OBJECT DETECTION USING IMAGE PROCESSING

# A MINI PROJECT REPORT

*Submitted by*

|  |  |
| --- | --- |
| **DEEPAK N** | **(711719205012)** |
| **SANTHIYA S** | **(711719205046)** |
| **SUDHAKAR N** | **(711719205056)** |
| **VAHINI G** | **(711719205058)** |

*in partial fulfillment for the completion of the course*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

# INFORMATION TECHNOLOGY

**KGiSL INSTITUTE OF TECHNOLOGY, COIMBATORE**

**ANNA UNIVERSITY::CHENNAI 600 025**

**JUNE 2022**

**CERTIFICATE EVALUATION**

**COLLEGE NAME :** KGiSL INSTITUTE OF TECHNOLOGY

**BRANCH :** INFORMATION TECHNOLOGY

**SEMESTER :** VI

**TITLE : OBJECT DETECTION USING IMAGE PROCESSING**

|  |  |  |
| --- | --- | --- |
| **STUDENT NAME** | **REGISTER No.** | **SUPERVISOR NAME** |
| DEEPAK N | 711719205012 | Dr. M. ANANTHI    ASSISTANT PROFESSOR  DEPARTMENT OF  INFORMATION TECHNOLOGY |
| SANTHIYA S | 711719205046 |
| SUDHAKAR N | 711719205056 |
| VAHINI G | 711719205058 |

The report of this mini project is submitted by the above students in partial fulfillment for the completion of the course in sixth semester Bachelor of Technology degree in Information Technology of Anna University and is evaluated and confirmed to the reports of work done by the above students.

Submitted for the University Project Viva Voice examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

ANNA UNIVERSITY::CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this mini project report "OBJECT DETECTION USING IMAGE PROCESSING" is the bonafide work of Deepak N (711719205012), Santhiya S (711719205046), Sudhakar N (711719205056), Vahini.G (711719205058), who carried out the mini project work under my supervision.

**SIGNATURE**

Dr. Sankar Ram N,

**HEAD OF THE DEPARTMENT,**

Professor,

Department of Information Technology, KGiSL Institute of Technology, Coimbatore-641035.

**SIGNATURE**

Dr. M.Ananthi, AP/IT,

**SUPERVISOR,**

Assistant Professor,

Department of Information Technology, KGiSL Institute of Technology, Coimbatore-641035.

**ACKNOWLEDGEMENT**

We wish express our deep sense of gratitude to **Padmashri. Dr. G Bakthavathsalam**,

Founder-chairman of KG Hospital, for having provided the facilities during the course

of our study in the college.

We express our gratitude to our Managing Director **Dr. Ashok Bakthavathsalam**, KGiSL Educational Institutions, who gave the opportunity to frame the project to the full satisfaction.

We are grateful to **Dr. Selvam M**, M.E., Ph.D., Principal, KGiSL Institute of Technology for his valuable guidance and blessings.

We express our deep and sincere gratitude to **Mr. Aravind Kumar** **Rajendran**, CEO, Academic Initiatives, KGiSL Educational Institutions, for his motivation.

Our grateful thanks is also extended to Head of the Department Dr. **Sankar Ram N**, M.E, Ph.D., FIE., for his support throughout the project.

In this regard, we wish to express our gratitude and sincere thanks to our supervisor**,** **Dr. Ananthi M**, M.E., Ph.D., for extending her inspiration and guidelines in implementing this project successfully.

Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S NO.** | **TITLE** | **PAGE NO.** |
|  | ABSTRACT | I |
| 1 | INTRODUCTION  1.1 OBJECTIVE | 1 |
| 2 | EXISTING SYSTEM | 1 |
| 3 | PROPOSED SYSTEM | 1 |
| 4 | MOTIVATION | 1 |
| 5 | PROBLEM STATEMENT | 2 |
| 6 | SOFTWARE SPECIFICATION REQUIREMENTS | 2 |
| 7 | FLOWCHART | 4 |
| 8 | CONCLUSION | 4 |
| 9 | ANNEXURES  9A. SCREEN SHOTS  9B. SOURCE CODE | 5  10 |
| 10 | REFERENCES | 20 |

**ABSTRACT**

In the last decades, the increasing potential of Information technology revolutionized data and information management, in particular, the data acquisition, data processing, and predictions. The effort has been truly interdisciplinary, where, image processing techniques, and AI based model implementation have played their roles. The latest technology innovations have enabled the researchers to execute computational experiments which would had never been possible if would have tried using the conventional methods. This provides study of various methodologies for object detection and also systematic analysis of various existing object detection techniques with precise and arranged representation. We have backed up the study with the merits and demerits of existing methods and the future scope in this area. This project “Object Detection using Image Processing” application is a tourist guiding application. This application is proposed towards the identification of the places for tourists by detecting the object using android camera. This helps tourists to identify the places with its history by just placing the object in front of the camera. This application. Mainly focuses on the Monuments and hence this application is specially build for Tourists.

1. INTRODUCTION

1.1. OBJECTIVE

To easily identify places and objects by just placing the camera on the object or place. An application that is efficient and easy to use by the Tourists.

2. EXISTING SYSTEM

The existing system was studied that are applications that detects some specified hotels and places like parks and the details.

Drawbacks – only for specified places like private places and does not help to get the information about common tourists places that is less popular.

3. PROPOSED SYSTEM

The system here developed is the detection of place with its well-known details.

This includes the detection of objects with certain knowledge about that object by placing in the camera through this application.

Features: Detection of place including the well-known details.

4. MOTIVATION

Currently there is a large movement in the tourist applications that helps them in guidance and improve current systems. As with any business, this effort is difficult without an “information at your fingertips” type of application. A well- designed application is a large improvement over the current methods of managing tourist case files and allows quick responses for tourists request for information used to know about the place or objects. Several problems with the current tourists using apps like google maps, etc. system have been identified. The current system would benefit from a centralized application for tourists for guiding.

5. PROBLEM STATEMENT

Developing an application that guides tourists in finding places with its well-known information using the guide which isn’t economically feasible for the tourists.

6. SOFTWARE SPECIFICATION REQUIREMENTS

This application is build using the following specifications that helps in the building a guidance application for the tourists for finding out the information by detection.

* Tensor Flow Lite
* Java JDK
* Android Studio IDE
* UI using XML

TensorFlow Lite

TensorFlow Lite is a mobile library for deploying models on mobile, micro-controllers and other edge devices. See the guide. Guides explain the concepts and components of TensorFlow Lite. See examples. Explore TensorFlow Lite Android and iOS apps.

Java JDK

The JDK is a development environment for building applications using the Java programming language. The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java TM platform.

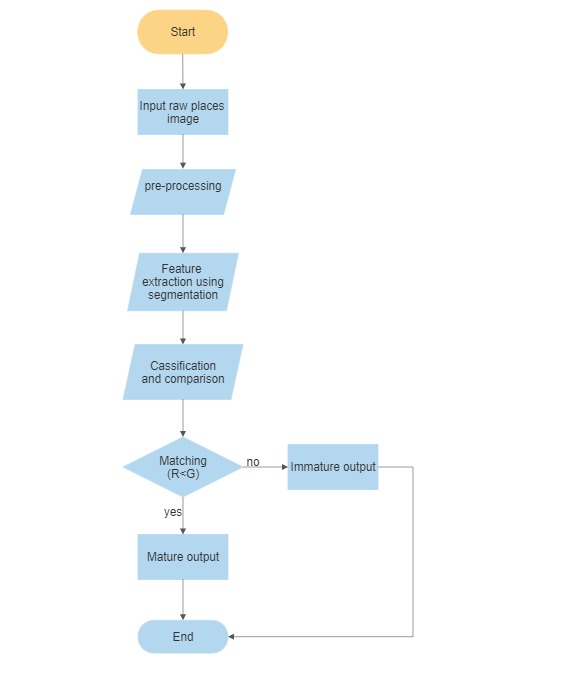
Android Studio IDE

**Android Studio** is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.

XML

Xml (eXtensible Markup Language) is a markup language. it is designed to store and transport data. It is also used for User Interface.

7. Flowchart

****

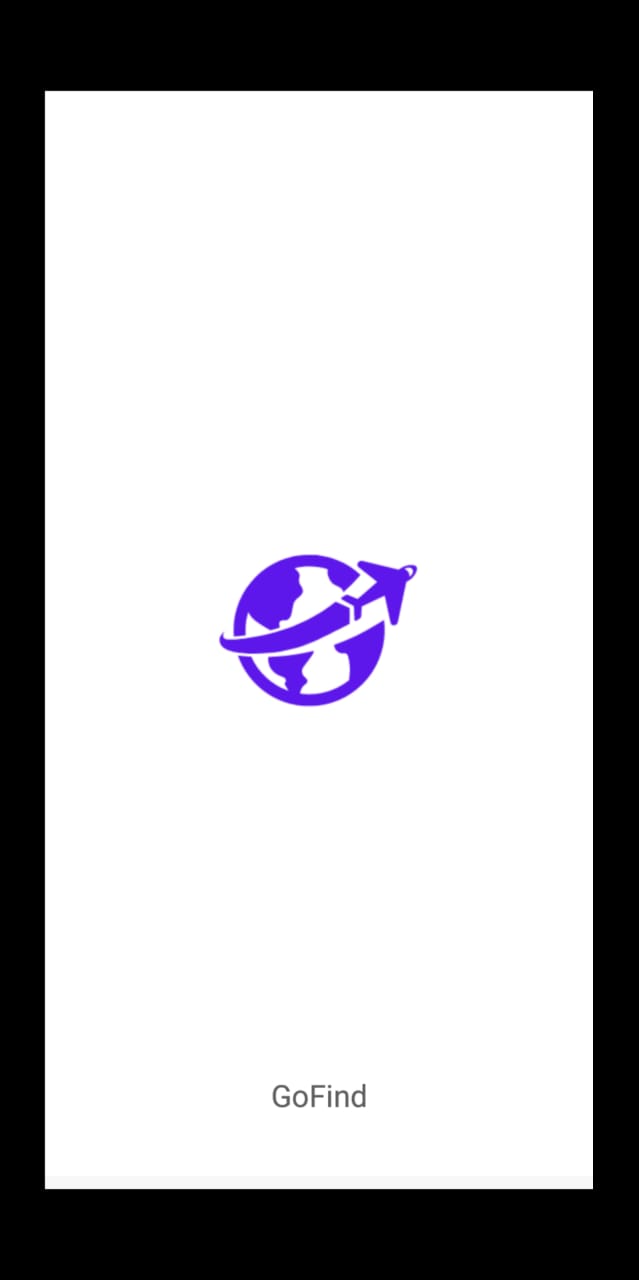
8.CONCLUSION

This application is build using the software specifications like Tensor Flow Lite, Java, Android, UI using xml and its main purpose is that it serves as a guidance app that is used for detecting places and objects just by using camera. This application overcomes any other similar applications drawbacks and additionally gives an information about the places or object using image processing.

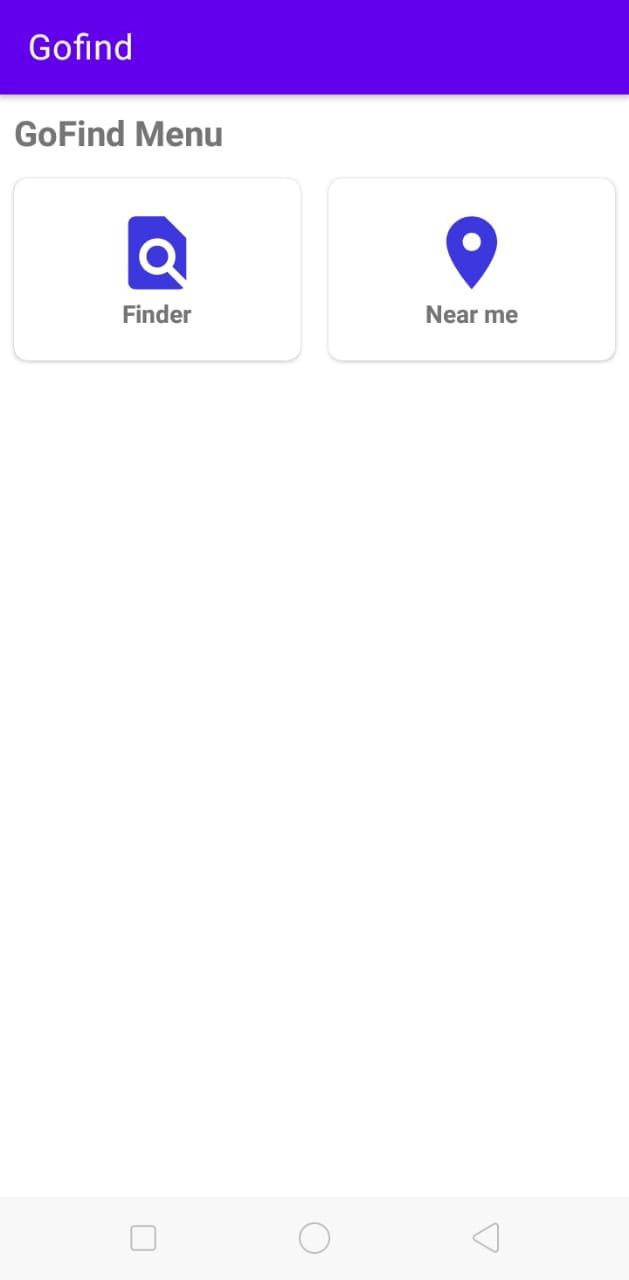
**ANNEXURES**

**9A. SCREENSHOT**

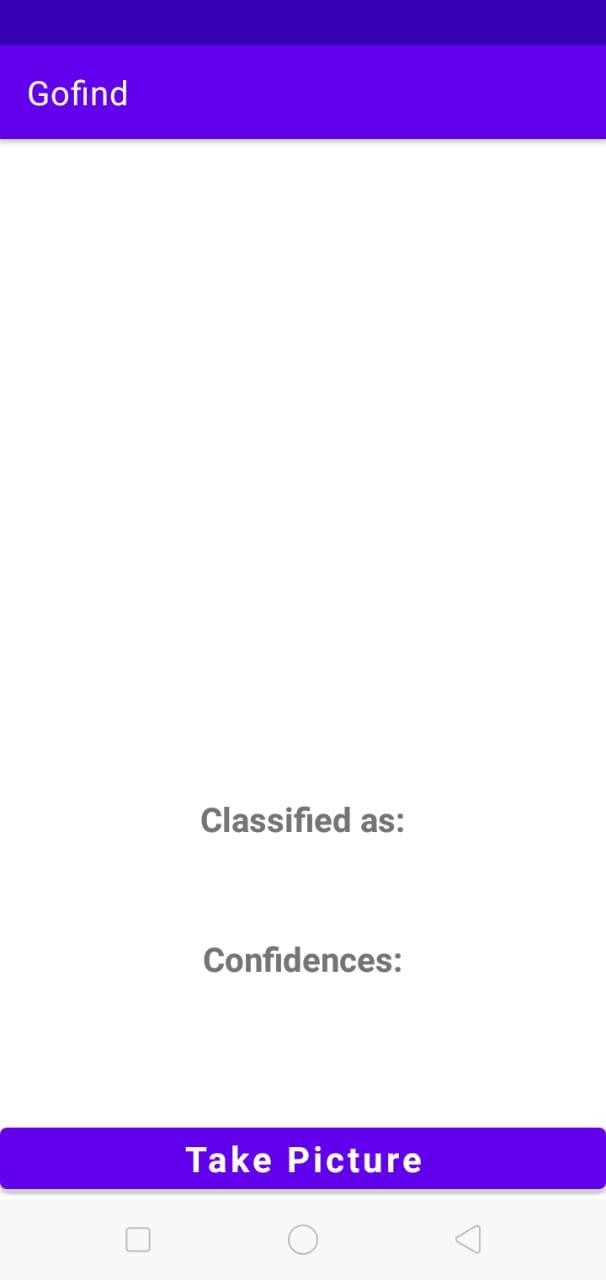
**Splash Screen:**

****

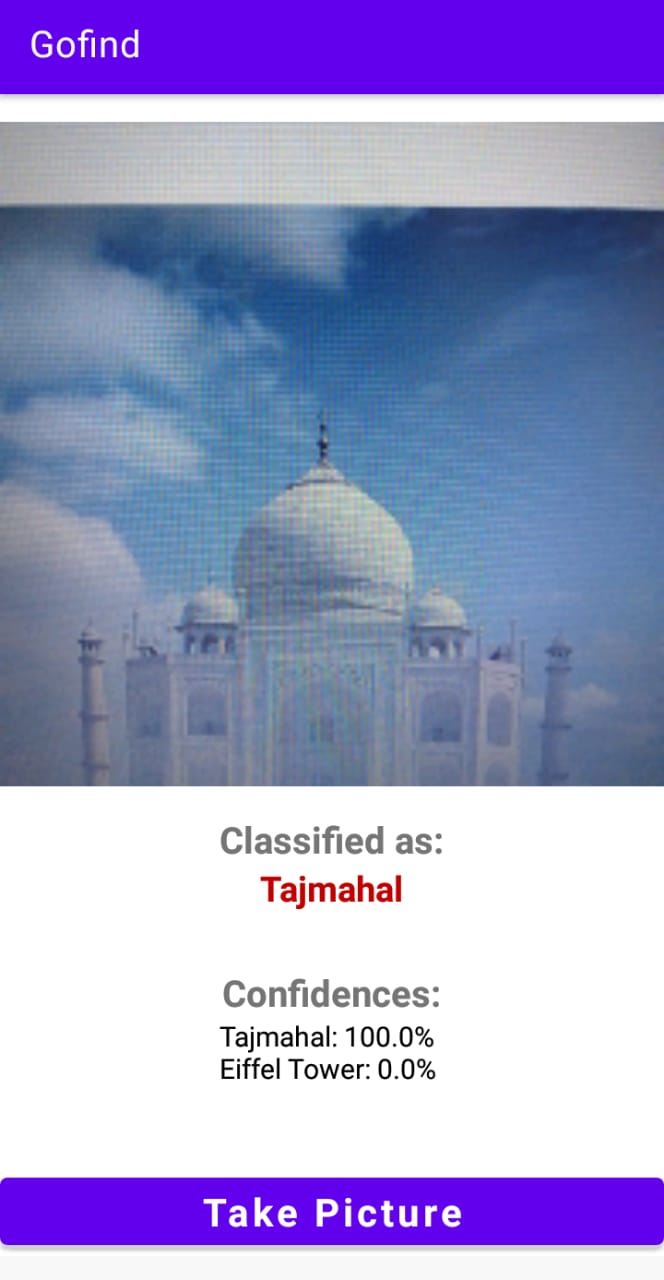
**Dashboard:**



**Place Finder:**



**Place Finder Detection:**

****

**Location:**

**9 B. SOURCE CODE:**

## AndriodManifest.xml

*<?*xml version="1.0" encoding="utf-8"*?>*

<manifest xmlns:android[="http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" xmlns:tools="<http://schemas.android.com/tools>" package="com.example.gofind">

<uses-permission android:name="com.javapapers.android.googleplacesdetail.permission.MAPS\_RECEIVE " />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"

/>

<uses-permission

android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"

/>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-feature

android:glEsVersion="0x00020000" android:required="true" />

<application

android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/Theme.Gofind">

*key.*

*<!--*

*TODO: Before you run your application, you need a Google Maps API*

*To get one, follow the directions here:*

*https://developers.google.com/maps/documentation/android-*

*sdk/get-api-key*

<meta-data

android:name="com.google.android.geo.API\_KEY" android:value=" AIzaSyApgI0pGtHMXSwZLmHPByq6fortd1ICGK4" />

<activity

android:name=".Nearbyplace" android:exported="false"

android:label="@string/title\_activity\_nearbyplace" />

<activity

android:name=".NearmeActivity" android:exported="false" />

<activity

android:name=".PlaceFinder" android:exported="false" />

<activity

android:name=".DashboardScreen" android:exported="false" />

<activity

android:name=".MainActivity" android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

## MainActivity.java

package com.example.gofind;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent; import android.os.Bundle; import android.os.Handler; import android.view.Window;

import android.view.WindowManager;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); requestWindowFeature(Window.*FEATURE\_NO\_TITLE*); getSupportActionBar().hide();

this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);

setContentView(R.layout.*activity\_main*);

new Handler().postDelayed(new Runnable() {

@Overrid

public void run() {

Intent intent = new Intent(MainActivity.this,DashboardScreen.class);

startActivity(intent);

}}

## DashboardScreen.java

package com.example.gofind;

import androidx.appcompat.app.AppCompatActivity; import androidx.cardview.widget.CardView;

import android.content.Intent;

import android.content.pm.ActivityInfo; import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView; import android.widget.GridView; import android.widget.Switch; import android.widget.Toast;

public class DashboardScreen extends AppCompatActivity implements View.OnClickListener{

private CardView card1, card2;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_dashboard\_screen*); this.setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);

card1 = findViewById(R.id.*finder*); card2 = findViewById(R.id.*nearme*);

card1.setOnClickListener(this);

@Override

public void onClick(View view) { Intent i;

switch (view.getId()){ case R.id.*finder*:

i = new Intent(this,PlaceFinder.class); startActivity(i);

break;

case R.id.*nearme*:

i = new Intent(this,Nearbyplace.class); startActivity(i);

break;

}

## PlaceFinder.java

package com.example.gofind;

import androidx.annotation.Nullable; import androidx.annotation.RequiresApi;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;

import android.annotation.SuppressLint; import android.content.Intent;

import android.content.pm.ActivityInfo; import android.content.pm.PackageManager; import android.graphics.Bitmap;

import android.media.ThumbnailUtils; import android.os.Build;

import android.os.Bundle;

import android.provider.MediaStore; import android.view.View;

import android.widget.Button; import android.widget.ImageView; import android.widget.TextView;

import com.example.gofind.ml.Model;

import org.tensorflow.lite.DataType;

import org.tensorflow.lite.support.tensorbuffer.TensorBuffer

import java.io.IOException; import java.nio.ByteBuffer;

public class PlaceFinder extends AppCompatActivity {

TextView result, confidences; ImageView imageView;

Button picture; int imageSize =96;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_place\_finder*); this.setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*); result = findViewById(R.id.*result*);

confidences = findViewById(R.id.*confidence*); imageView = findViewById(R.id.*imageView*); picture = findViewById(R.id.*button*);

picture.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

*// Launch camera if we have permission*

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {

if (checkSelfPermission(Manifest.permission.*CAMERA*) == PackageManager.*PERMISSION\_GRANTED*) {

Intent cameraIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);

startActivityForResult(cameraIntent,1);

} else {

*//Request camera permission if we don't have it.*

requestPermissions(new String[]{Manifest.permission.*CAMERA*}, 100);

}

}

}

});

}

## Nearbyplace.java

package com.example.gofind;

import androidx.annotation.Nullable; import androidx.annotation.RequiresApi;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;

import android.annotation.SuppressLint;

startActivityForResult(cameraIntent,1);

} else {

*//Request camera permission if we don't have it.*

requestPermissions(new String[]{Manifest.permission.*CAMERA*}, 100);

import com.example.gofind.ml.Model;

import org.tensorflow.lite.DataType;

import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;

import java.io.IOException; import java.nio.ByteBuffer;

public class PlaceFinder extends AppCompatActivity {

TextView result, confidences; ImageView imageView;

Button picture; int imageSize =96;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_place\_finder*); this.setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*); result = findViewById(R.id.*result*);

confidences = findViewById(R.id.*confidence*); imageView = findViewById(R.id.*imageView*); picture = findViewById(R.id.*button*);

picture.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {

if (checkSelfPermission(Manifest.permission.*CAMERA*) == PackageManager.*PERMISSION\_GRANTED*) {

Intent cameraIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);

## FetchData.java

public class FetchData extends AsyncTask<Object,String,String> { String googleNearByPlacesData;

GoogleMap googleMap; String url;

@Override

protected String doInBackground(Object... objects) { try{

googleMap=(GoogleMap) objects[0]; url = (String) objects[1];

DownloadUrl downloadUrl = new DownloadUrl(); googleNearByPlacesData= downloadUrl.retireveUrl(url);

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

}

return googleNearByPlacesData

# REFERENCES:

1. <https://www.airbnb.co.in/help/article/84/can-i-refer-someone-to-join-airbnb>
2. [https://help.gasbuddy.com/hc/en-us/articles/115015715587-Referring-a-Friend-](https://help.gasbuddy.com/hc/en-us/articles/115015715587-Referring-a-Friend-For-More-Savings)

For-More-Savings

1. Google Map API:

<https://developers.google.com/maps/documentation>

1. Google lens: <https://www.google.com/search?q=google+lens+system+report&rlz=1C1YTUH_>

enIN1005IN1005&oq=google+lens&aqs=chrome.0.69i59l2j69i57l2j0i271l2j69i64j6

9i60.6517j0j7&sourceid=chrome&ie=UTF-8