



# Android Developers Blog

13 AUGUST 2015



## SEARCH

## ARCHIVE

- ▶ 2016 (72)
- ▼ 2015 (130)
  - ▶ December (9)
  - ▶ November (13)
  - ▶ October (20)
  - ▶ September (15)
  - ▼ August (13)
    - Games developer, Dots, share their Do's and Don'ts...
    - Announcing the Android Auto Desktop Head Unit
    - Building better apps with Runtime Permissions
    - Get the Do's and Don'ts for Notifications from Gam...
    - Hungry for some Big Android BBQ?
    - Interactive watch faces with the latest Android We...
    - Develop a sweet spot for Marshmallow: Official And...
    - Barcode Detection in Google Play services
    - Face Detection in Google Play services
    - Google Play services 7.8 - Let's see what's Nearby...
    - Android Developer Story: Zabob Studio and Buff Stu...
    - Android Experiments: A celebration of creativity a...
    - Low-overhead rendering with Vulkan
- ▶ July (9)
- ▶ June (8)
- ▶ May (10)
- ▶ April (10)
- ▶ March (12)
- ▶ February (8)

## Face Detection in Google Play services

Posted by [Laurence Moroney](#), Developer Advocate

With the release of Google Play services 7.8, we announced the addition of new Mobile Vision APIs, which includes a new Face API that finds human faces in images and video better and faster than before. This API is also smarter at distinguishing faces at different orientations and with different facial features facial expressions.

### Face Detection

Face Detection is a leap forward from the previous Android FaceDetector.Face API. It's designed to better detect human faces in images and video for easier editing. It's smart enough to detect faces even at different orientations – so if your subject's head is turned sideways, it can detect it. Specific landmarks can also be detected on faces, such as the eyes, the nose, and the edges of the lips.

#### Important Note

This is **not** a face *recognition* API. Instead, the new API simply detects areas in the image or video that are human faces. It also infers from changes in the position frame to frame that faces in consecutive frames of video are the same face. If a face leaves the field of view, and re-enters, it isn't recognized as a previously detected face.

### Detecting a face

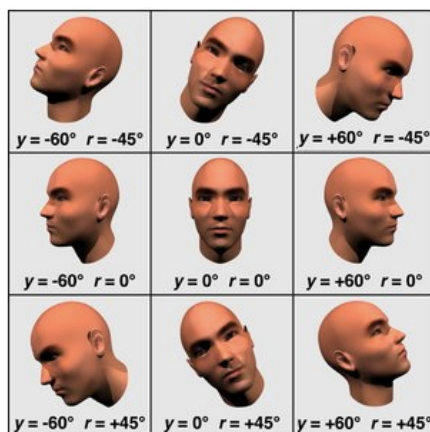
When the API detects a human face, it is returned as a Face object. The Face object provides the spatial data for the face so you can, for example, draw bounding rectangles around a face, or, if you use landmarks on the face, you can add features to the face in the correct place, such as giving a person a new hat.

- `getPosition()` - Returns the top left coordinates of the area where a face was detected
- `getWidth()` - Returns the width of the area where a face was detected
- `getHeight()` - Returns the height of the area where a face was detected
- `getId()` - Returns an ID that the system associated with a detected face

### Orientation

The Face API is smart enough to detect faces in multiple orientations. As the head is a solid object that is capable of moving and rotating around multiple axes, the view of a face in an image can vary wildly.

Here's an example of a human face, instantly recognizable to a human, despite being oriented in greatly different ways:



The API is capable of detecting this as a face, even in the circumstances where as much as half of the facial data is missing, and the face is oriented at an angle, such as in the corners of the above image.

► January (3)

- 2014 (73)
- 2013 (48)
- 2012 (41)
- 2011 (68)
- 2010 (72)
- 2009 (63)
- 2008 (40)
- 2007 (8)

## COMMUNITY

Android Developers



Follow

+1

Subscribe

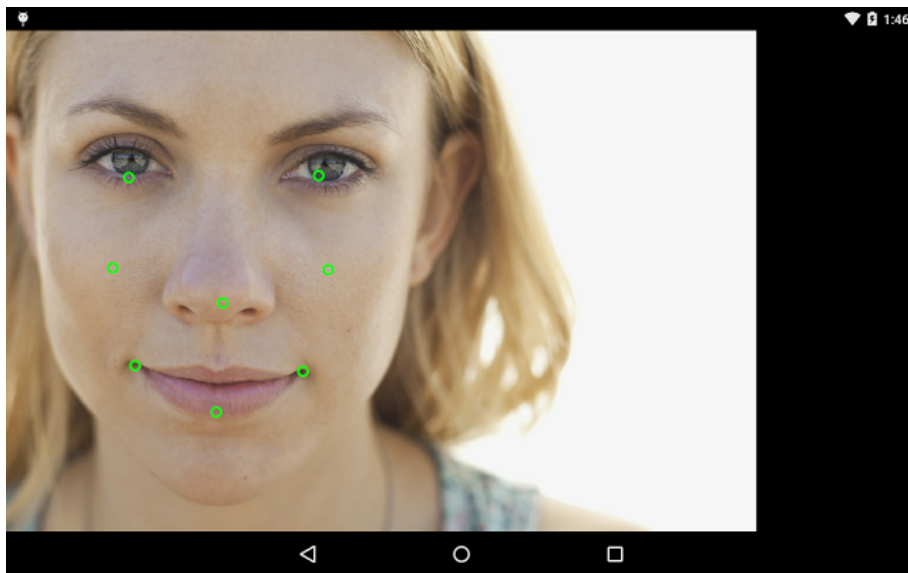
 20475 readers  
BY FEEDBURNER

Here are the method calls available to a face object:

- `getEulerY()` - Returns the rotation of the face around the vertical axis -- i.e. has the neck turned so that the face is looking left or right [The y degree in the above image]
- `getEulerZ()` - Returns the rotation of the face around the Z axis -- i.e. has the user tilted their neck to cock the head sideways [The r degree in the above image]

**Landmarks**

A landmark is a point of interest within a face. The API provides a `getLandmarks()` method which returns a `List`, where a `Landmark` object returns the coordinates of the landmark, where a landmark is one of the following: Bottom of mouth, left cheek, left ear, left ear tip, left eye, left mouth, base of nose, right cheek, right ear, right ear tip, right eye or right mouth.

**Activity**

In addition to detecting the landmark, the API offers the following function calls to allow you to smartly detect various facial states:

- `getIsLeftEyeOpenProbability()` - Returns a value between 0 and 1, giving probability that the left eye is open
- `getIsRighteyeOpenProbability()` - Same but for right eye
- `getIsSmilingProbability()` - Returns a value between 0 and 1 giving a probability that the face is smiling

Thus, for example, you could write an app that only takes a photo when all of the subjects in the image are smiling.

**Learn More**

It's easy to build applications that use facial detection using the Face API, and we've provided lots of great resources that will allow you to do so. Check them out here:

[Follow the Code Lab](#)
[Read the Documentation](#)
[Explore the sample](#)



Posted by [Reto Meier](#) at 3:45 PM

Labels: [Develop](#), [face detection](#), [Featured](#), [Google Play services](#), [mobile vision APIs](#)

---

## Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)