AndroidManifest.xml

```
<!--  AndroidManifest.xml  -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.traffic">
   <!--  Permissions  -->
   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
   <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.AppCompat.Light.NoActionBar">
      <!--  Main Activity  -->
      <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>
```

Activity_main.xml

```
<!-- res/layout/activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center"
   android:padding="16dp">

   <TextView
      android:id="@+id/label"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Traffic Sign Classifier"
    android:textSize="24sp"
    android:textStyle="bold"/>

<ImageView
    android:id="@+id/imageView"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_marginTop="20dp"
    android:background="@drawable/ic_launcher_foreground"
    android:contentDescription="Image Preview"/>
<Button
    android:id="@+id/uploadButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Upload Image"
    android:layout_marginTop="20dp"/>

<Button
    android:id="@+id/classify_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Classify Image"
    android:layout_marginTop="20dp"/>

<TextView
    android:id="@+id/resultTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:textSize="18sp"
        android:text="Prediction will appear here"
        android:layout_marginTop="20dp"/>

    <!-- Adding "by Abdul Jameel" at the bottom -->
    <TextView
        android:id="@+id/creditTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="by Abdul Jameel"
        android:textSize="14sp"
        android:layout_marginTop="40dp"/>

</LinearLayout>
```

Main.java

```
package com.example.traffic;

import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import com.example.traffic.ml.TrafficSignModel;
import org.tensorflow.lite.DataType;
import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;
```

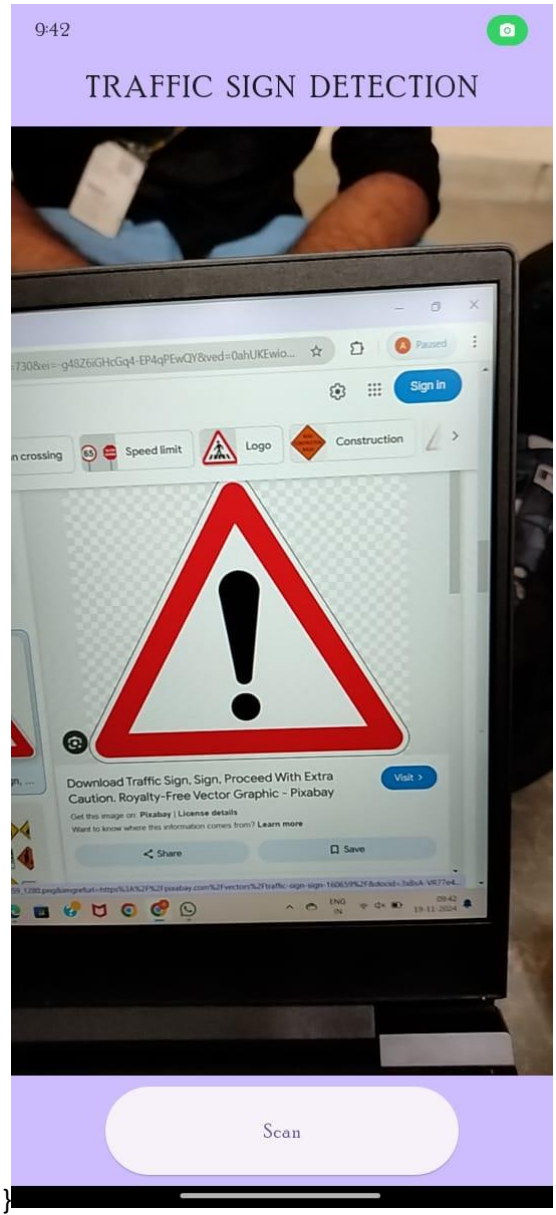
```
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class MainActivity extends AppCompatActivity {

    private static final int SELECT_PICTURE = 1;
    private TrafficSignModel model;
    private ImageView imageView;
    private Bitmap selectedImage;

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

        imageView = findViewById(R.id.imageView);

        try {
            model = TrafficSignModel.newInstance(this);
        } catch (IOException e) {
            Log.e("MainActivity", "Error initializing TensorFlow Lite model.", e);
            Toast.makeText(this, "Model initialization failed.", Toast.LENGTH_SHORT).show();
        }

        // Button to select an image
        Button selectImageButton = findViewById(R.id.uploadButton);
        selectImageButton.setOnClickListener(v -> openImageChooser());

        // Button to classify the selected image
```

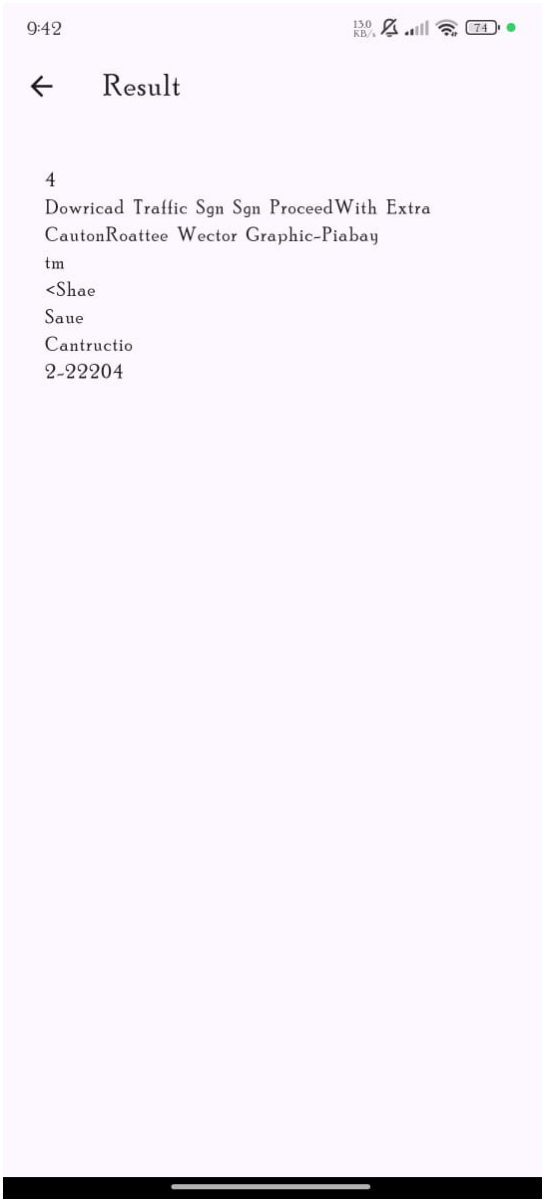
```
        Button classifyButton = findViewById(R.id.classify_button);
        classifyButton.setOnClickListener(v -> {
            if (selectedImage != null && model != null) {
                classifyImage(selectedImage);
            } else {
                Toast.makeText(this, "Please select an image first", Toast.LENGTH_SHORT).show();
            }
        });
    }
import androidx.appcompat.app.AppCompatActivity;
import com.example.traffic.ml.TrafficSignModel;
import org.tensorflow.lite.DataType;
import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class MainActivity extends AppCompatActivity {

    private static final int SELECT_PICTURE = 1;
    private TrafficSignModel model;
    private ImageView imageView;
    private Bitmap selectedImage;

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

        imageView = findViewById(R.id.imageView);
```

```
    try {
        model = TrafficSignModel.newInstance(this);
    } catch (IOException e) {
        Log.e("MainActivity", "Error initializing TensorFlow Lite model.", e);
        Toast.makeText(this, "Model initialization failed.", Toast.LENGTH_SHORT).show();
    }

    // Button to select an image
    Button selectImageButton = findViewById(R.id.uploadButton);
    selectImageButton.setOnClickListener(v -> openImageChooser());

    // Button to classify the selected image
    Button classifyButton = findViewById(R.id.classify_button);
    classifyButton.setOnClickListener(v -> {
        if (selectedImage != null && model != null) {
            classifyImage(selectedImage);
        } else {
            Toast.makeText(this, "Please select an image first", Toast.LENGTH_SHORT).show();
        }
    });
 }
bitmap.getPixels(pixels, 0, 50, 0, 0, 50, 50);

    for (int pixel : pixels) {
        int r = (pixel >> 16) & 0xFF;
        int g = (pixel >> 8) & 0xFF;
        int b = pixel & 0xFF;
        // Normalize pixel values to [0, 1] and add to buffer
        byteBuffer.putFloat(r / 255.0f);
        byteBuffer.putFloat(g / 255.0f);
```

```
            byteBuffer.putFloat(b / 255.0f);
        }
        return byteBuffer;
    }

    private void displayResults(TensorBuffer outputBuffer) {
        float[] outputArray = outputBuffer.getFloatArray();
        int predictedClass = getMaxProbabilityIndex(outputArray);

        // Get the traffic sign label from string resources
        String[] trafficSignLabels = getResources().getStringArray(R.array.traffic_labels);
        String predictedLabel = predictedClass < trafficSignLabels.length ?
trafficSignLabels[predictedClass] : "Unknown class";

        // Display prediction result
        Log.i("MainActivity", "Predicted Class: " + predictedLabel);
        Toast.makeText(this, "Predicted Class: " + predictedLabel, Toast.LENGTH_SHORT).show();
    }


    private int getMaxProbabilityIndex(float[] probabilities) {
        int maxIndex = -1;
        float maxProbability = -1;
        for (int i = 0; i < probabilities.length; i++) {
            if (probabilities[i] > maxProbability) {
                maxProbability = probabilities[i];
                maxIndex = i;
            }
        }
        return maxIndex;
    }
```

```
  @Override
  protected void onDestroy() {
      super.onDestroy();
      if (model != null) {
model.close();
      }
  }
```





}