Home » Android » Android Face Detection Example

Andr ction Example

28 May, 201 on / Mobile Vision API by Mohit Gupt (updated on October 7, 2017)

Nowadays face filter apps are one of the most common apps available on a users phone, where a user can apply various funny filters on their face pictures. Interestingly the base tech that powers these sort of apps is similar to what we are going to discuss in this tutorial. That's right we are going to discuss an Android Face Detection API. Interestingly to power all these apps, officially Google has released an Android Face Detection API in their Mobile Vision set of APIs. Basically all the face filter apps detect a face through a face detection API, and apply various overlays on the selected picture. Although when speaking of Face Detection APIs for Android, we have multiple options. But for this article we will be discussing Google backed Mobile Vision APIs only. As its fast and has deeply integrated Android development SDK/APIs.

Android Face Detection API – Mobile Vision

When speaking of face detection, it is often misunderstood by face recognition, therefore let me put it discretely. As of now Mobile Vision APIs do not support Face Recognition, right now they only support face detection. Although it does have the ability to identify the characteristics of a face, which includes eyes, nose, mouth and smile etc. But besides these characteristics Mobile Vision Face Detection API can also track a face in a video sequence, which is again not an application of face recognition, but face detection as it is identified by tracking the movement of that particular face in the sequence. Also since this API is a part of Google's Play Services library, it is not bundled as a part of the APK, instead an ad-on package is downloaded internally by the play services itself if it is not present to support Android face detection. To do so, we need to set up the play services with the mobile vision dependency in build.gradle file as shown below:

build.gradle (app)

```
compile 'com.google.android.gms:play-services-vision:11.4.0'
```

Also the bu ve this:

build.gradle(project)

```
allprojects {
    repositories {
        jcenter()
        maven {
            url "https://maven.google.com"
        }
    }
}
```

Post this a `<meta-data>` tag for face detection api along with `<uses-feature>` and `<uses-permission>` tag for camera and accessing external storage needs to be added in the manifest as shown below:

AndroidManifest.xml XHTML

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

    <uses-feature
        android:name="android.hardware.camera"
        android:required="true"/>
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <meta-data
            android:name="com.google.android.gms.vision.DEPENDENCIES"
            android:value="face"/>

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities="${applicationId}.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths"/>
        </provider>
    </application>

</manifest>
```

Android Face Detection Library – Features

Next lets understand the features of Mobile Vision Face Detection APIs. As amazing things can be done by correct usage of this API. Broadly speaking, this API can perform detailed facial analysis- like detection of facial features and identifying their states. Currently its features can be divided into three categories shown below:

1. Landmark Detection: One of the most important features of Mobile Vision Face Detection APIs is the detection of facial landmarks. In case you are wondering what are facial landmarks? They are the basic facial features or points of interest, like Nose, Eyes, and Mouth etc. Interestingly these features are detected very accurately by this Android Face Detection API. As of now following facial features are detected by this API:

- Landr
- Landr
- Landr
- Landr
- Landr
- Landr
- Landr
- Landr
- Landmark.RIGHT_EAR_TIP
- Landmark.RIGHT_EAR
- Landmark.RIGHT_EYE
- Landmark.RIGHT_MOUTH

2. Facial Classification: Interestingly this API can also apply some logic on the detected facial landmarks and identify facial classifications. For ex. it can be detected through this API if the detected face has its eyes open or is smiling or not. Although this feature set may not sound very impressive as of now; but its very accurate and has a lot of room to grow. Maybe in near future some additional features might come in.

3. Face Tracking: Another very powerful feature of *Mobile Vision Face Detection APIs* is this face tracking feature. It simply gives us the ability to track a face in a video sequence. Once again, here I would like to clarify that this is not Face Recognition, this feature simply works on face detection only. As it tracks that face based on its movement in the consecutive video frames.

Android Face Detection API – Example

Now that we have a basic understanding of how the Face Detection APIs work, here in this section we would build a short example where we showcase its capabilities. Since Android Face Detection is itself a huge topic we would limit the scope of this tutorial, and showcase the Facial Classification feature with Landmark Detection only. Also this would solve our primary use case of Face Detection. Therefore for this Android Face Detection Example we would simply take a picture from a camera and run face detection on it, by using the Mobile Vision Face Detection APIs. To start building, lets continue from the steps mentioned in the first section of this article and define a layout to take a picture as shown below:

activity_main.xml XHTML

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    android:id="@+id/activity_main"
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.truiton.mobile.vision.facedetection.MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="70dp"
        android:layout_height="70dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:srcCompat="@mipmap/truiton"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:text="Scan Face"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        tools:layout_constraintLeft_creator="1"
        tools:layout_constraintRight_creator="1"
        />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Scan Results:"
        android:textAllCaps="false"
        android:textStyle="normal|bold"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_constraintLeft_creator="1"
        tools:layout_constraintRight_creator="1"/>

    <ScrollView
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="8dp"
        android:paddingLeft="5dp"
        android:paddingRight="5dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="1.0"
        tools:layout_constraintBottom_creator="1"
        tools:layout_constraintLeft_creator="1"
        tools:layout_constraintRight_creator="1"
        tools:layout_constraintTop_creator="1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/results"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="8dp"/>

            <ImageView
                android:id="@+id/scannedResults"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:layout_marginBottom="8dp"
                android:layout_marginTop="8dp"/>
        </LinearLayout>
    </ScrollView>
</android.support.constraint.ConstraintLayout>
```

Here for this layout I have used `ConstraintLayout` as my root layout, but its not mandatory to use this for your activity's layout. You can also use the normal `RelativeLayout` or `LinearLayout` as per your needs, but if you wish to use `ConstraintLayout`, please don't forget to add its dependency in your build.gradle file, as shown below:

```
build.gradle (app)
compile 'com.android.support.constraint:constraint-layout:1.0.2'
```

Also full source code is available at the end of this tutorial. Next lets define the MainActivity for this Android Face Detection tutorial.

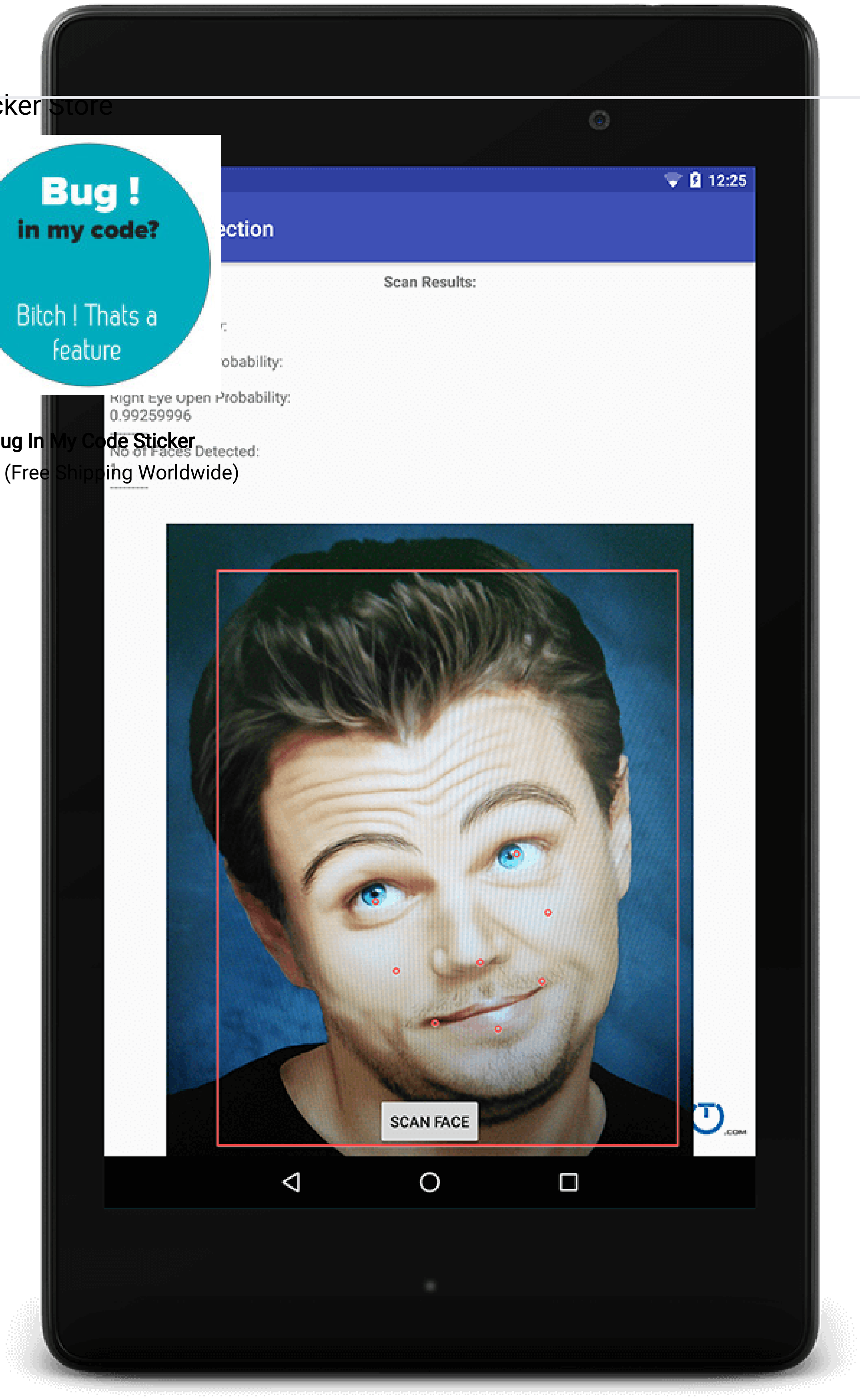
```
MainActivity.java                                                    Java
```

```java
package com.truiton.mobile.vision.facedetection;


import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.FileProvider;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.util.SparseArray;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.vision.Frame;
import com.google.android.gms.vision.face.Face;
import com.google.android.gms.vision.face.FaceDetector;
import com.google.android.gms.vision.face.Landmark;

import java.io.File;
import java.io.FileNotFoundException;

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = "FACE API";
    private static final int PHOTO_REQUEST = 10;
    private TextView scanResults;
    private ImageView imageView;
    private Uri imageUri;
    private FaceDetector detector;
    private static final int REQUEST_WRITE_PERMISSION = 20;
    private static final String SAVED_INSTANCE_URI = "uri";
    private static final String SAVED_INSTANCE_BITMAP = "bitmap";
    private static final String SAVED_INSTANCE_RESULT = "result";
    Bitmap editedBitmap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button) findViewById(R.id.button);
        scanResults = (TextView) findViewById(R.id.results);
        imageView = (ImageView) findViewById(R.id.scannedResults);
        if (savedInstanceState != null) {
            editedBitmap = savedInstanceState.getParcelable(SAVED_INSTANCE_BITMAP);
            if (savedInstanceState.getString(SAVED_INSTANCE_URI) != null) {
                imageUri = Uri.parse(savedInstanceState.getString(SAVED_INSTANCE_URI));
            }
            imageView.setImageBitmap(editedBitmap);
            scanResults.setText(savedInstanceState.getString(SAVED_INSTANCE_RESULT));
        }
        detector = new FaceDetector.Builder(getApplicationContext())
                .setTrackingEnabled(false)
                .setLandmarkType(FaceDetector.ALL_LANDMARKS)
                .setClassificationType(FaceDetector.ALL_CLASSIFICATIONS)
                .build();
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ActivityCompat.requestPermissions(MainActivity.this, new
                        String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, REQUEST_WRITE_PERMISSION)
            }
        });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
            case REQUEST_WRITE_PERMISSION:
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    takePicture();
                } else {
                    Toast.makeText(MainActivity.this, "Permission Denied!", Toast.LENGTH_SHORT).show();
                }
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == PHOTO_REQUEST && resultCode == RESULT_OK) {
            launchMediaScanIntent();
            try {
                scanFaces();
            } catch (Exception e) {
                Toast.makeText(this, "Failed to load Image", Toast.LENGTH_SHORT).show();
                Log.e(LOG_TAG, e.toString());
            }
        }
    }

    private void scanFaces() throws Exception {
        Bitmap bitmap = decodeBitmapUri(this, imageUri);
        if (detector.isOperational() && bitmap != null) {
            editedBitmap = Bitmap.createBitmap(bitmap.getWidth(), bitmap
                    .getHeight(), bitmap.getConfig());
            float scale = getResources().getDisplayMetrics().density;
            Paint paint = new Paint(Paint.ANTI_ALIAS_FLAG);
            paint.setColor(Color.rgb(255, 61, 61));
            paint.setTextSize((int) (14 * scale));
            paint.setShadowLayer(1f, 0f, 1f, Color.WHITE);
            paint.setStyle(Paint.Style.STROKE);
            paint.setStrokeWidth(3f);
            Canvas canvas = new Canvas(editedBitmap);
            canvas.drawBitmap(bitmap, 0, 0, paint);
            Frame frame = new Frame.Builder().setBitmap(editedBitmap).build();
            SparseArray<Face> faces = detector.detect(frame);
            scanResults.setText(null);
            for (int index = 0; index < faces.size(); ++index) {
                Face face = faces.valueAt(index);
                canvas.drawRect(
                        face.getPosition().x,
                        face.getPosition().y,
                        face.getPosition().x + face.getWidth(),
                        face.getPosition().y + face.getHeight(), paint);
                scanResults.setText(scanResults.getText() + "Face " + (index + 1) + "\n");
                scanResults.setText(scanResults.getText() + "Smile probability:" + "\n");
                scanResults.setText(scanResults.getText() + String.valueOf(face.getIsSmilingProbability
                scanResults.setText(scanResults.getText() + "Left Eye Open Probability: " + "\n");
                scanResults.setText(scanResults.getText() + String.valueOf(face.getIsLeftEyeOpenProbabi
                scanResults.setText(scanResults.getText() + "Right Eye Open Probability: " + "\n");
                scanResults.setText(scanResults.getText() + String.valueOf(face.getIsRightEyeOpenProbab
                scanResults.setText(scanResults.getText() + "---------" + "\n");

                for (Landmark landmark : face.getLandmarks()) {
                    int cx = (int) (landmark.getPosition().x);
                    int cy = (int) (landmark.getPosition().y);
                    canvas.drawCircle(cx, cy, 5, paint);
                }
            }

            if (faces.size() == 0) {
                scanResults.setText("Scan Failed: Found nothing to scan");
            } else {
                imageView.setImageBitmap(editedBitmap);
                scanResults.setText(scanResults.getText() + "No of Faces Detected: " + "\n");
                scanResults.setText(scanResults.getText() + String.valueOf(faces.size()) + "\n");
                scanResults.setText(scanResults.getText() + "---------" + "\n");
            }
        } else {
            scanResults.setText("Could not set up the detector!");
        }
    }

    private void takePicture() {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        File photo = new File(Environment.getExternalStorageDirectory(), "picture.jpg");
        imageUri = FileProvider.getUriForFile(MainActivity.this,
                BuildConfig.APPLICATION_ID + ".provider", photo);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
        startActivityForResult(intent, PHOTO_REQUEST);
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        if (imageUri != null) {
            outState.putParcelable(SAVED_INSTANCE_BITMAP, editedBitmap);
            outState.putString(SAVED_INSTANCE_URI, imageUri.toString());
            outState.putString(SAVED_INSTANCE_RESULT, scanResults.getText().toString());
        }
        super.onSaveInstanceState(outState);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        detector.release();
    }

    private void launchMediaScanIntent() {
        Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
        mediaScanIntent.setData(imageUri);
        this.sendBroadcast(mediaScanIntent);
    }

    private Bitmap decodeBitmapUri(Context ctx, Uri uri) throws FileNotFoundException {
        int targetW = 600;
        int targetH = 600;
        BitmapFactory.Options bmOptions = new BitmapFactory.Options();
        bmOptions.inJustDecodeBounds = true;
        BitmapFactory.decodeStream(ctx.getContentResolver().openInputStream(uri), null, bmOptions);
        int photoW = bmOptions.outWidth;
        int photoH = bmOptions.outHeight;

        int scaleFactor = Math.min(photoW / targetW, photoH / targetH);
        bmOptions.inJustDecodeBounds = false;
        bmOptions.inSampleSize = scaleFactor;

        return BitmapFactory.decodeStream(ctx.getContentResolver()
                .openInputStream(uri), null, bmOptions);
    }
}
```

In the above piece of code, in `onCreate` method I have simply initialized the face detector by calling the `FaceDetector.Builder(getApplicationContext())` builder. This would download the Google Play Service dependencies for performing face detection and initialize them. In a way this also works as a safety measure to download the dependencies even when we specified the app to download the dependencies for face detection in the manifest (shown in first step). Also to make it event more reliable, we have also put a check; to check whether the detector is operational or not just before scanning the actual image in `scanFaces()` method. Full source code is available here:

Full Source Code

Also as you can see above we have initialized the Mobile Vision Face Detector with two capabilities, i.e. `setLandmarkType(FaceDetector.ALL_LANDMARKS)` and `setClassificationType(FaceDetector.ALL_CLASSIFICATIONS)`. This would identify all the facial landmarks and classifications on a detected face, rest of the code just shows how we have plotted it on screen, which is self explanatory. The end result would look something like this:



Scan Results:

obability:

Right Eye Open Probability:
0.99259996

No of Faces Detected:

Additional Capabilities

In addition to all what we have discussed above in this Android Face Detection Example, there is one more capability present in these Face Detection APIs. That is the face tracking capability with MultiProcessor. The great thing about this feature is that it can not only track a single face, but can also track multiple faces in a video sequence. But due to limited scope of this article it is not covered here. Also since the face detection APIs are a part of Google's Mobile Vision Suite, we have the capability to build a multi detector. Where we can track multiple faces and multiple bar codes or QR codes in a single video sequence by using the MultiDetector class. This feature is something very new and very powerful which opens a whole new area to explore into. Connect with us on Twitter, Facebook and Google+ for more updates on this.

You may a

Introducir API

Android Material Design With Backward Compatibility

Android Chart Example: MP Android Chart library

Join Us

About Mohit Gupt

An android enthusiast, and an iPhone user with a keen interest in development of innovative applications.

Web | Twitter | Facebook | Google+ | More Posts (78)

Share this:

41

More

Leave a comment

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

I'r

reCAPTCHA
Privacy - Terms

Post Comment

□ Notify me of new posts by email.

4 thoughts on “Android Face Detection Example”

pourya

Reply ↓

August 9, 2017 at 11:53 am

hey mohit
thank you for this code
there one question and that is can this app run in background?i mean can we extend it using broadcast recever and service to detect face?
i need it for a specific project
any help will be appriciated

Ignacio Maturano

Reply ↓

November 8, 2017 at 6:21 pm

gracias hermano, muchas cosas q no conocia. CAPOOOOO!!!!

hana

Reply ↓

November 14, 2017 at 1:43 pm

Thank you for your source.
But there is one problem… I get the error ‘scan failed: found nothing to scan’ after taking a picture. What is the problem? Can you tell me where to fix it?

Edwine

Reply ↓

March 2, 2018 at 1:34 pm

Hae Mohit how can someone write a camera program that can identify the person capture and show some of his/her details e.g name or ID number

Post navigation

← Android Bottom Navigation Bar Example

Introducing Android Mobile Vision API →

Recent Posts

- Intr n API
- Anc
- Anc Example
- Opt Android – OCR
- Anc ically Scan QR Code and Bar Code

Rec

- Vempati Satya Suryanarayana on Android RecyclerView Tutorial
- Amigo on Android Bottom Navigation Bar Example
- Karuppasamy M on Android Places API: Autocomplete with getPlaceByID
- karuppasamy M on Obtaining SHA1 Fingerprint from Android Keystore
- Shivaram Ganesan on How To Stream RTMP live in Android

Archives

- May 2017
- January 2017
- November 2016
- September 2016
- July 2016
- June 2016
- April 2016
- March 2016
- December 2015
- June 2015
- May 2015
- April 2015
- March 2015
- February 2015
- January 2015
- December 2014
- November 2014
- October 2014
- September 2014
- August 2014
- June 2014
- May 2014
- January 2014
- November 2013
- August 2013
- July 2013
- June 2013

- May 2013
- April 2013
- March 2013

Funny Sticker Store

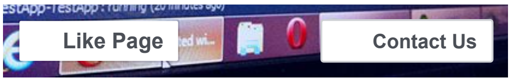
- October 2012

Cat

- Andı
- iPho
- New
- Unca

Follow Truiton

Bug In My Code Sticker
$2.99 (Free Shipping Worldwide)



Truiton

google.com/+Truitonline

Technology Reaching Us In Time - Online

G+ Follow

Follow @truitonline

Follow Mohit Gupt

Recent Comments

- Vempati Satya Suryanarayana on Android RecyclerView Tutorial
- Amigo on Android Bottom Navigation Bar Example
- Karuppasamy M on Android Places API: Autocomplete with getPlaceByID
- karuppasamy M on Obtaining SHA1 Fingerprint from Android Keystore
- Shivaram Ganesan on How To Stream RTMP live in Android

Recent Posts

- Introducing Android Mobile Vision API
- Android Face Detection Example
- Android Bottom Navigation Bar Example
- Optical Character Recognition on Android – OCR
- Android Example – Programmatically Scan QR Code and Bar Code

Search Truiton.com

Search

· Privacy Policy | Join Us ·

Funny Sticker Store

Bug In My Code Sticker

$2.99 (Free Shipping Worldwide)