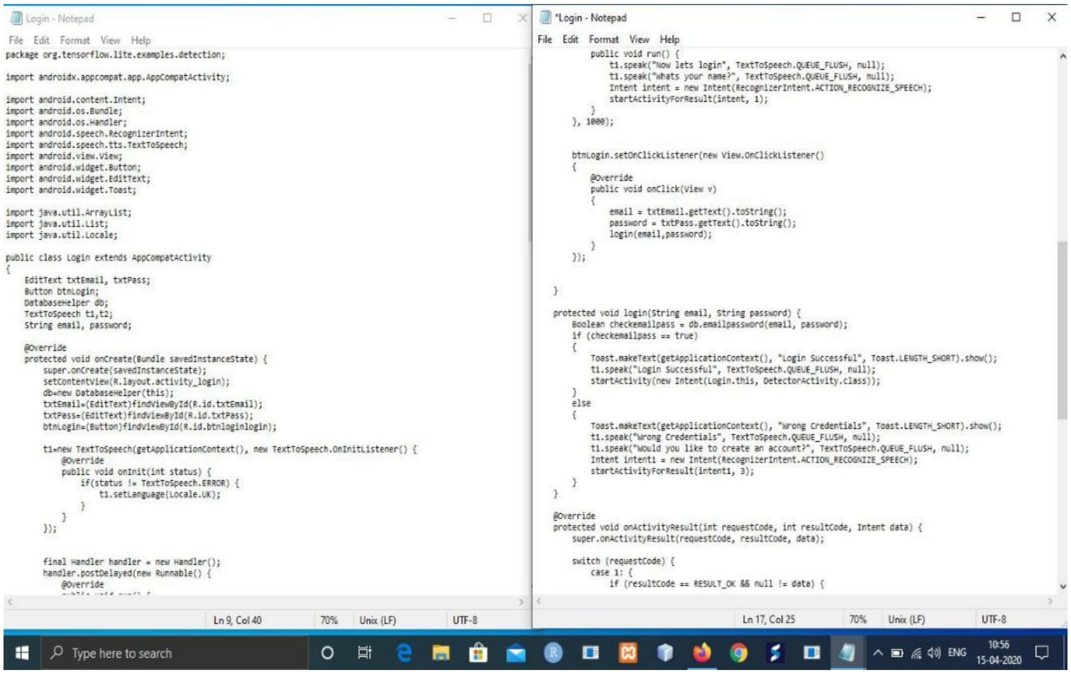


```java
package org.tensorflow.lite.examples.detection;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

public class Login extends AppCompatActivity
{
    EditText txtEmail, txtPass;
    Button btnLogin;
    DatabaseHelper db;
    TextToSpeech t1,t2;
    String email, password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        db=new DatabaseHelper(this);
        txtEmail=(EditText)findViewById(R.id.txtEmail);
        txtPass=(EditText)findViewById(R.id.txtPass);
        btnLogin=(Button)findViewById(R.id.btnloginlogin);

        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });
```

```java
            public void run() {
                t1.speak("Now lets login", TextToSpeech.QUEUE_FLUSH, null);
                t1.speak("Whats your name?", TextToSpeech.QUEUE_FLUSH, null);
                Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                startActivityForResult(intent, 1);
            }
        }, 1000);


        btnLogin.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                email = txtEmail.getText().toString();
                password = txtPass.getText().toString();
                login(email,password);
            }
        });


    }

    protected void login(String email, String password) {
        Boolean checkemailpass = db.emailpassword(email, password);
        if (checkemailpass == true)
        {
            Toast.makeText(getApplicationContext(), "Login Successful", Toast.LENGTH_SHORT).show();
            t1.speak("Login Successful", TextToSpeech.QUEUE_FLUSH, null);
            startActivity(new Intent(Login.this, DetectorActivity.class));
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Wrong Credentials", Toast.LENGTH_SHORT).show();
            t1.speak("Wrong Credentials", TextToSpeech.QUEUE_FLUSH, null);
            t1.speak("Would you like to create an account?", TextToSpeech.QUEUE_FLUSH, null);
            Intent intent1 = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
            startActivityForResult(intent1, 3);
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
```

Fig 4 : SignUp.java_2

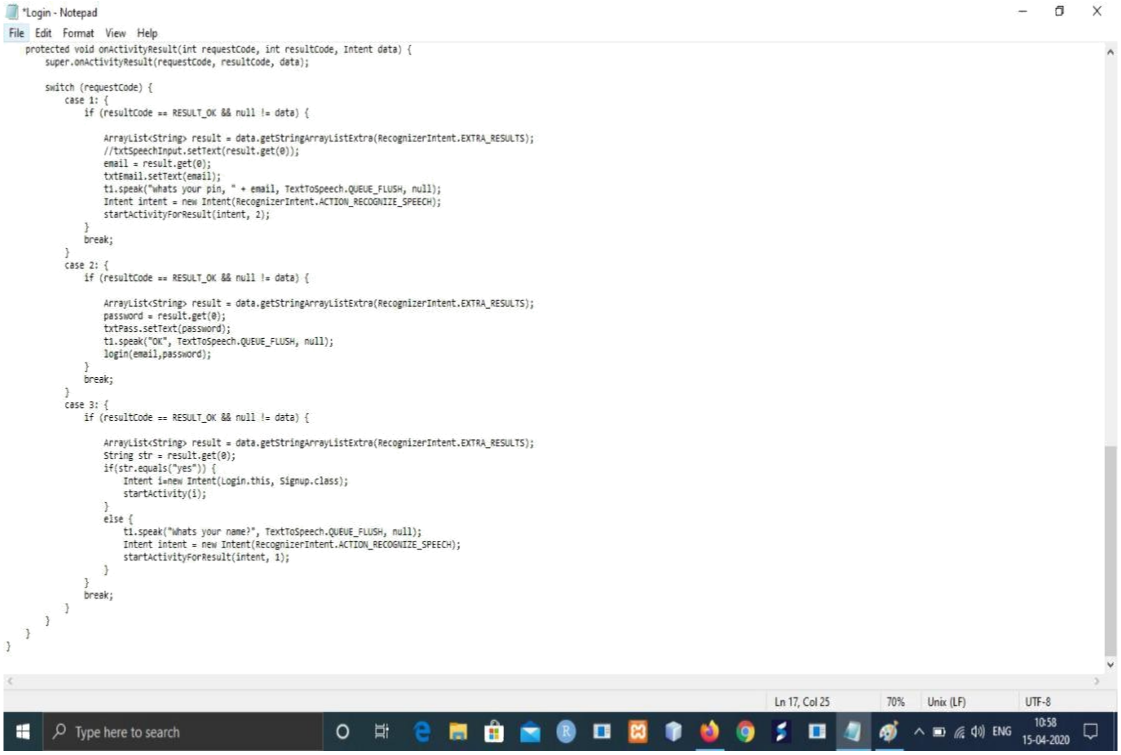
- **LOGIN.JAVA :**



```java
package org.tensorflow.lite.examples.detection;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

public class Login extends AppCompatActivity
{
    EditText txtEmail, txtPass;
    Button btnLogin;
    DatabaseHelper db;
    TextToSpeech t1,t2;
    String email, password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        db=new DatabaseHelper(this);
        txtEmail=(EditText)findViewById(R.id.txtEmail);
        txtPass=(EditText)findViewById(R.id.txtPass);
        btnLogin=(Button)findViewById(R.id.btnloginlogin);

        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });


        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
```

```java
            public void run() {
                t1.speak("Now lets login", TextToSpeech.QUEUE_FLUSH, null);
                t1.speak("Whats your name?", TextToSpeech.QUEUE_FLUSH, null);
                Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                startActivityForResult(intent, 1);
            }
        }, 1000);


        btnLogin.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                email = txtEmail.getText().toString();
                password = txtPass.getText().toString();
                login(email,password);
            }
        });


    }

    protected void login(String email, String password) {
        Boolean checkemailpass = db.emailpassword(email, password);
        if (checkemailpass == true)
        {
            Toast.makeText(getApplicationContext(), "Login Successful", Toast.LENGTH_SHORT).show();
            t1.speak("Login Successful", TextToSpeech.QUEUE_FLUSH, null);
            startActivity(new Intent(Login.this, DetectorActivity.class));
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Wrong Credentials", Toast.LENGTH_SHORT).show();
            t1.speak("Wrong Credentials", TextToSpeech.QUEUE_FLUSH, null);
            t1.speak("Would you like to create an account?", TextToSpeech.QUEUE_FLUSH, null);
            Intent intent1 = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
            startActivityForResult(intent1, 3);
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case 1: {
                if (resultCode == RESULT_OK && null != data) {
```

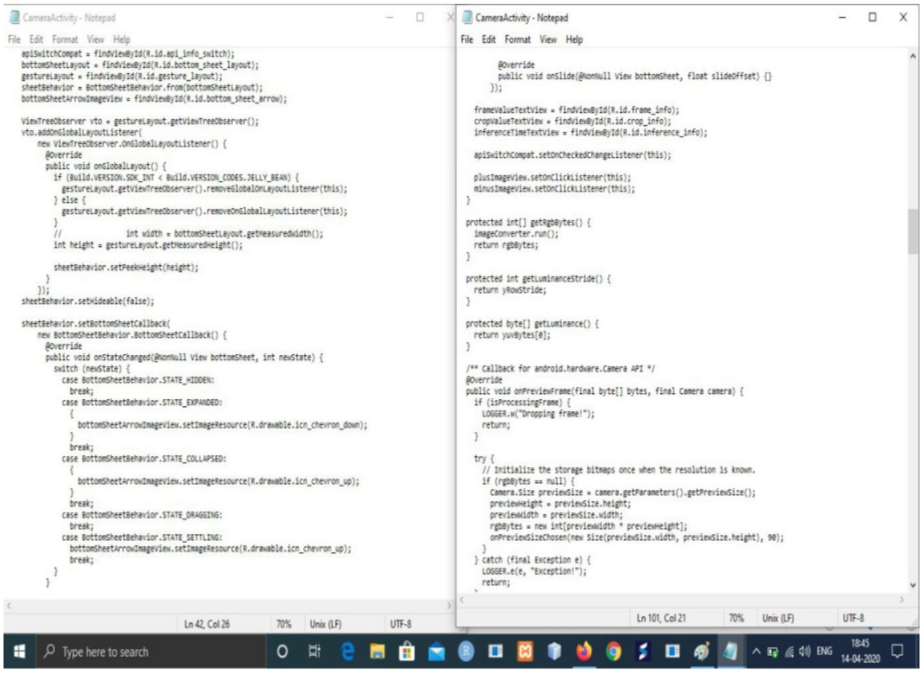
Fig 5 : Login.java_1



```
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case 1: {
                if (resultCode == RESULT_OK && null != data) {

                    ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    //txtSpeechInput.setText(result.get(0));
                    email = result.get(0);
                    txtEmail.setText(email);
                    t1.speak("whats your pin, " + email, TextToSpeech.QUEUE_FLUSH, null);
                    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                    startActivityForResult(intent, 2);
                }
                break;
            }
            case 2: {
                if (resultCode == RESULT_OK && null != data) {

                    ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    password = result.get(0);
                    txtPass.setText(password);
                    t1.speak("OK", TextToSpeech.QUEUE_FLUSH, null);
                    login(email,password);
                }
                break;
            }
            case 3: {
                if (resultCode == RESULT_OK && null != data) {

                    ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    String str = result.get(0);
                    if(str.equals("yes")) {
                        Intent i=new Intent(Login.this, Signup.class);
                        startActivity(i);
                    }
                    else {
                        t1.speak("Whats your name?", TextToSpeech.QUEUE_FLUSH, null);
                        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                        startActivityForResult(intent, 1);
                    }
                }
                break;
            }
        }
    }
}
```

Fig 6 : Login.java_2

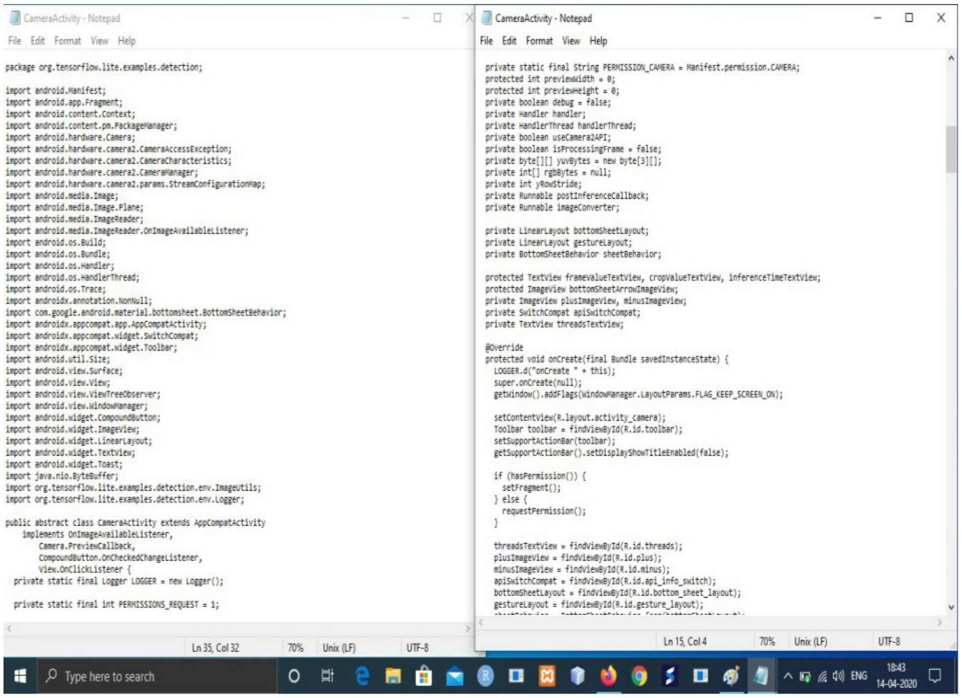
- **<u>CAMERAACTIVITY.JAVA :</u>**



```
    apiSwitchCompat = findViewById(R.id.api_info_switch);
    bottomSheetLayout = findViewById(R.id.bottom_sheet_layout);
    gestureLayout = findViewById(R.id.gesture_layout);
    sheetBehavior = BottomSheetBehavior.from(bottomSheetLayout);
    bottomSheetArrowImageView = findViewById(R.id.bottom_sheet_arrow);

    ViewTreeObserver vto = gestureLayout.getViewTreeObserver();
    vto.addOnGlobalLayoutListener(
        new ViewTreeObserver.OnGlobalLayoutListener() {
          @Override
          public void onGlobalLayout() {
            if (Build.VERSION.SDK_INT < Build.VERSION_CODES.JELLY_BEAN) {
              gestureLayout.getViewTreeObserver().removeGlobalOnLayoutListener(this);
            } else {
              gestureLayout.getViewTreeObserver().removeOnGlobalLayoutListener(this);
            }
            //                int width = bottomSheetLayout.getMeasuredWidth();
            int height = gestureLayout.getMeasuredHeight();

            sheetBehavior.setPeekHeight(height);
          }
        });
    sheetBehavior.setHideable(false);

    sheetBehavior.setBottomSheetCallback(
        new BottomSheetBehavior.BottomSheetCallback() {
          @Override
          public void onStateChanged(@NonNull View bottomSheet, int newState) {
            switch (newState) {
              case BottomSheetBehavior.STATE_HIDDEN:
                break;
              case BottomSheetBehavior.STATE_EXPANDED:
                {
                  bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_down);
                }
                break;
              case BottomSheetBehavior.STATE_COLLAPSED:
                {
                  bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_up);
                }
                break;
              case BottomSheetBehavior.STATE_DRAGGING:
                break;
              case BottomSheetBehavior.STATE_SETTLING:
                bottomSheetArrowImageView.setImageResource(R.drawable.icn_chevron_up);
                break;
            }
          }
```

```
          @Override
          public void onSlide(@NonNull View bottomSheet, float slideOffset) {}
        });

    frameValueTextView = findViewById(R.id.frame_info);
    cropValueTextView = findViewById(R.id.crop_info);
    inferenceTimeTextView = findViewById(R.id.inference_info);

    apiSwitchCompat.setOnCheckedChangeListener(this);

    plusImageView.setOnClickListener(this);
    minusImageView.setOnClickListener(this);
  }

  protected int[] getRgbBytes() {
    imageConverter.run();
    return rgbBytes;
  }

  protected int getLuminanceStride() {
    return yRowStride;
  }

  protected byte[] getLuminance() {
    return yuvBytes[0];
  }

  /** Callback for android.hardware.Camera API */
  @Override
  public void onPreviewFrame(final byte[] bytes, final Camera camera) {
    if (isProcessingFrame) {
      LOGGER.w("Dropping frame!");
      return;
    }

    try {
      // Initialize the storage bitmaps once when the resolution is known.
      if (rgbBytes == null) {
        Camera.Size previewSize = camera.getParameters().getPreviewSize();
        previewHeight = previewSize.height;
        previewWidth = previewSize.width;
        rgbBytes = new int[previewWidth * previewHeight];
        onPreviewSizeChosen(new Size(previewSize.width, previewSize.height), 90);
      }
    } catch (final Exception e) {
      LOGGER.e(e, "Exception!");
      return;
```

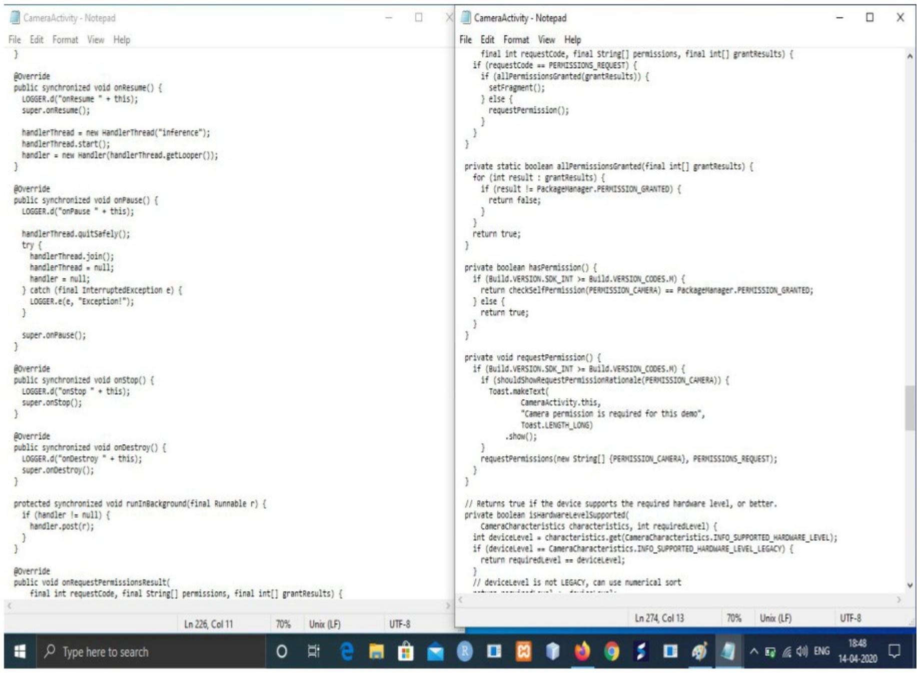
Fig 7 : CameraActivity.java_1



```
package org.tensorflow.lite.examples.detection;

import android.Manifest;
import android.app.Fragment;
import android.content.Context;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraCharacteristics;
import android.hardware.camera2.CameraManager;
import android.hardware.camera2.params.StreamConfigurationMap;
import android.media.Image;
import android.media.Image.Plane;
import android.media.ImageReader;
import android.media.ImageReader.OnImageAvailableListener;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.Trace;
import androidx.annotation.NonNull;
import com.google.android.material.bottomsheet.BottomSheetBehavior;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SwitchCompat;
import androidx.appcompat.widget.Toolbar;
import android.util.Size;
import android.view.Surface;
import android.view.View;
import android.view.ViewTreeObserver;
import android.view.WindowManager;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import java.nio.ByteBuffer;
import org.tensorflow.lite.examples.detection.env.ImageUtils;
import org.tensorflow.lite.examples.detection.env.Logger;

public abstract class CameraActivity extends AppCompatActivity
    implements OnImageAvailableListener,
        Camera.PreviewCallback,
        CompoundButton.OnCheckedChangeListener,
        View.OnClickListener {
  private static final Logger LOGGER = new Logger();

  private static final int PERMISSIONS_REQUEST = 1;
```

```
private static final String PERMISSION_CAMERA = Manifest.permission.CAMERA;
protected int previewWidth = 0;
protected int previewHeight = 0;
private boolean debug = false;
private Handler handler;
private HandlerThread handlerThread;
private boolean useCamera2API;
private boolean isProcessingFrame = false;
private byte[][] yuvBytes = new byte[3][];
private int[] rgbBytes = null;
private int yRowStride;
private Runnable postInferenceCallback;
private Runnable imageConverter;

private LinearLayout bottomSheetLayout;
private LinearLayout gestureLayout;
private BottomSheetBehavior sheetBehavior;

protected TextView frameValueTextView, cropValueTextView, inferenceTimeTextView;
protected ImageView bottomSheetArrowImageView;
private ImageView plusImageView, minusImageView;
private SwitchCompat apiSwitchCompat;
private TextView threadsTextView;

@Override
protected void onCreate(final Bundle savedInstanceState) {
  LOGGER.d("onCreate " + this);
  super.onCreate(null);
  getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

  setContentView(R.layout.activity_camera);
  Toolbar toolbar = findViewById(R.id.toolbar);
  setSupportActionBar(toolbar);
  getSupportActionBar().setDisplayShowTitleEnabled(false);

  if (hasPermission()) {
    setFragment();
  } else {
    requestPermission();
  }

  threadsTextView = findViewById(R.id.threads);
  plusImageView = findViewById(R.id.plus);
  minusImageView = findViewById(R.id.minus);
  apiSwitchCompat = findViewById(R.id.api_info_switch);
  bottomSheetLayout = findViewById(R.id.bottom_sheet_layout);
  gestureLayout = findViewById(R.id.gesture_layout);
```

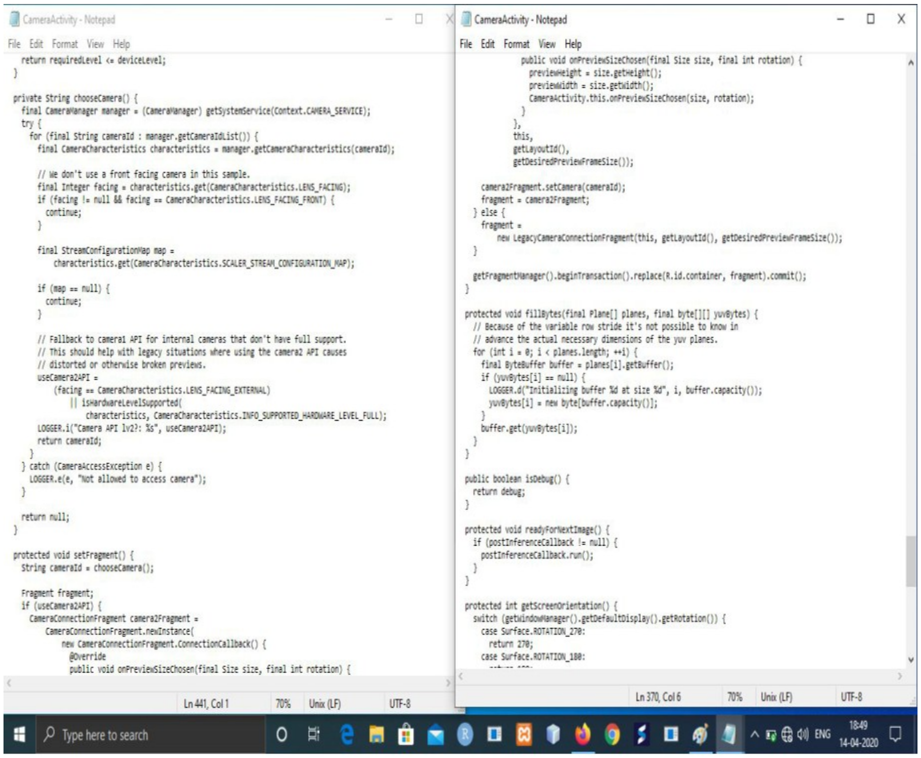
Fig 8 : CameraActivity.java_2



```
}

@Override
public synchronized void onResume() {
  LOGGER.d("onResume " + this);
  super.onResume();

  handlerThread = new HandlerThread("inference");
  handlerThread.start();
  handler = new Handler(handlerThread.getLooper());
}

@Override
public synchronized void onPause() {
  LOGGER.d("onPause " + this);

  handlerThread.quitSafely();
  try {
    handlerThread.join();
    handlerThread = null;
    handler = null;
  } catch (final InterruptedException e) {
    LOGGER.e(e, "Exception!");
  }

  super.onPause();
}

@Override
public synchronized void onStop() {
  LOGGER.d("onStop " + this);
  super.onStop();
}

@Override
public synchronized void onDestroy() {
  LOGGER.d("onDestroy " + this);
  super.onDestroy();
}

protected synchronized void runInBackground(final Runnable r) {
  if (handler != null) {
    handler.post(r);
  }
}

@Override
public void onRequestPermissionsResult(
    final int requestCode, final String[] permissions, final int[] grantResults) {
```

```
    final int requestCode, final String[] permissions, final int[] grantResults) {
  if (requestCode == PERMISSIONS_REQUEST) {
    if (allPermissionsGranted(grantResults)) {
      setFragment();
    } else {
      requestPermission();
    }
  }
}

private static boolean allPermissionsGranted(final int[] grantResults) {
  for (int result : grantResults) {
    if (result != PackageManager.PERMISSION_GRANTED) {
      return false;
    }
  }
  return true;
}

private boolean hasPermission() {
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    return checkSelfPermission(PERMISSION_CAMERA) == PackageManager.PERMISSION_GRANTED;
  } else {
    return true;
  }
}

private void requestPermission() {
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    if (shouldShowRequestPermissionRationale(PERMISSION_CAMERA)) {
      Toast.makeText(
              CameraActivity.this,
              "Camera permission is required for this demo",
              Toast.LENGTH_LONG)
          .show();
    }
    requestPermissions(new String[] {PERMISSION_CAMERA}, PERMISSIONS_REQUEST);
  }
}

// Returns true if the device supports the required hardware level, or better.
private boolean isHardwareLevelSupported(
    CameraCharacteristics characteristics, int requiredLevel) {
  int deviceLevel = characteristics.get(CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL);
  if (deviceLevel == CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL_LEGACY) {
    return requiredLevel == deviceLevel;
  }
  // deviceLevel is not LEGACY, can use numerical sort
```

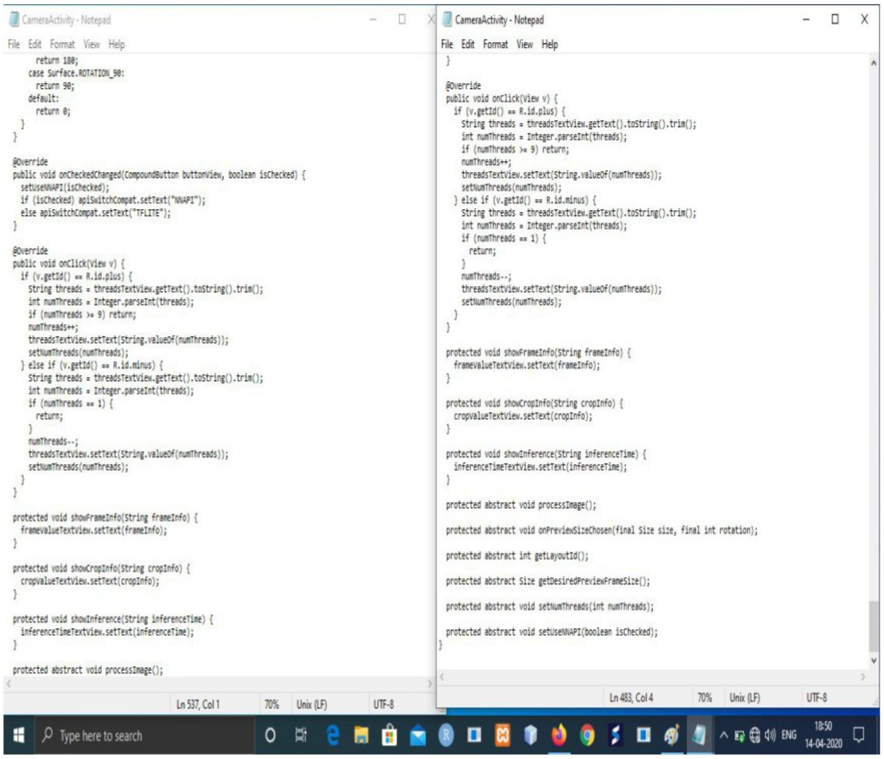
Fig 9 : CameraActivity.java_3



```
    return requiredLevel <= deviceLevel;
  }

  private String chooseCamera() {
    final CameraManager manager = (CameraManager) getSystemService(Context.CAMERA_SERVICE);
    try {
      for (final String cameraId : manager.getCameraIdList()) {
        final CameraCharacteristics characteristics = manager.getCameraCharacteristics(cameraId);

        // We don't use a front facing camera in this sample.
        final Integer facing = characteristics.get(CameraCharacteristics.LENS_FACING);
        if (facing != null && facing == CameraCharacteristics.LENS_FACING_FRONT) {
          continue;
        }

        final StreamConfigurationMap map =
            characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP);

        if (map == null) {
          continue;
        }

        // Fallback to camera1 API for internal cameras that don't have full support.
        // This should help with legacy situations where using the camera2 API causes
        // distorted or otherwise broken previews.
        useCamera2API =
            (facing == CameraCharacteristics.LENS_FACING_EXTERNAL)
                || isHardwareLevelSupported(
                    characteristics, CameraCharacteristics.INFO_SUPPORTED_HARDWARE_LEVEL_FULL);
        LOGGER.i("Camera API lv2?: %s", useCamera2API);
        return cameraId;
      }
    } catch (CameraAccessException e) {
      LOGGER.e(e, "Not allowed to access camera");
    }

    return null;
  }

  protected void setFragment() {
    String cameraId = chooseCamera();

    Fragment fragment;
    if (useCamera2API) {
      CameraConnectionFragment camera2Fragment =
          CameraConnectionFragment.newInstance(
              new CameraConnectionFragment.ConnectionCallback() {
                @Override
                public void onPreviewSizeChosen(final Size size, final int rotation) {
```

```
                public void onPreviewSizeChosen(final Size size, final int rotation) {
                  previewHeight = size.getHeight();
                  previewWidth = size.getWidth();
                  CameraActivity.this.onPreviewSizeChosen(size, rotation);
                }
              },
              this,
              getLayoutId(),
              getDesiredPreviewFrameSize());

      camera2Fragment.setCamera(cameraId);
      fragment = camera2Fragment;
    } else {
      fragment =
          new LegacyCameraConnectionFragment(this, getLayoutId(), getDesiredPreviewFrameSize());
    }

    getFragmentManager().beginTransaction().replace(R.id.container, fragment).commit();
  }

  protected void fillBytes(final Plane[] planes, final byte[][] yuvBytes) {
    // Because of the variable row stride it's not possible to know in
    // advance the actual necessary dimensions of the yuv planes.
    for (int i = 0; i < planes.length; ++i) {
      final ByteBuffer buffer = planes[i].getBuffer();
      if (yuvBytes[i] == null) {
        LOGGER.d("Initializing buffer %d at size %d", i, buffer.capacity());
        yuvBytes[i] = new byte[buffer.capacity()];
      }
      buffer.get(yuvBytes[i]);
    }
  }

  public boolean isDebug() {
    return debug;
  }

  protected void readyForNextImage() {
    if (postInferenceCallback != null) {
      postInferenceCallback.run();
    }
  }

  protected int getScreenOrientation() {
    switch (getWindowManager().getDefaultDisplay().getRotation()) {
      case Surface.ROTATION_270:
        return 270;
      case Surface.ROTATION_180:
```

Fig 10 : CameraActivity.java_4

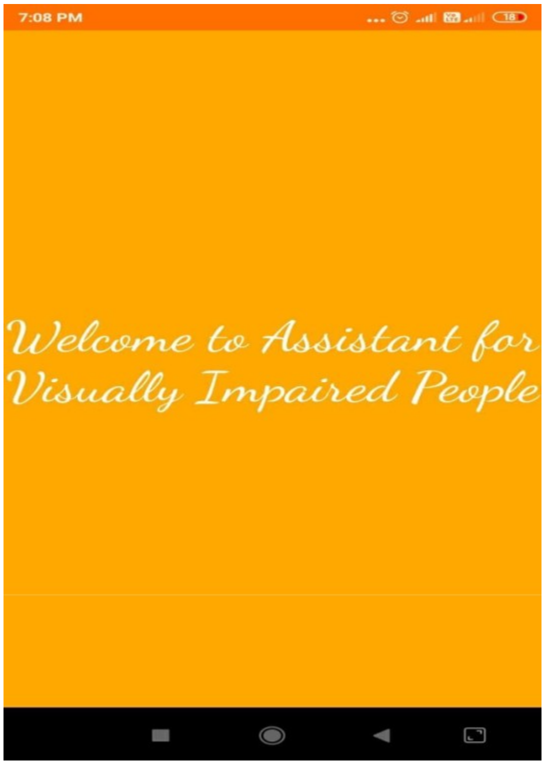


```
        return 180;
      case Surface.ROTATION_90:
        return 90;
      default:
        return 0;
    }
  }

  @Override
  public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    setUseNNAPI(isChecked);
    if (isChecked) apiSwitchCompat.setText("NNAPI");
    else apiSwitchCompat.setText("TFLITE");
  }

  @Override
  public void onClick(View v) {
    if (v.getId() == R.id.plus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads >= 9) return;
      numThreads++;
      threadsTextView.setText(String.valueOf(numThreads));
      setNumThreads(numThreads);
    } else if (v.getId() == R.id.minus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads == 1) {
        return;
      }
      numThreads--;
      threadsTextView.setText(String.valueOf(numThreads));
      setNumThreads(numThreads);
    }
  }

  protected void showFrameInfo(String frameInfo) {
    frameValueTextView.setText(frameInfo);
  }

  protected void showCropInfo(String cropInfo) {
    cropValueTextView.setText(cropInfo);
  }

  protected void showInference(String inferenceTime) {
    inferenceTimeTextView.setText(inferenceTime);
  }

  protected abstract void processImage();
```

```
  }

  @Override
  public void onClick(View v) {
    if (v.getId() == R.id.plus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads >= 9) return;
      numThreads++;
      threadsTextView.setText(String.valueOf(numThreads));
      setNumThreads(numThreads);
    } else if (v.getId() == R.id.minus) {
      String threads = threadsTextView.getText().toString().trim();
      int numThreads = Integer.parseInt(threads);
      if (numThreads == 1) {
        return;
      }
      numThreads--;
      threadsTextView.setText(String.valueOf(numThreads));
      setNumThreads(numThreads);
    }
  }

  protected void showFrameInfo(String frameInfo) {
    frameValueTextView.setText(frameInfo);
  }

  protected void showCropInfo(String cropInfo) {
    cropValueTextView.setText(cropInfo);
  }

  protected void showInference(String inferenceTime) {
    inferenceTimeTextView.setText(inferenceTime);
  }

  protected abstract void processImage();

  protected abstract void onPreviewSizeChosen(final Size size, final int rotation);

  protected abstract int getLayoutId();

  protected abstract Size getDesiredPreviewFrameSize();

  protected abstract void setNumThreads(int numThreads);

  protected abstract void setUseNNAPI(boolean isChecked);
}
```

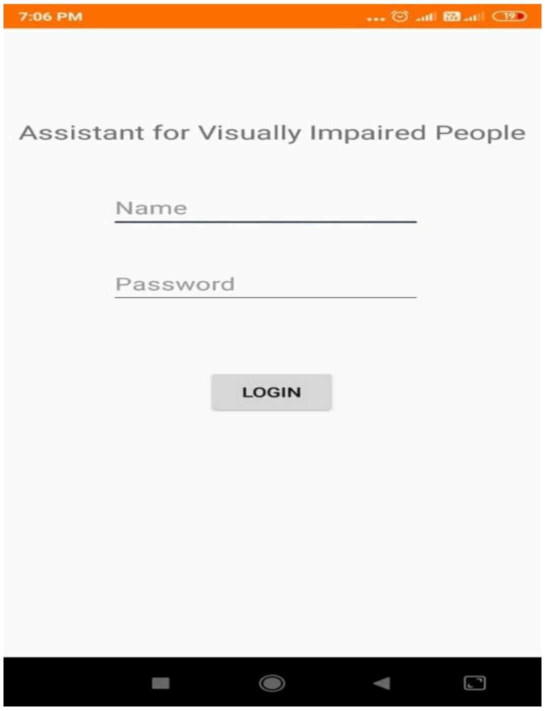
Fig 11 : CameraActivity.java_5

OUTPUT :


7:08 PM

Welcome to Assistant for Visually Impaired People

Fig 12 : Home Screen

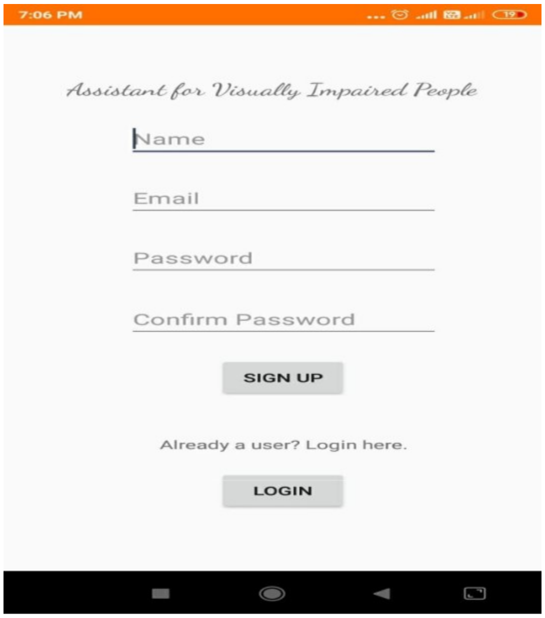

7:06 PM

Assistant for Visually Impaired People

Name

Password

Fig 13 : Login Screen



7:06 PM

Assistant for Visually Impaired People

Name

Email

Password

Confirm Password

SIGN UP

Already a user? Login here.

LOGIN

Fig 14 : SignUp Screen

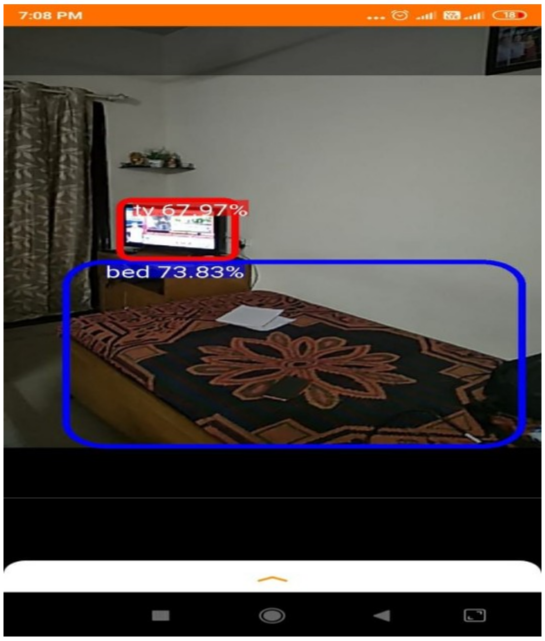


Fig 15 : Object Detection Screen

RESULTS

The said application is meant for visually imapired people by providing them real time access to the objects around their current location by making use of GPS and voice assistance to them. Speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread. It allows environmental barriers to be removed for people with a wide range of disabilities. Many believe that this mode is a big step for speech recognition technology. The accuracy of the system has significantly increased and become more accessible to everyone. The object detection is tested on different objects and a voice assistant is embedded in the application which helps them to know the real time objects and entities surrounding them. The said app is divided into different modules having different technologies tied together. The accuracy of the system is collective accuracies of the different sub modules.

The common performance metrics for object detection is Mean Average Precision (mAP). The higher the value of mAP, higher the performance of the model. In order to use mAP in object detection, all predicted boxes and classes are sorted in decreasing order of probability and matched with ground-truth boxes and classes. If the classes of the prediction and the ground truth match and their Intersection over Union (IoU) is greater or equal to 0.5 (0.5IOU), the prediction is considered a match. The match is predicted as a true positive if and only if it has not previously been used. The Average Precision is computed as the area under the precision/recall curve by numerical integration, and the mAP is achieved by calculating the mean of the Average Precision of all classes.

From calculations it has been observed that the SSD MobileNet V2 model has mAP = 91.90% with inference time 438 ms.

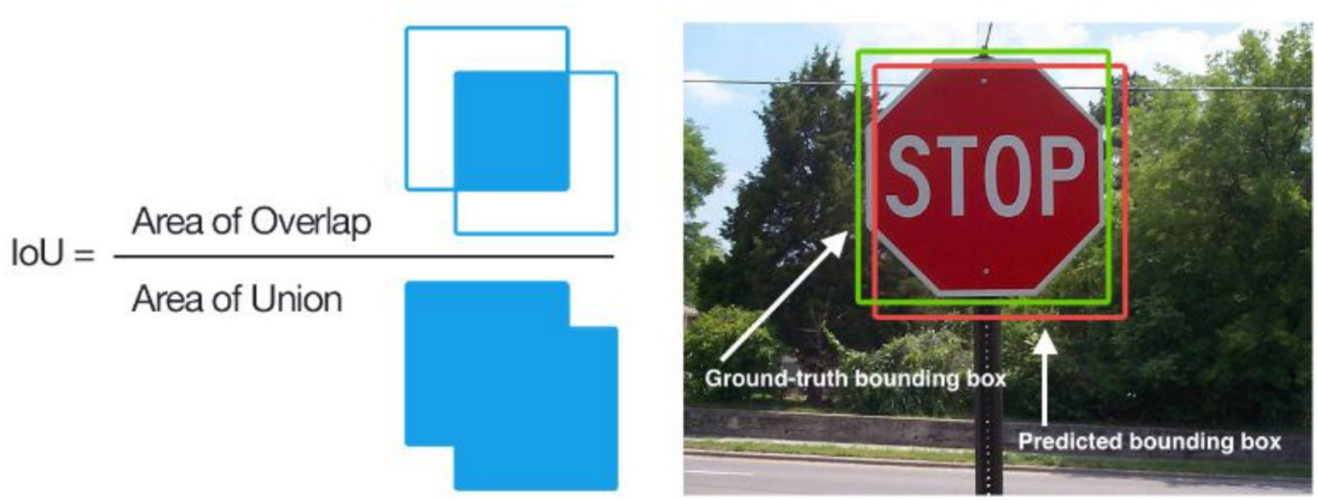


Fig 16: IOU

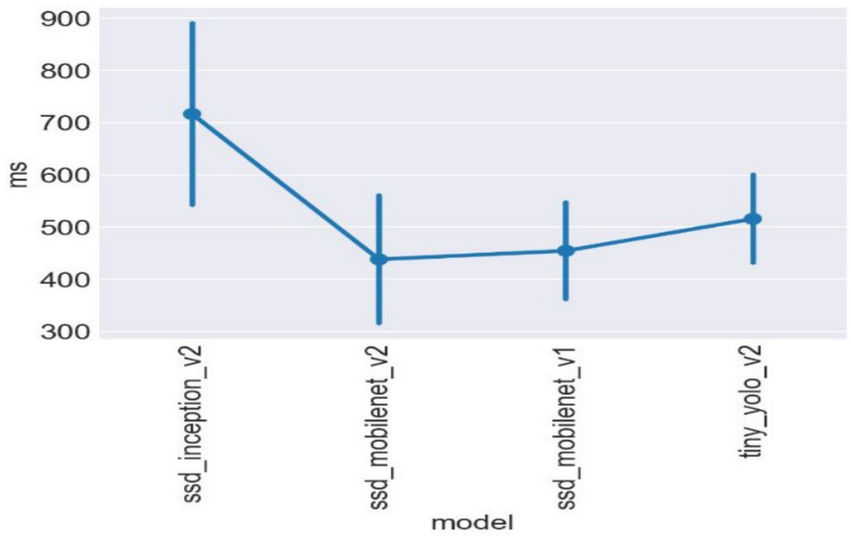


Fig 17: Graph of Inference Time Vs Model

CONCLUSION

In this report we have developed an application for visually impaired people using object detection CNN and voice assistance for use on mobile devices. Utilising multiple base models and object detection frameworks as well as using text to speech framework we successfully trained and implemented a model that could be used for real-time detection of objects in an Android application. Using Transfer Learning, the network could be trained on a relatively small amount of data while still achieving high mAP scores. Our results conclude that the app will be reliable, secure as well as useful for visually challenged people.

FUTURE SCOPE

The addition of Google Maps Turn by Turn Application Programming Interface would allow a visually impaired person to navigate and reach their destination with the help of voice based, step by step instructions. However, access to this API is paid, which meant it was outside the scope of our mini project.

Also, currently the application is only capable of recognizing objects categorized into 140 types based on the labels provided in training from the COCO dataset. This can be improved upon by providing the object detection model with a larger array of labels in the training phase. A bigger model would also require more processing power on mobile phones in order to keep inferencing time acceptably low, and year over year advancements in mobile silicon should help with the same.

REFERENCES

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

[2] https://medium.com/@rdeep/tensorflow-lite-tutorial-easy-implementation-in-android-14544 3ec3775

[3] http://www.diva-portal.org/smash/get/diva2%3A1242627/FULLTEXT01.pdf