AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.atharvakale.facerecognition">

  <application
    android:allowBackup="true"
    android:label="@string/app_name"
    android:largeHeap="true"
    android:roundIcon="@drawable/face_icon2"
    android:supportsRtl="true"
    android:theme="@style/Theme.FaceRecognition.NoActionBar">

    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      android:theme="@style/Theme.FaceRecognition.NoActionBar"
      android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

  <uses-permission android:name="android.permission.CAMERA" />
  <uses-feature android:name="android.hardware.camera" android:required="false" />

</manifest>
```

Activity_main.xml

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

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/coordinatorLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/round_bg"
        android:text="ACTIONS"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.25" />

    <FrameLayout
        android:id="@+id/container"
        android:layout_width="297dp"
        android:layout_height="279dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.060000002">
    <Button
        android:id="@+id/button5"
        android:layout_width="68dp"
        android:layout_height="65dp"
        android:layout_marginStart="229dp"
        android:layout_marginEnd="229dp"
        android:background="@drawable/cam"
        android:backgroundTint="@color/teal_200"
        android:elevation="1dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_conversion_absoluteHeight="63dp"
        tools:layout_conversion_absoluteWidth="62dp" />

    <androidx.camera.view.PreviewView
        android:id="@+id/previewView"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.camera.view.PreviewView>
</FrameLayout>

<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="164dp"
    android:layout_marginBottom="168dp"
    android:elevation="2dp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:srcCompat="@android:drawable/ic_input_add" />
```

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="203dp"
    android:layout_height="200dp"
    android:background="#2C7E57C2"
    android:elevation="1dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/container"
    app:layout_constraintVertical_bias="0.435"
    tools:srcCompat="@android:drawable/screen_background_light_transparent" />

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/container">

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/round_bg"
        android:rotation="0"
        android:rotationX="0"
        android:rotationY="0"
        android:text="Add Face"
        android:textAlignment="center"
        android:visibility="visible"
        tools:visibility="visible" />
</LinearLayout>

<TextView
    android:id="@+id/textView"
    android:layout_width="164dp"
    android:layout_height="wrap_content"
    android:hint="Add Face"
    android:textAlignment="center"
    android:textColor="@color/purple_500"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintHorizontal_bias="0.487"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toTopOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.479" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="164dp"
    android:layout_height="180dp"
    android:textColor="@color/black"
    android:textSize="15sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintHorizontal_bias="0.487"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toTopOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.914" />

<TextView
    android:id="@+id/textAbovePreview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:textSize="15sp"
    app:layout_constraintBottom_toTopOf="@+id/imageView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout"
    app:layout_constraintVertical_bias="1.0" />

<!-- Watermark TextView for "by Abdul Jameel" -->
<TextView
    android:id="@+id/textByAbdul Jameel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="by Abdul Jameel"
    android:textColor="#808080"
    android:textSize="12sp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintHorizontal_bias="0.5" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Main.java

```
package com.atharvakale.facerecognition;


import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.content.res.AssetFileDescriptor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.ImageFormat;
import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.RectF;
import android.graphics.YuvImage;
import android.media.Image;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AlertDialog;
import androidx.camera.core.CameraSelector;
import androidx.camera.core.ImageAnalysis;
import androidx.camera.core.ImageProxy;
import androidx.camera.core.Preview;
import androidx.camera.lifecycle.ProcessCameraProvider;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.common.util.concurrent.ListenableFuture;


import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.google.mlkit.vision.common.InputImage;
import com.google.mlkit.vision.face.Face;
import com.google.mlkit.vision.face.FaceDetection;
import com.google.mlkit.vision.face.FaceDetector;
import com.google.mlkit.vision.face.FaceDetectorOptions;

import androidx.appcompat.app.AppCompatActivity;
import androidx.camera.view.PreviewView;
import androidx.core.content.ContextCompat;
import androidx.lifecycle.LifecycleOwner;

import android.os.ParcelFileDescriptor;
import android.text.InputType;
import android.util.Pair;

import android.util.Size;

import android.view.View;


import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;


import org.tensorflow.lite.Interpreter;


import java.io.ByteArrayOutputStream;

import java.io.FileDescriptor;

import java.io.FileInputStream;

import java.io.IOException;

import java.nio.ByteBuffer;

import java.nio.ByteOrder;

import java.nio.MappedByteBuffer;

import java.nio.ReadOnlyBufferException;

import java.nio.channels.FileChannel;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.concurrent.ExecutionException;

import java.util.concurrent.Executor;

import java.util.concurrent.Executors;


public class MainActivity extends AppCompatActivity {
    FaceDetector detector;

    private ListenableFuture<ProcessCameraProvider> cameraProviderFuture;
    PreviewView previewView;
    ImageView face_preview;
    Interpreter tfLite;
    TextView reco_name,preview_info,textAbove_preview;
    Button recognize,camera_switch, actions;
    ImageButton add_face;
    CameraSelector cameraSelector;
    boolean developerMode=false;
    float distance= 1.0f;
    boolean start=true,flipX=false;
    Context context=MainActivity.this;
    int cam_face=CameraSelector.LENS_FACING_BACK; //Default Back Camera

    int[] intValues;
    int inputSize=112;  //Input size for model
    boolean isModelQuantized=false;
    float[][] embeedings;
    float IMAGE_MEAN = 128.0f;
    float IMAGE_STD = 128.0f;
    int OUTPUT_SIZE=192; //Output size of model
    private static int SELECT_PICTURE = 1;
    ProcessCameraProvider cameraProvider;
    private static final int MY_CAMERA_REQUEST_CODE = 100;

    String modelFile="mobile_face_net.tflite"; //model name

    private HashMap<String, SimilarityClassifier.Recognition> registered = new HashMap<>(); //saved
Faces
```

```
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        registered=readFromSP(); //Load saved faces from memory when app starts
        setContentView(R.layout.activity_main);
        face_preview =findViewById(R.id.imageView);
        reco_name =findViewById(R.id.textView);
        preview_info =findViewById(R.id.textView2);
        textAbove_preview =findViewById(R.id.textAbovePreview);
        add_face=findViewById(R.id.imageButton);
        add_face.setVisibility(View.INVISIBLE);

        SharedPreferences sharedPref = getSharedPreferences("Distance",Context.MODE_PRIVATE);
        distance = sharedPref.getFloat("distance",1.00f);

        face_preview.setVisibility(View.INVISIBLE);
        recognize=findViewById(R.id.button3);
        camera_switch=findViewById(R.id.button5);
        actions=findViewById(R.id.button2);
        textAbove_preview.setText("Recognized Face:");
//        preview_info.setText("        Recognized Face:");
        //Camera Permission
        if (checkSelfPermission(Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[]{Manifest.permission.CAMERA},
MY_CAMERA_REQUEST_CODE);
        }
        //On-screen Action Button
        actions.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
      public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Select Action:");

        // add a checkbox list
        String[] names= {"View Recognition List","Update Recognition List","Save
Recognitions","Load Recognitions","Clear All Recognitions","Import Photo
(Beta)","Hyperparameters","Developer Mode"};

        builder.setItems(names, new DialogInterface.OnClickListener() {
          @Override
          public void onClick(DialogInterface dialog, int which) {

            switch (which)
            {
              case 0:
                displaynameListview();
                break;
              case 1:
                updatenameListview();
                break;
              case 2:
                insertToSP(registered,0); //mode: 0:save all, 1:clear all, 2:update all
                break;
              case 3:
                registered.putAll(readFromSP());
                break;
              case 4:
                clearnameList();
                break;
              case 5:
```

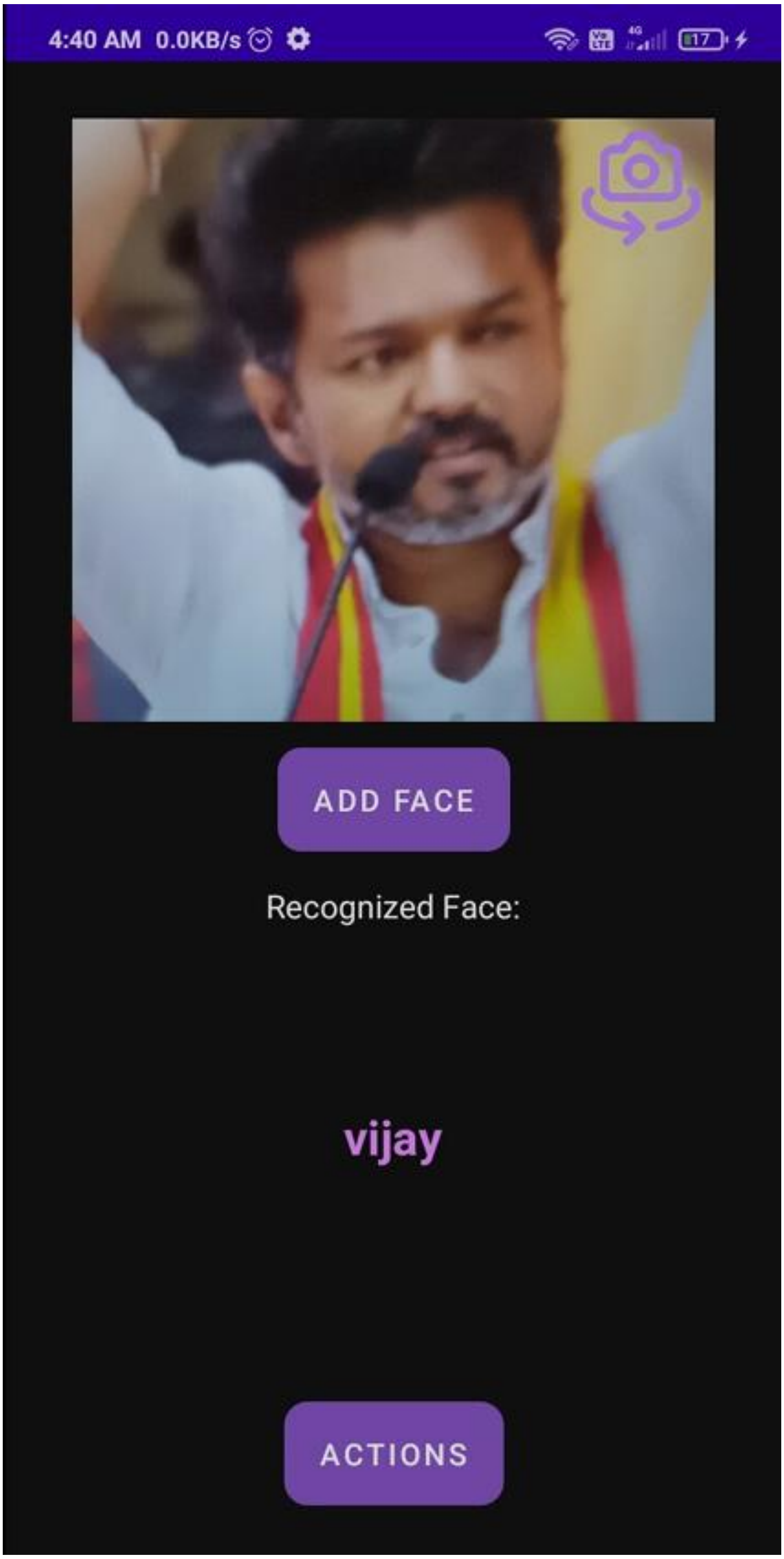
```
                        loadphoto();
                        break;
                    case 6:
                        testHyperparameter();
                        break;
                    case 7:
                        developerMode();
                        break;
                }

            }
        });



        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
        });
        builder.setNegativeButton("Cancel", null);

        // create and show the alert dialog
        AlertDialog dialog = builder.create();
        dialog.show();
    }
});

//On-screen switch to toggle between Cameras.
camera_switch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (cam_face==CameraSelector.LENS_FACING_BACK) {
            cam_face = CameraSelector.LENS_FACING_FRONT;
            flipX=true;
        }
        else {
            cam_face = CameraSelector.LENS_FACING_BACK;
            flipX=false;
        }
        cameraProvider.unbindAll();
        cameraBind();
    }
});

add_face.setOnClickListener((new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        addFace();
    }
}));



recognize.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(recognize.getText().toString().equals("Recognize"))
        {
         start=true;
         textAbove_preview.setText("Recognized Face:");
        recognize.setText("Add Face");
```

```
            add_face.setVisibility(View.INVISIBLE);
            reco_name.setVisibility(View.VISIBLE);
            face_preview.setVisibility(View.INVISIBLE);
            preview_info.setText("");
            //preview_info.setVisibility(View.INVISIBLE);
            }
            else
            {
              textAbove_preview.setText("Face Preview: ");
              recognize.setText("Recognize");
              add_face.setVisibility(View.VISIBLE);
              reco_name.setVisibility(View.INVISIBLE);
              face_preview.setVisibility(View.VISIBLE);
              preview_info.setText("1.Bring Face in view of Camera.\n\n2.Your Face preview will
appear here.\n\n3.Click Add button to save face.");


            }

          }
        });

        //Load model
        try {
          tfLite=new Interpreter(loadModelFile(MainActivity.this,modelFile));
        } catch (IOException e) {
          e.printStackTrace();
        }
        //Initialize Face Detector
        FaceDetectorOptions highAccuracyOpts =
            new FaceDetectorOptions.Builder()
```

```
                .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_ACCURATE)
                .build();
        detector = FaceDetection.getClient(highAccuracyOpts);

        cameraBind();




    }
    private void testHyperparameter()
    {

        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Select Hyperparameter:");

        // add a checkbox list
        String[] names= {"Maximum Nearest Neighbour Distance"};

        builder.setItems(names, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                switch (which)
                {
                    case 0:
//                        Toast.makeText(context, "Clicked", Toast.LENGTH_SHORT).show();
                        hyperparameters();
                        break;

                }
            }


            });
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {


            }
        });
        builder.setNegativeButton("Cancel", null);


        // create and show the alert dialog
        AlertDialog dialog = builder.create();
        dialog.show();
    }
    private void developerMode()
    {
        if (developerMode) {
            developerMode = false;
            Toast.makeText(context, "Developer Mode OFF", Toast.LENGTH_SHORT).show();
        }
        else {
            developerMode = true;
            Toast.makeText(context, "Developer Mode ON", Toast.LENGTH_SHORT).show();
        }
    }
    private void addFace()
    {
        {


            start=false;
```

```
AlertDialog.Builder builder = new AlertDialog.Builder(context);
builder.setTitle("Enter Name");

    // Set up the input
final EditText input = new EditText(context);

input.setInputType(InputType.TYPE_CLASS_TEXT );
builder.setView(input);

    // Set up the buttons
builder.setPositiveButton("ADD", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //Toast.makeText(context, input.getText().toString(), Toast.LENGTH_SHORT).show();

        //Create and Initialize new object with Face embeddings and Name.
        SimilarityClassifier.Recognition result = new SimilarityClassifier.Recognition(
                "0", "", -1f);
        result.setExtra(embeedings);

        registered.put( input.getText().toString(),result);
        start=true;

    }
});
builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        start=true;
        dialog.cancel();
    }
```

```
        });


        builder.show();
    }
}
private  void clearnameList()
{
    AlertDialog.Builder builder =new AlertDialog.Builder(context);
    builder.setTitle("Do you want to delete all Recognitions?");
    builder.setPositiveButton("Delete All", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            registered.clear();
            Toast.makeText(context, "Recognitions Cleared", Toast.LENGTH_SHORT).show();
        }
    });
    insertToSP(registered,1);
    builder.setNegativeButton("Cancel",null);
    AlertDialog dialog = builder.create();
    dialog.show();
}
private void updatenameListview()
{
    AlertDialog.Builder builder = new AlertDialog.Builder(context);
    if(registered.isEmpty()) {
        builder.setTitle("No Faces Added!!");
        builder.setPositiveButton("OK",null);
    }
    else{
        builder.setTitle("Select Recognition to delete:");
    // add a checkbox list
    String[] names= new String[registered.size()];
    boolean[] checkedItems = new boolean[registered.size()];
     int i=0;
        for (Map.Entry<String, SimilarityClassifier.Recognition> entry : registered.entrySet())
        {
          //System.out.println("NAME"+entry.getKey());
          names[i]=entry.getKey();
          checkedItems[i]=false;
          i=i+1;

        }

        builder.setMultiChoiceItems(names, checkedItems, new
DialogInterface.OnMultiChoiceClickListener() {
          @Override
          public void onClick(DialogInterface dialog, int which, boolean isChecked) {
            // user checked or unchecked a box
            //Toast.makeText(MainActivity.this, names[which], Toast.LENGTH_SHORT).show();
           checkedItems[which]=isChecked;

          }
        });



    builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialog, int which) {

           // System.out.println("status:"+ Arrays.toString(checkedItems));
            for(int i=0;i<checkedItems.length;i++)
            {
              //System.out.println("status:"+checkedItems[i]);
              if(checkedItems[i])
              {
//                  Toast.makeText(MainActivity.this, names[i], Toast.LENGTH_SHORT).show();
                registered.remove(names[i]);
              }

            }
        insertToSP(registered,2); //mode: 0:save all, 1:clear all, 2:update all
        Toast.makeText(context, "Recognitions Updated", Toast.LENGTH_SHORT).show();
      }
    });
    builder.setNegativeButton("Cancel", null);

    // create and show the alert dialog
    AlertDialog dialog = builder.create();
    dialog.show();
  }
  }
  private void hyperparameters()
  {
    AlertDialog.Builder builder = new AlertDialog.Builder(context);
    builder.setTitle("Euclidean Distance");
    builder.setMessage("0.00 -> Perfect Match\n1.00 -> Default\nTurn On Developer Mode to find optimum value\n\nCurrent Value:");
    // Set up the input
    final EditText input = new EditText(context);

    input.setInputType(InputType.TYPE_CLASS_NUMBER | InputType.TYPE_NUMBER_FLAG_DECIMAL);
    builder.setView(input);
    SharedPreferences sharedPref = getSharedPreferences("Distance",Context.MODE_PRIVATE);
    distance = sharedPref.getFloat("distance",1.00f);
    input.setText(String.valueOf(distance));
    // Set up the buttons
    builder.setPositiveButton("Update", new DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialog, int which) {
        //Toast.makeText(context, input.getText().toString(), Toast.LENGTH_SHORT).show();

       distance= Float.parseFloat(input.getText().toString());



        SharedPreferences sharedPref =
getSharedPreferences("Distance",Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPref.edit();
        editor.putFloat("distance", distance);
        editor.apply();


      }
    });
    builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialog
```



Recognized Face:

vijay