CS 528 Mobile and Ubiquitous Computing
Lecture 4b: Multimedia: Camera and ML Kit Overview

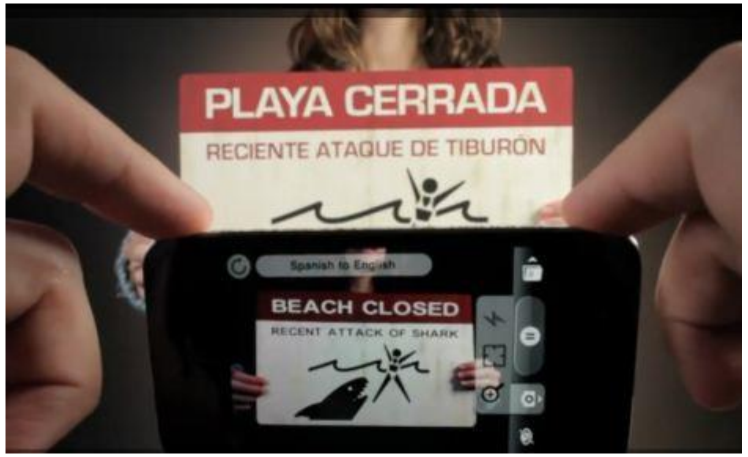
Emmanuel Agu

The Mobile Camera

Interesting application

Word Lens Feature of Google Translate

- Word Lens: translates text/signs in foreign Language in real time
- Example use case: tourist can understand signs, restaurant menus
- Uses Optical Character Recognition technology
- Google bought company in 2014, now part of Google Translate



[Original Word Lens App]

[Word Lens as part of Google Translate]

Camera: Taking Pictures

3 Generations of Android Camera

https://developer.android.com/training/camera/choose-camera-library

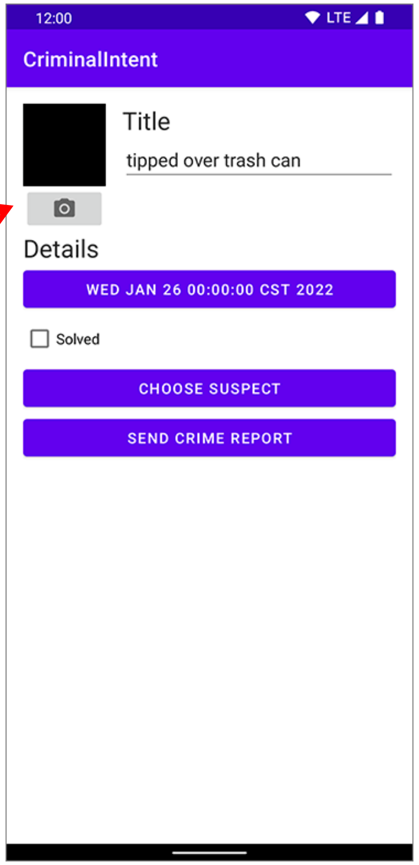
- **Camera:** Original Android Camera class, now deprecated
- **Camera2:**
  - Newer, works on Android 5.0 (API 21) or higher
  - Somewhat complex, requires programming device-specific configurations
- **CameraX:** (recommended)
  - Jetpack Camera class
  - Supports most Android devices, requires Android 5.0 and higher
  - High-level API, does not require writing device-specific code
- New apps currently use Camera2 or CameraX

Taking Pictures with Intents

Ref: Ch 17 Android Nerd Ranch 5th edition

- Would like to take picture of "Crime" (e.g. plate in sink) to document it.
- Use implicit intent to start Camera app from our CrimeIntent app
- **Recall:** Implicit intent used to call component in different activity

Click here to take picture


12:00 LTE

CriminalIntent

Title

tipped over trash can

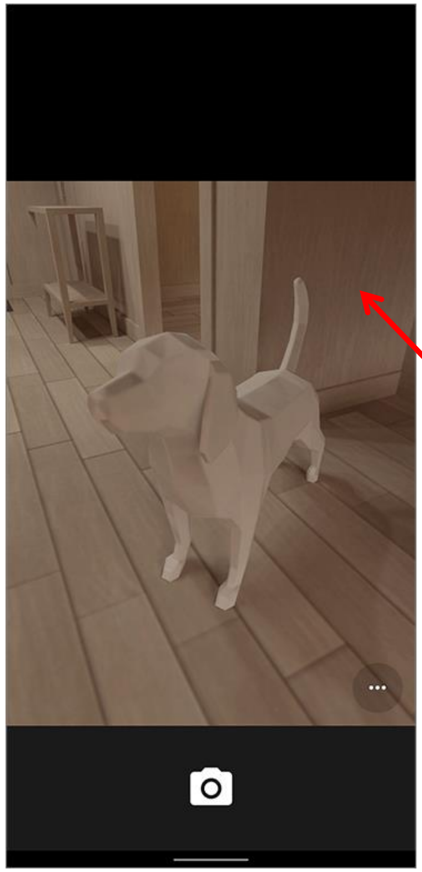
Details

WED JAN 26 00:00:00 CST 2022

Solved

CHOOSE SUSPECT

SEND CRIME REPORT

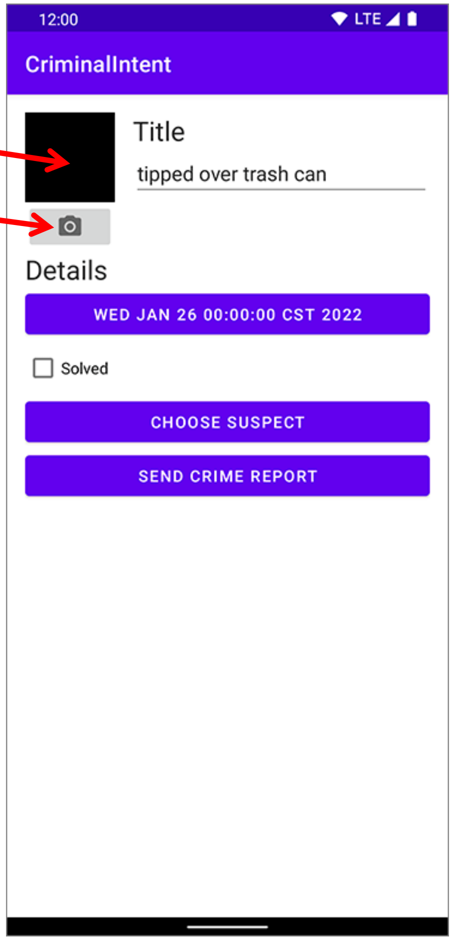


Launches Camera app

Create Placeholder for Picture

Ref: Ch 17 Android Nerd Ranch 5th edition

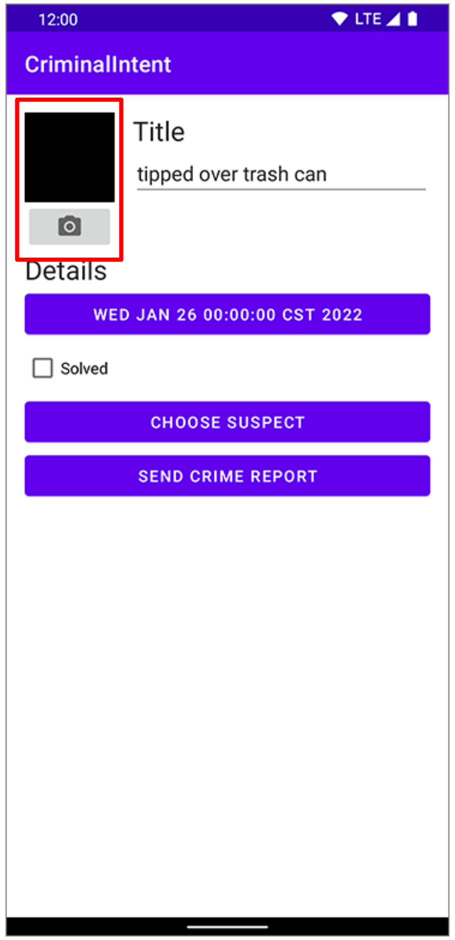
- Modify layout to include
  - ImageView for picture
  - Button to take picture

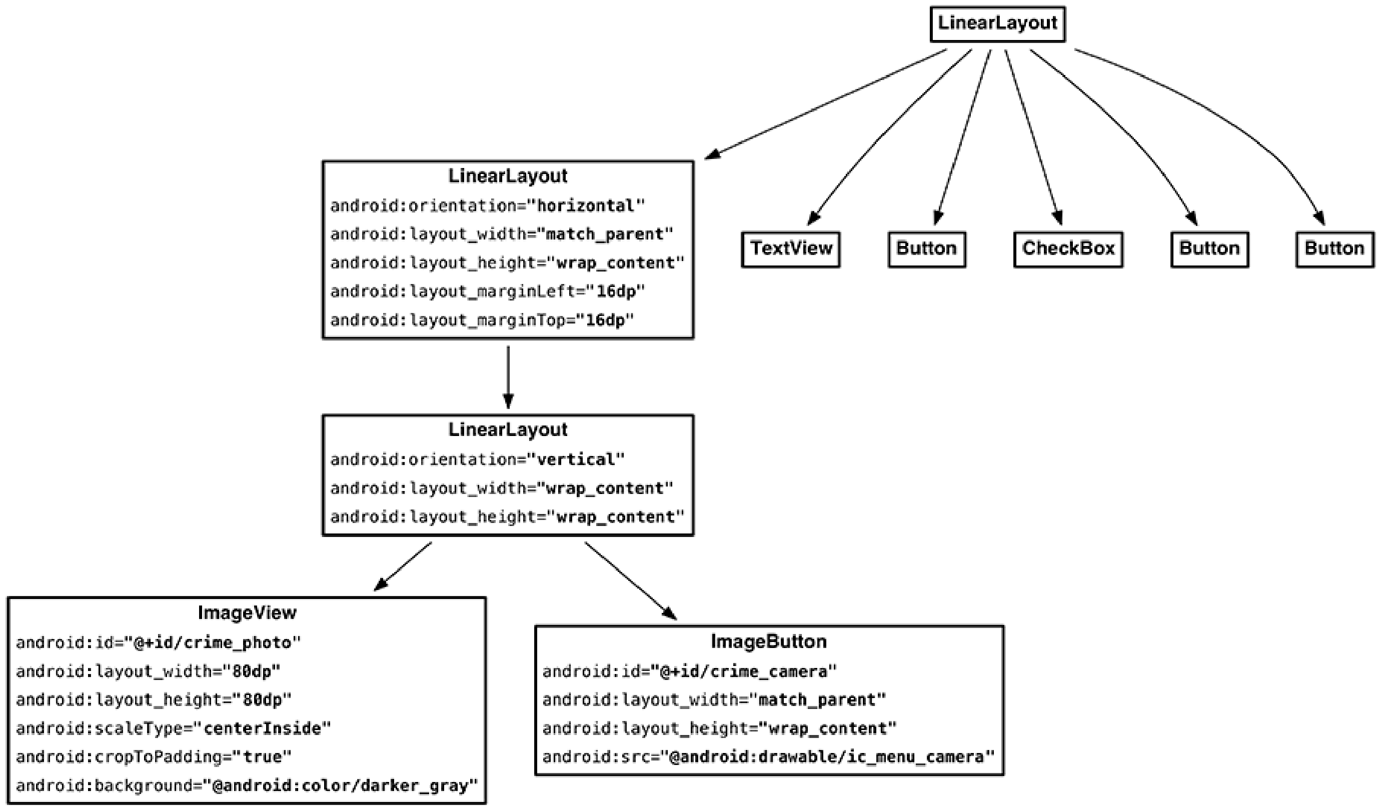


Create Layout for Thumbnail and Button

Ref: Ch 17 Android Nerd Ranch 5th edition

- First, build out left side





```
LinearLayout
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginTop="16dp"
```

```
LinearLayout
android:orientation="vertical"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

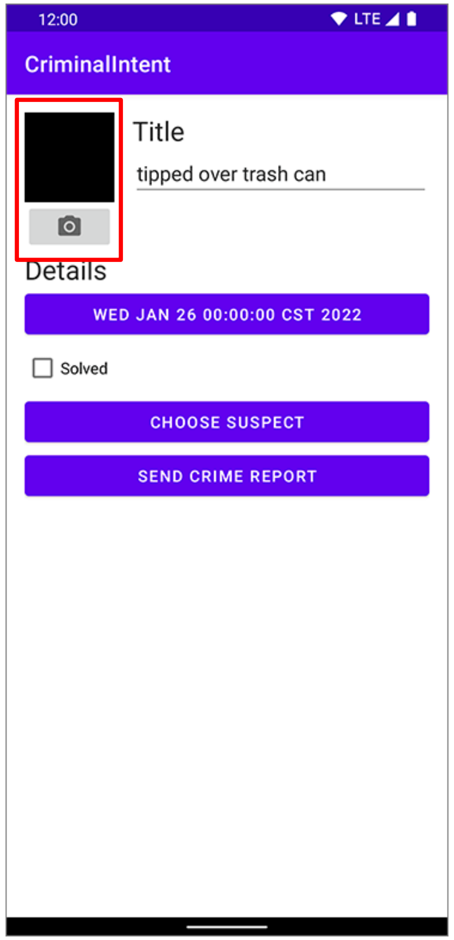
```
ImageView
android:id="@+id/crime_photo"
android:layout_width="80dp"
android:layout_height="80dp"
android:scaleType="centerInside"
android:cropToPadding="true"
android:background="@android:color/darker_gray"
```

```
ImageButton
android:id="@+id/crime_camera"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:src="@android:drawable/ic_menu_camera"
```

Create Layout for Thumbnail and Button

Ref: Ch 17 Android Nerd Ranch 5th edition

- First, build out left side

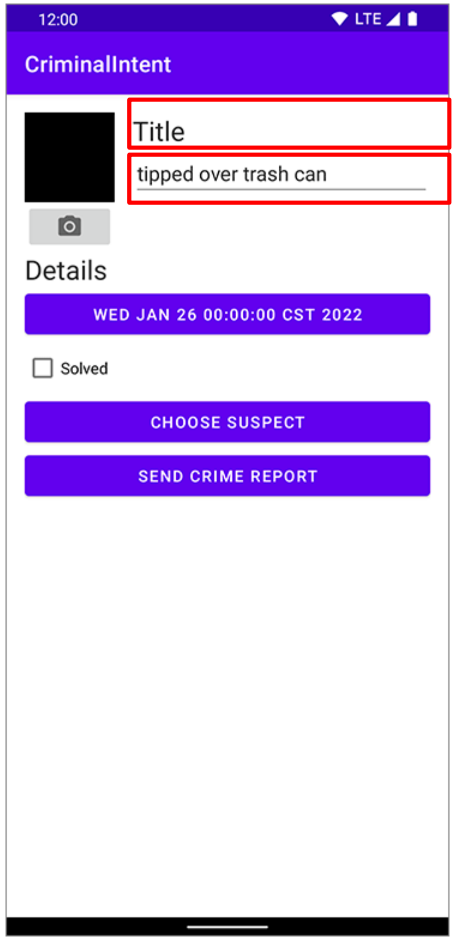


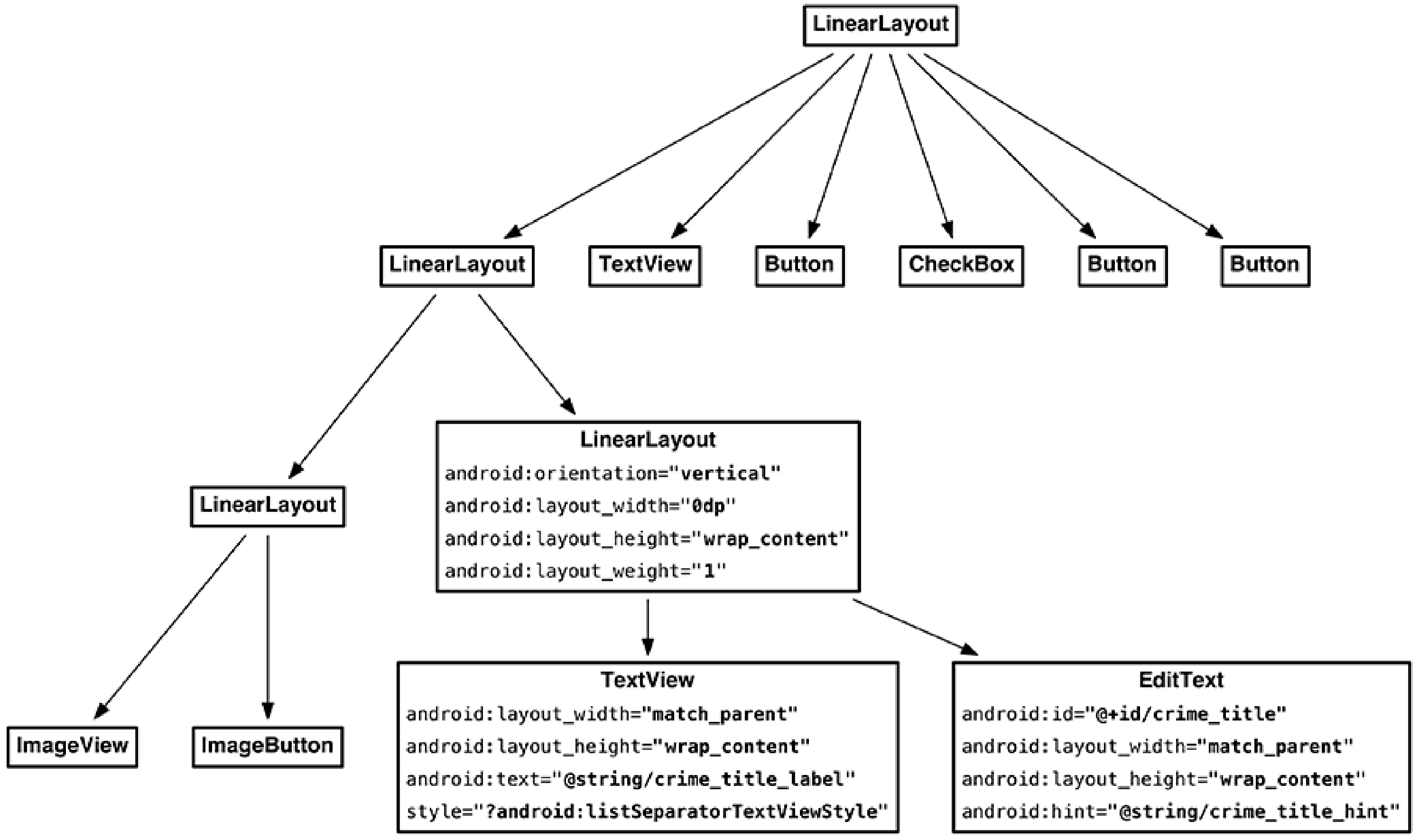
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              ... >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginEnd="16dp">

            <ImageView
                android:id="@+id/crime_photo"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:scaleType="centerInside"
                android:cropToPadding="true"
                android:background="@color/black"/>

            <ImageButton
                android:id="@+id/crime_camera"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:src="@drawable/ic_camera"/>
        </LinearLayout>
    </LinearLayout>
```

Create Title and Crime Entry EditText

Ref: Ch 17 Android Nerd Ranch 5th edition

- Build out right side





```
LinearLayout
android:orientation="vertical"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="1"
```

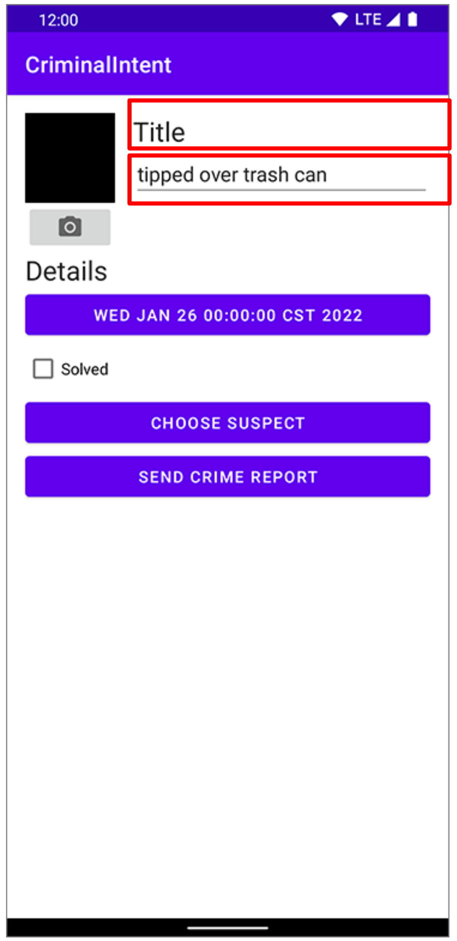
```
TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/crime_title_label"
style="?android:listSeparatorTextViewStyle"
```

```
EditText
android:id="@+id/crime_title"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/crime_title_hint"
```

Create Title and Crime Entry EditText

Ref: Ch 17 Android Nerd Ranch 5th edition

- Build out right side

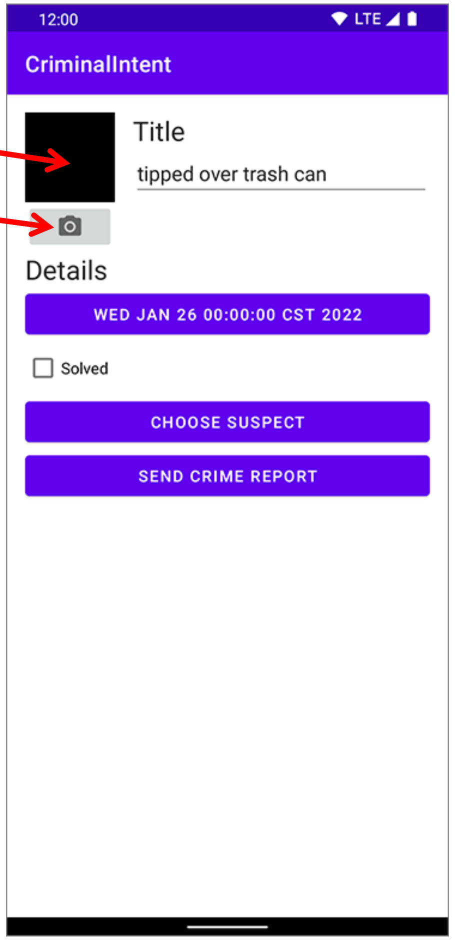


```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?attr/textAppearanceHeadline5"
        android:text="@string/crime_title_label" />

    <EditText
        android:id="@+id/crime_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:importantForAutofill="no"
        android:hint="@string/crime_title_hint"
        android:inputType="text" />
</LinearLayout>
```

Compile and Run CriminalIntent at this point

Ref: Ch 17 Android Nerd Ranch 5th edition

- Modified layout to include
  - ImageView for picture
  - Button to take picture
  - Crime title



File Storage

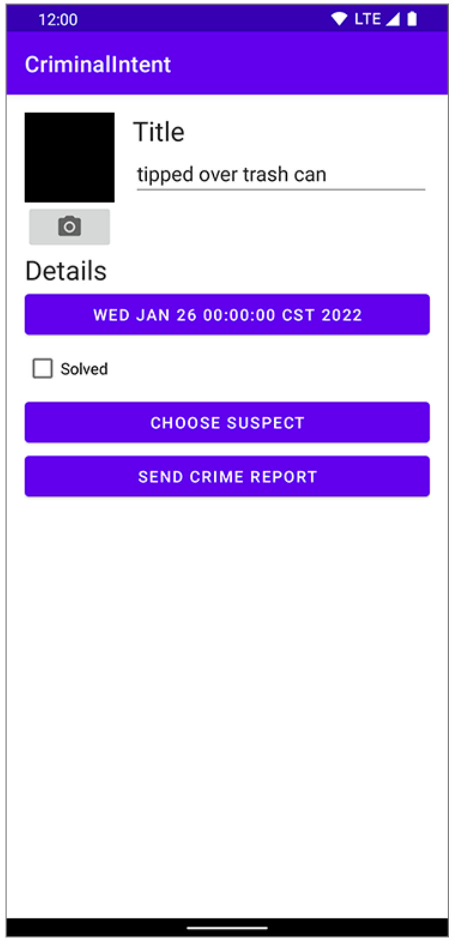
Ref: Ch 17 Android Nerd Ranch 5th edition

- Need place on filesystem to store images **CriminalIntent** receives from Camera app
- Can use Android **FileProvider** class to share files between apps
  - E.g. between camera app and **CriminalIntent**
  - **FileProvider** extends **ContentProvider** class
- First declare **FileProvider** as a **ContentProvider** in Android manifest

```
<activity android:name=".MainActivity">
    ...
</activity>
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="com.bignerdranch.android.criminalintent.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
</provider>
...
```

Location file will be saved to → android:authorities

Ensures data NOT visible to other apps → android:exported



Taking Picture Using Camera Intent

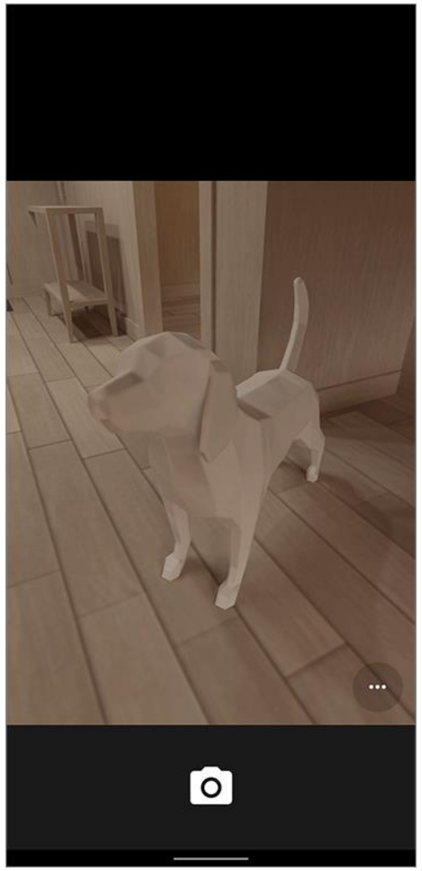
Ref: Ch 17 Android Nerd Ranch 5th edition

- 2 types of Android file locations:
  - **Private/internal:** only visible to this app (we will use this)
  - **Public/external:** shared by multiple apps

- Taking picture process:
  - Launch external camera app
  - User takes a photo
  - Update crime with path to the new file

- Take a picture using **ActivityResultContracts.TakePicture( )**
  - Returns boolean indicating whether image was saved to file

```
private val takePhoto = registerForActivityResult(
    ActivityResultContracts.TakePicture()
) { didTakePhoto: Boolean ->
    // Handle the result
}
```



Taking Picture Using Camera Intent

Ref: Ch 17 Android Nerd Ranch 5th edition

- On Camera button click, launch the Camera app

Name of .JPG file to save image

File to store full-sized image

```
crimeCamera.setOnClickListener {
    val photoName = "IMG_${Date()}.JPG"
    val photoFile = File(requireContext().applicationContext.filesDir,
                         photoName)
    val photoUri = FileProvider.getUriForFile(
        requireContext(),
        "com.bignerdranch.android.criminalintent.fileprovider",
        photoFile
    )

    takePhoto.launch(photoUri)
}
```

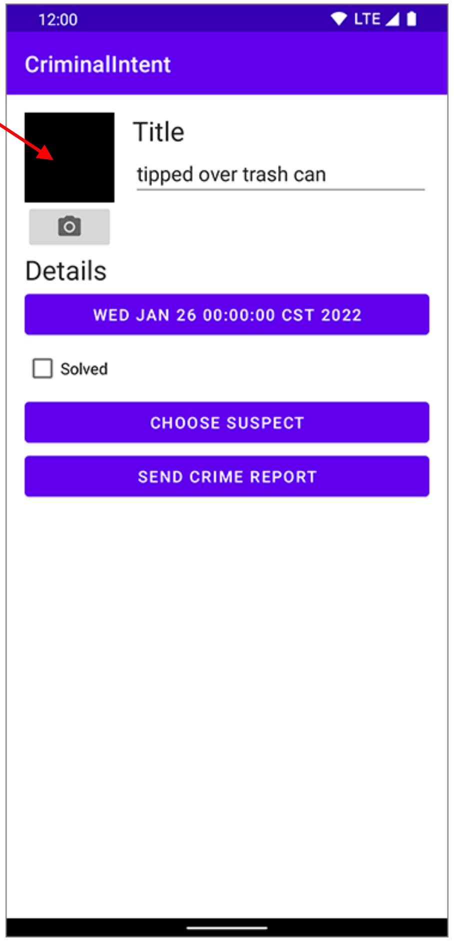
URI location to store captured image (E.g. file//xyz)

Taking Picture Using Camera Intent

Ref: Ch 17 Android Nerd Ranch 5th edition

- Handle the result
  - If photo taken and photo filename is not null, update crime phone

```
class CrimeDetailFragment : Fragment() {
    ...
    private val takePhoto = registerForActivityResult(
        ActivityResultContracts.TakePicture()
    ) { didTakePhoto ->
        ~~// Handle the result~~
        if (didTakePhoto && photoName != null) {
            crimeDetailViewModel.updateCrime { oldCrime ->
                oldCrime.copy(photoFileName = photoName)
            }
        }
    }

    private var photoName: String? = null
```



Taking Picture Using Camera Intent

Ref: Ch 17 Android Nerd Ranch 5th edition

- On some phones, there may be no camera app that can take picture
- Disable Camera button if no app on user's phone can take a picture

```
    val captureImageIntent = takePhoto.contract.createIntent(
        requireContext(),
        null
    )
    crimeCamera.isEnabled = canResolveIntent(captureImageIntent)
}
```

Taking Picture Using Camera Intent

Ref: Ch 17 Android Nerd Ranch 5th edition

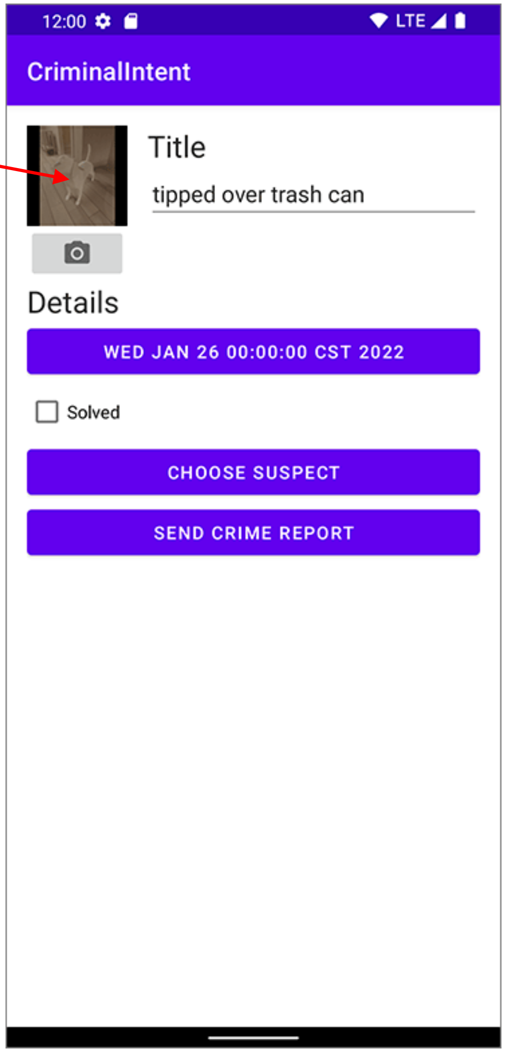
- Allow CriminalIntent to query for/get list of camera apps
  - Add query Intent to Android Manifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bignerdranch.android.criminalintent">

    <application ...>
        ...
    </application>
    <queries>
        <intent>
            <action android:name="android.intent.action.PICK" />
            <data android:mimeType="vnd.android.cursor.dir/contact" />
        </intent>
        <intent>
            <action android:name="android.media.action.IMAGE_CAPTURE" />
        </intent>
    </queries>
</manifest>
```

Taking Picture Using Camera Intent

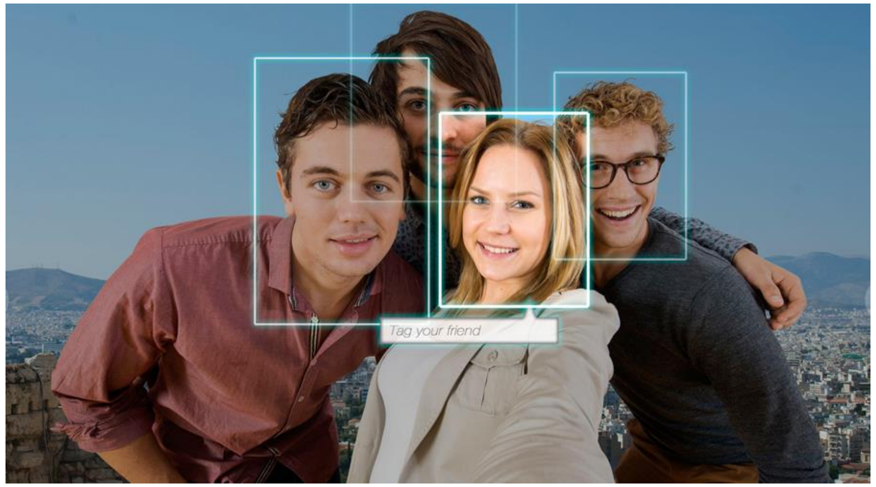
Ref: Ch 17 Android Nerd Ranch 5th edition

- Now have full-sized picture
- But need thumbnail to insert into Crime record
- Solution:
  - Scale full-size image to thumbnail
  - Insert thumbnail into appropriate location on Crime
  - See ANR (5th edition), Ch 17



Face Recognition

Face Recognition



- Answers the question:

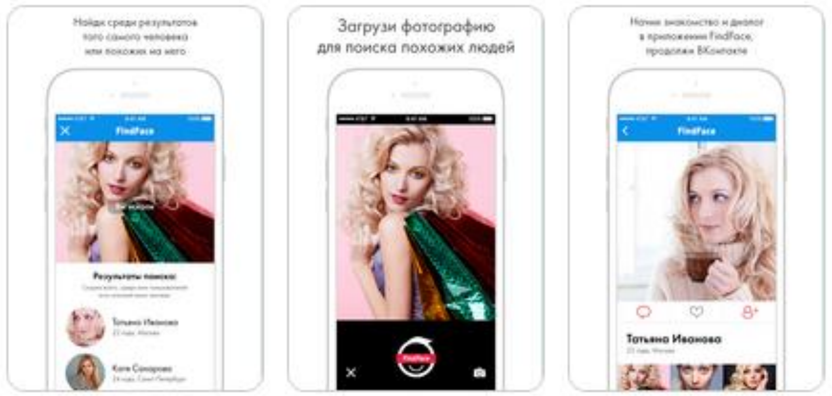
Who is this person in this picture?

Example answer: John Smith

- Compares unknown face to database of faces with known identity
- Neural networks/deep learning now makes comparison faster

FindFace App: Find out who a strenger is

- See stranger you like? Take a picture
- App searches 1 billion pictures using neural networks < 1 second
- Finds person’s picture, identity, link on VK (Russian Facebook)
  - You can send friend Request
- ~ 70% accurate!
- Can also upload picture of celebrity you like
- Finds 10 strangers on Facebook who look similar, can send friend request


Найди среди результатов того самого человека или похожих на него
Загрузи фотографию для поиска похожих людей
Начни знакомство и диалог в приложении FindFace, продолжи ВКонтакте
FindFace
Результаты поиска
Татьяна Иванова

Google ML Kit

ML Kit

Ref: https://developers.google.com/ml-kit/guides

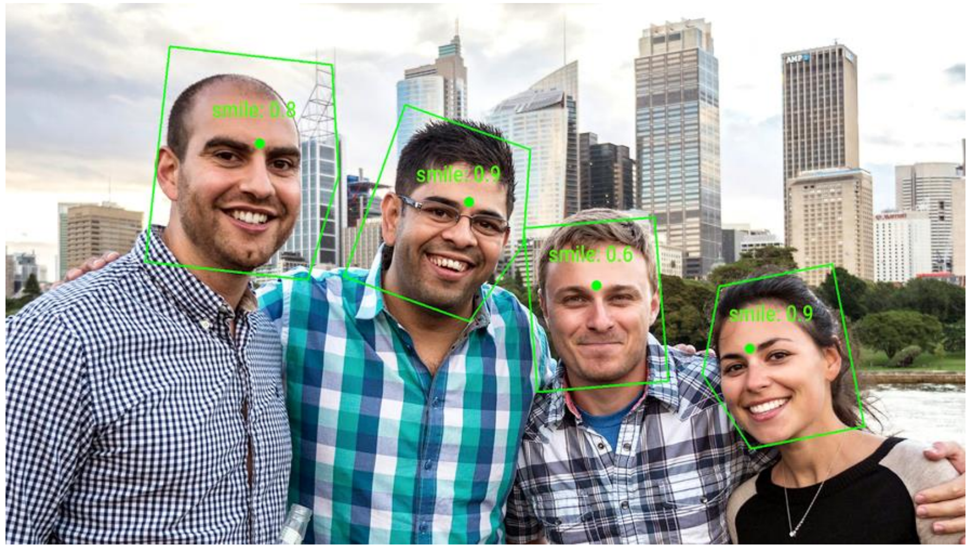
- ML kit is mobile SDK for on-device machine learning
- Mostly computer vision and Natural Language Processing (NLP) APIs including
  - Text recognition
  - Face detection
  - Barcode scanning
  - Image labeling
  - Object detection and tracking
  - Pose detection
  - Selfie segmentation
  - Smart Reply
  - Text Translation
  - Language identification

Face Detection

ML Kit Face Detection

https://developers.google.com/ml-kit/guides

- ML kit does face **detection** but NOT **recognition**
- **Face Detection:** Are there [any] faces in this picture?
- **How?** Locate face in photos and video and
  - **Facial landmarks:** Eyes, nose and mouth
  - **State of facial features:** Eyes open? Smiling?
  - **Contours** of detected faces



ML Kit Face Detection: Key features

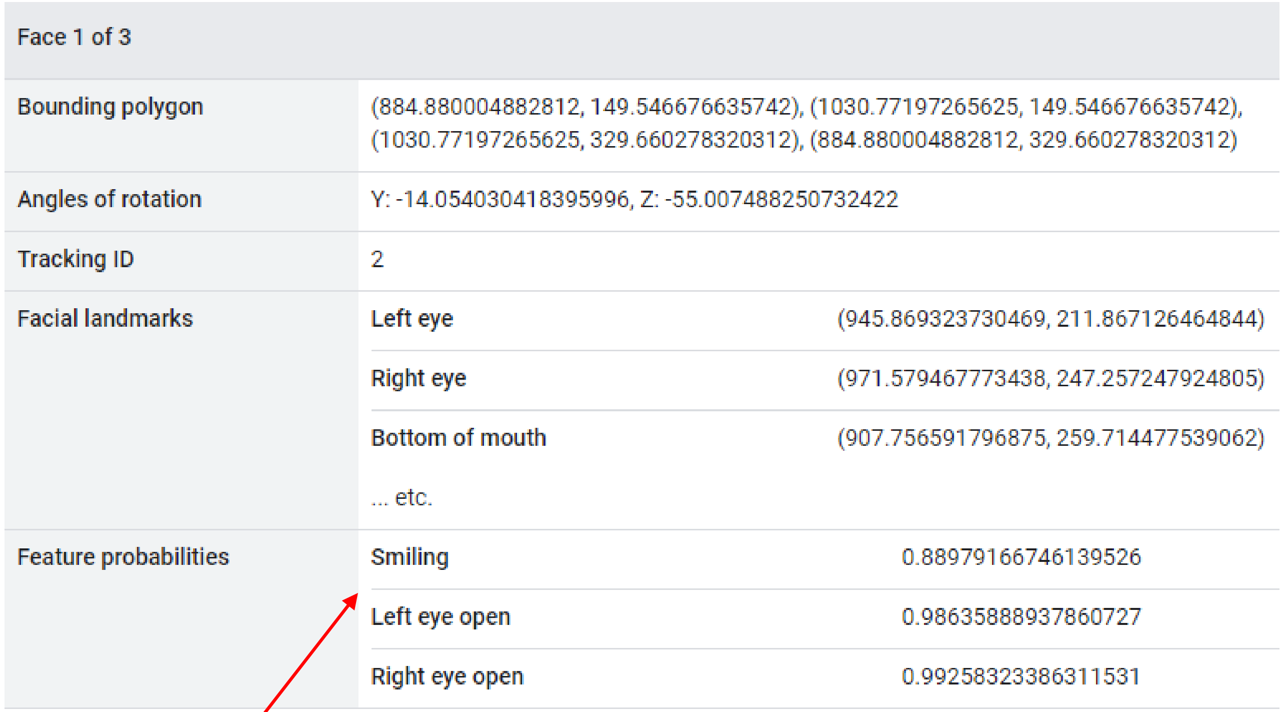
https://developers.google.com/ml-kit/guides

- **Recognize and locate facial features:** Coordinates of eyes, ears, cheeks, nose, and mouth of every face detected
- **Get contours (shape) of facial features:** Contours of detected faces and eyes, eyebrows, lips and nose
- **Recognize facial expressions:** smiling or eyes closed
- **Track faces across video frames:**
  - E.g. if same face appears in multiple frames
  - Enables manipulation of specific person’s image in video stream
- **Real-time processing of video frames:** to detect faces, on device

ML Kit Face Detection

Ref: https://developers.google.com/ml-kit/vision/face-detection

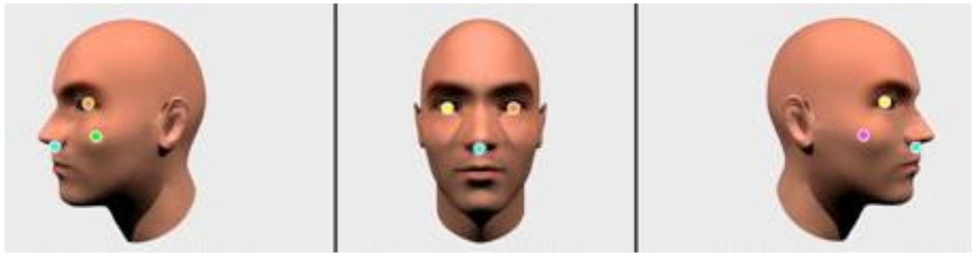
- For each face detected, following data is returned:

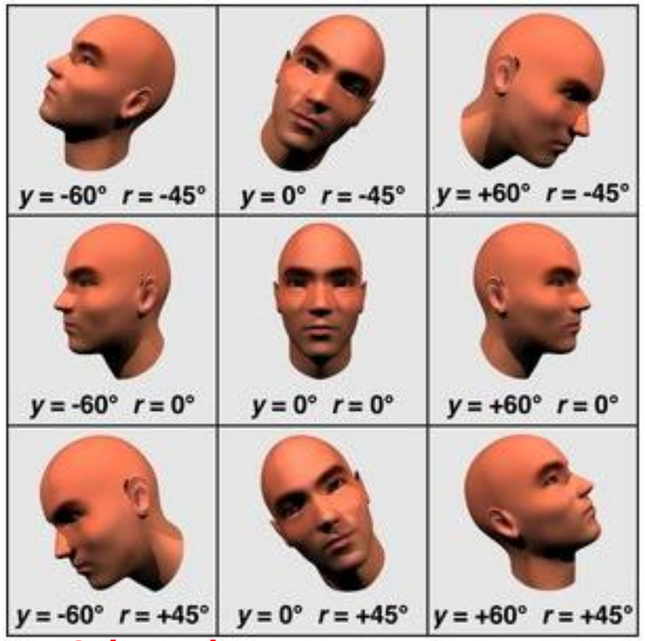


Face 1 of 3		
Bounding polygon	(884.880004882812, 149.546676635742), (1030.77197265625, 149.546676635742), (1030.77197265625, 329.660278320312), (884.880004882812, 329.660278320312)	
Angles of rotation	Y: -14.054030418395996, Z: -55.007488250732422	
Tracking ID	2	
Facial landmarks	Left eye	(945.869323730469, 211.867126464844)
	Right eye	(971.579467773438, 247.257247924805)
	Bottom of mouth	(907.756591796875, 259.714477539062)
	... etc.	
Feature probabilities	Smiling	0.88979166746139526
	Left eye open	0.98635888937860727
	Right eye open	0.99258323386311531

- Returns confidence that a facial characteristic is present
  - Confidence > 0.7 means facial characteristic is present
  - E.g. > 0.7 confidence means it's likely person is smiling

Landmarks



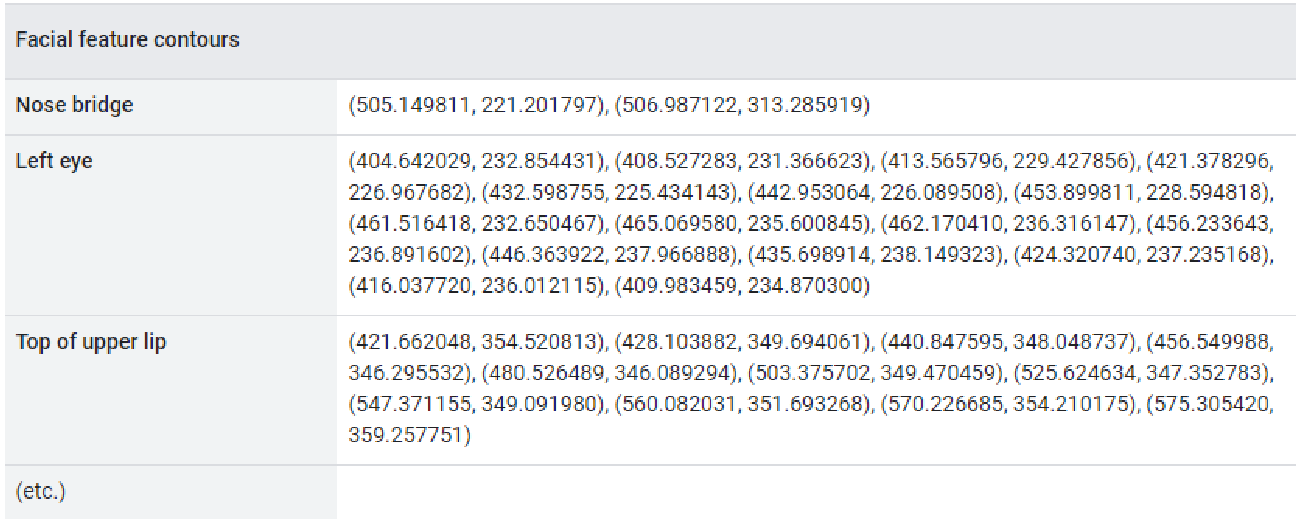


Orientation

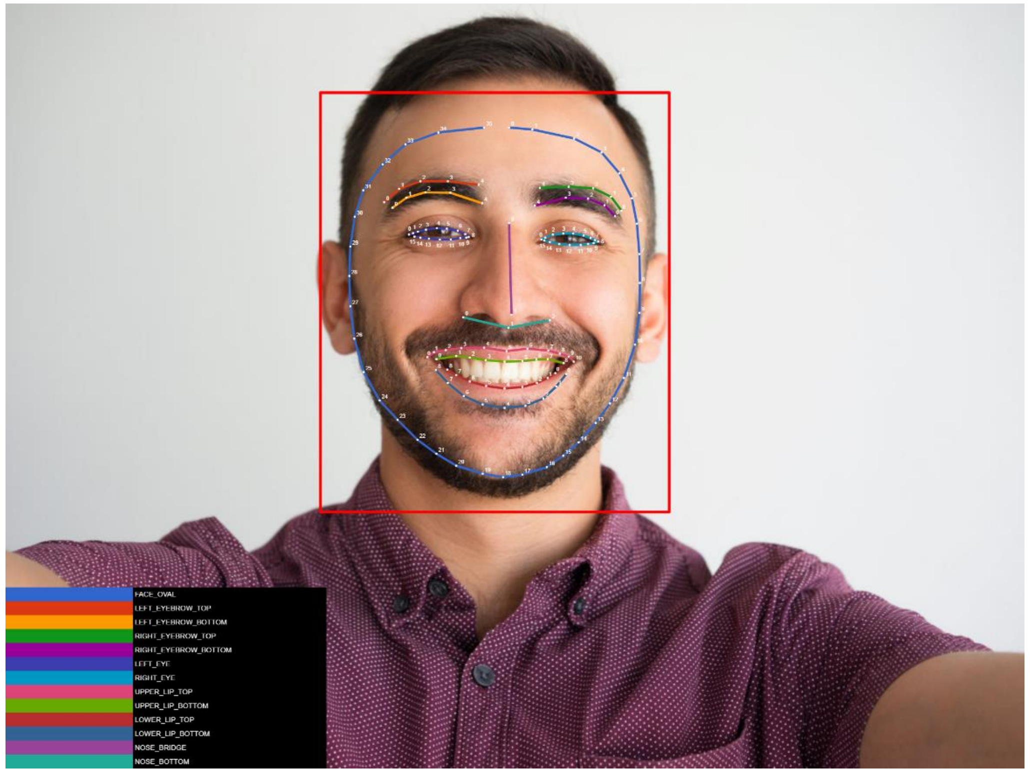
Face Detection

Ref: https://developers.google.com/ml-kit/vision/face-detection

- For each face detected, when face contour detection is enabled
  - Get list of points defining shape of feature (or contours)



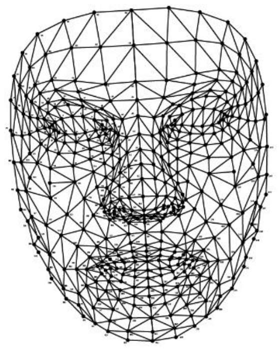
Facial feature contours	
Nose bridge	(505.149811, 221.201797), (506.987122, 313.285919)
Left eye	(404.642029, 232.854431), (408.527283, 231.366623), (413.565796, 229.427856), (421.378296, 226.967682), (432.598755, 225.434143), (442.953064, 226.089508), (453.899811, 228.594818), (461.516418, 232.650467), (465.069580, 235.600845), (462.170410, 236.316147), (456.233643, 236.891602), (446.363922, 237.966888), (435.698914, 238.149323), (424.320740, 237.235168), (416.037720, 236.012115), (409.983459, 234.870300)
Top of upper lip	(421.662048, 354.520813), (428.103882, 349.694061), (440.847595, 348.048737), (456.549988, 346.295532), (480.526489, 346.089294), (503.375702, 349.470459), (525.624634, 347.352783), (547.371155, 349.091980), (560.082031, 351.693268), (570.226685, 354.210175), (575.305420, 359.257751)
(etc.)	



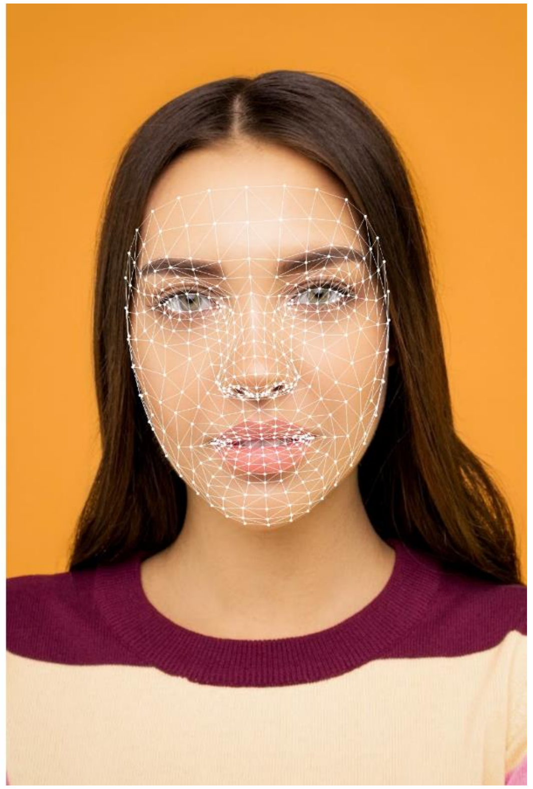
Face Mesh Detection

Ref: https://developers.google.com/ml-kit/vision/face-mesh-detection

- Generates high accuracy mesh of 468 3D points for selfie-like images in real time
- **Recognize and locate faces**
  - Get bounding box (rectangular area) of detected faces
- **Get face mesh information**
  - 468 3D points and triangle info for each detected face.
- **Real-time processing of video frames on device**

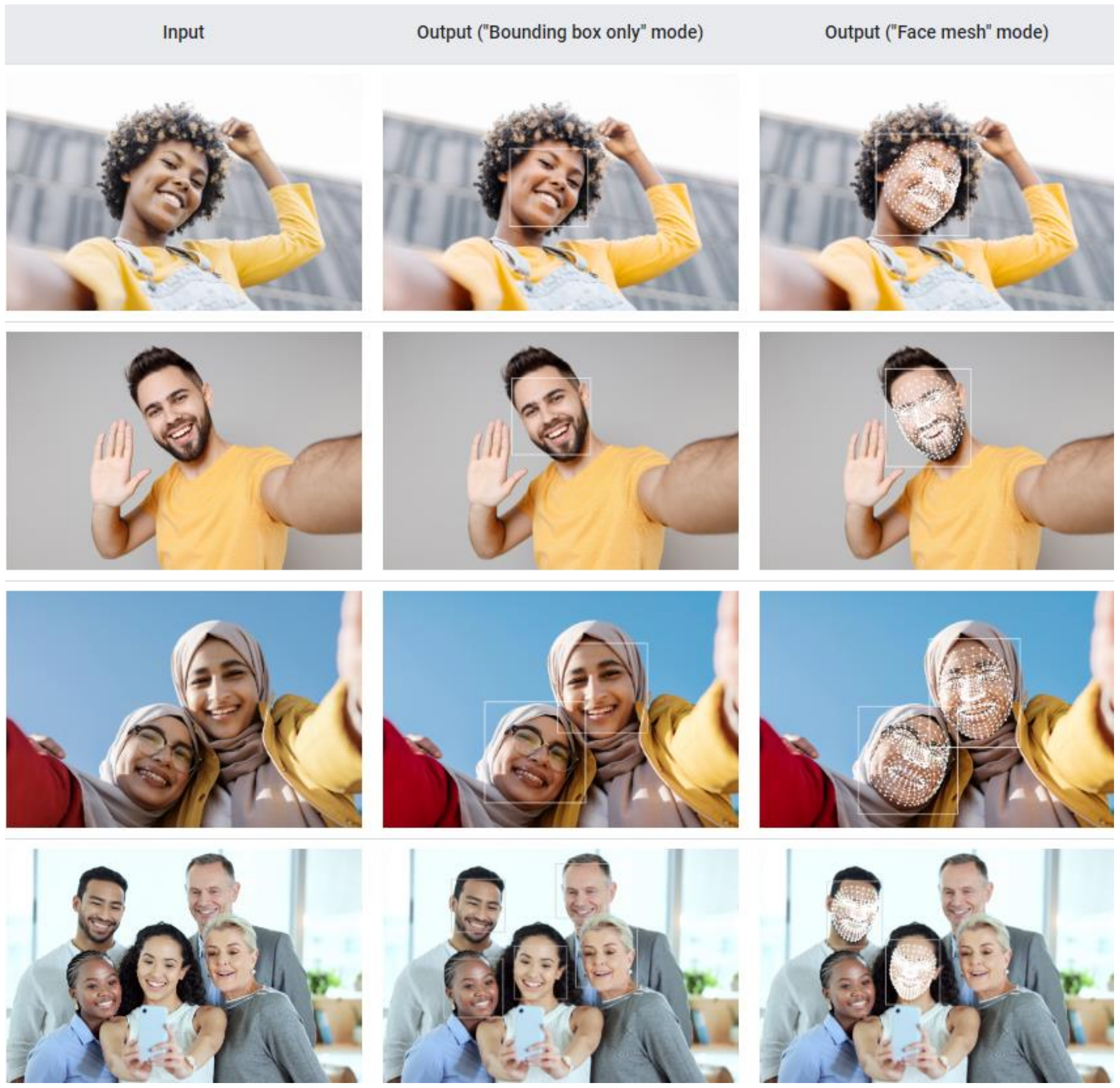


468 points



Face Mesh Detection: Example Results

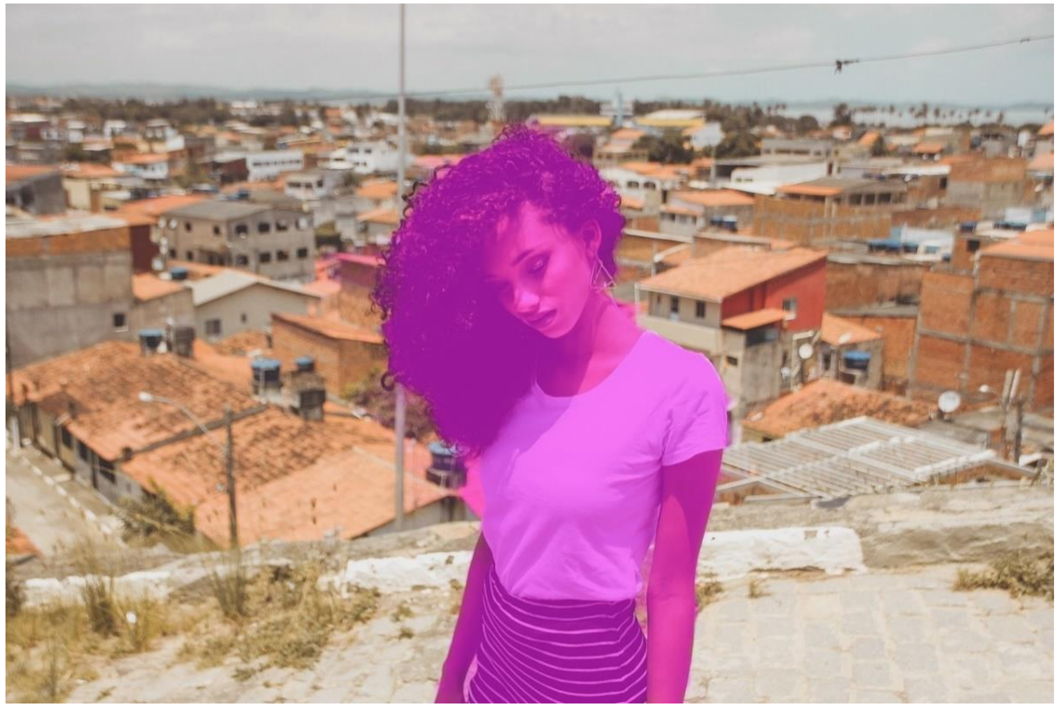
Ref: https://developers.google.com/ml-kit/vision/face-mesh-detection



Selfie Segmentation

Ref: https://developers.google.com/ml-kit/vision/selfie-segmentation

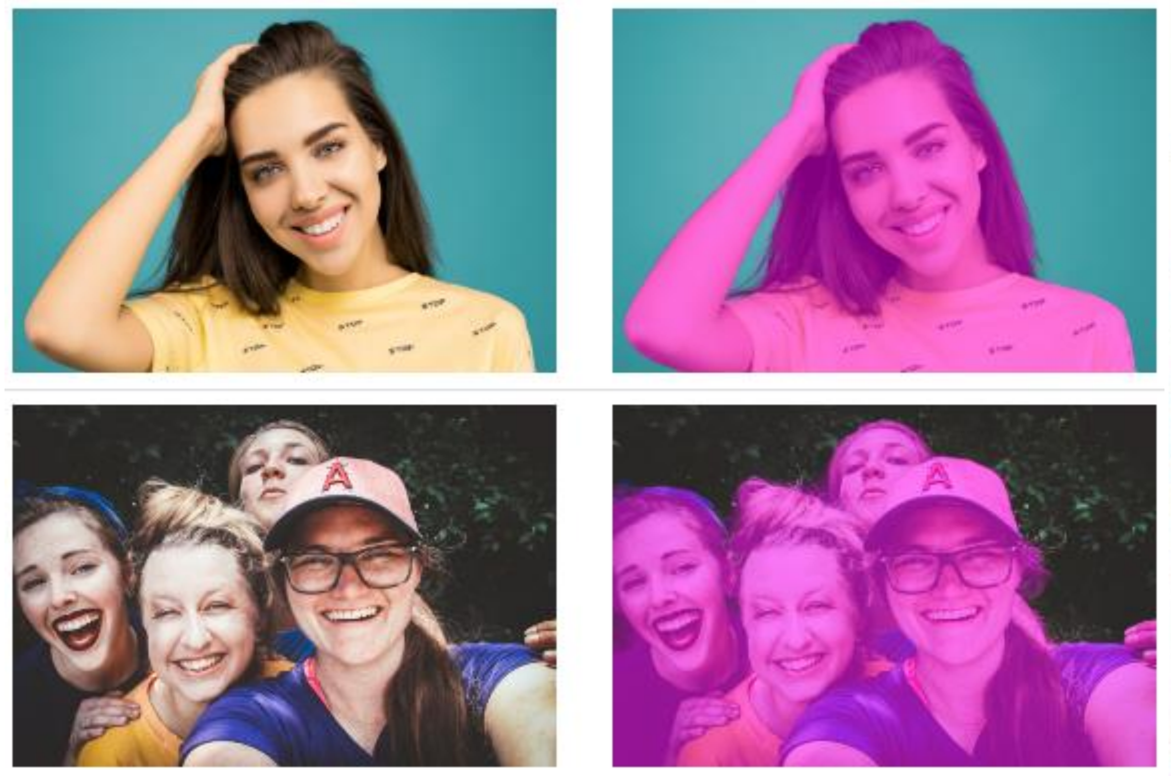
- Generates output mask from input selfie image
- Each pixel of mask assigned floating point number between 0 and 1
  - Closer to 1: Higher confidence pixel represents a person
- Works on static images or videos

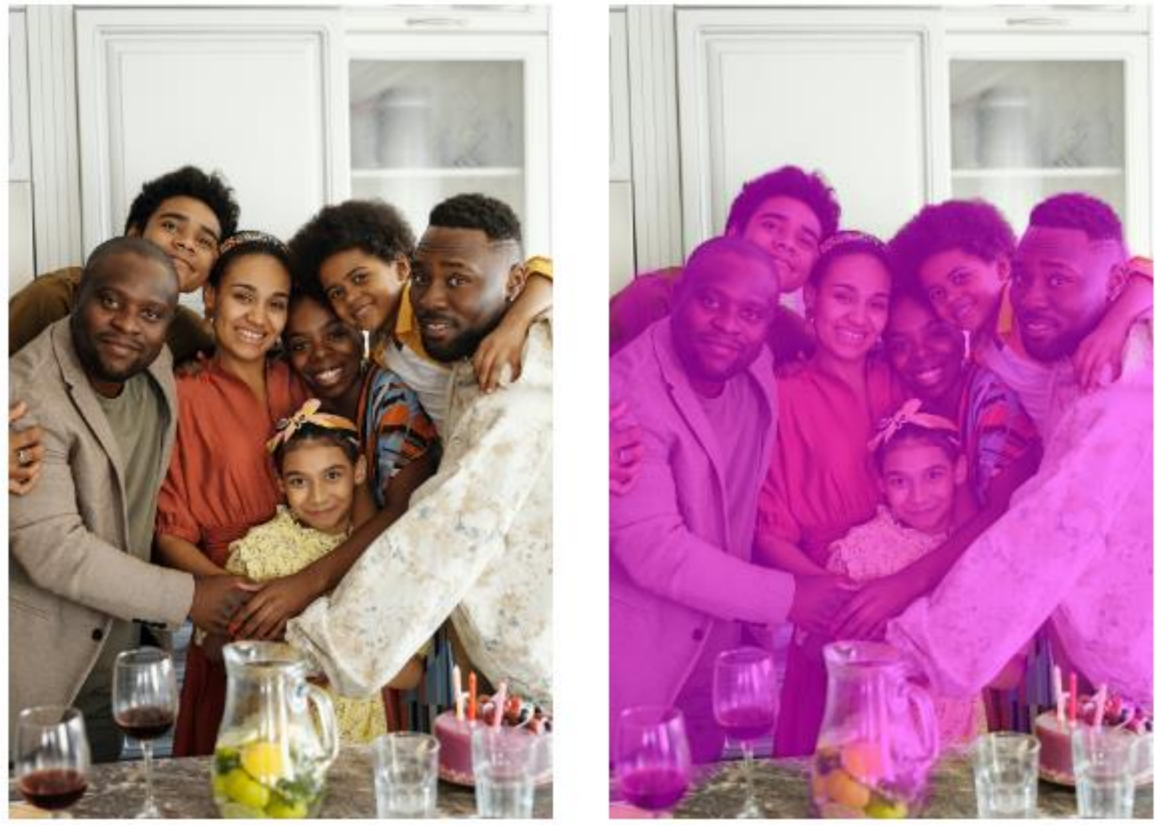


Selfie Segmentation

Ref: https://developers.google.com/ml-kit/vision/selfie-segmentation

- Examples

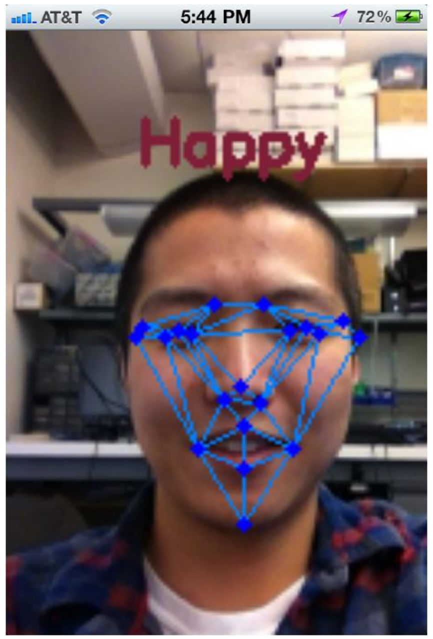




Face Interpretation

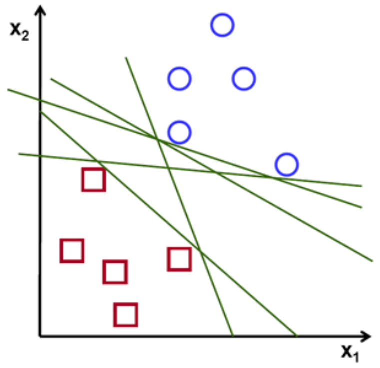
Visage Face Interpretation Engine

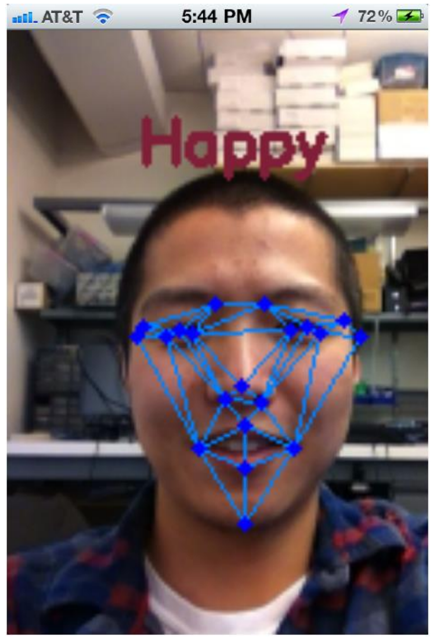
- Idea proposed in paper:
  - Yang, Xiaochao, et al. "Visage: A face interpretation engine for smartphone applications." *Mobile Computing, Applications, and Services Conference*. Springer Berlin Heidelberg, 2012. 149-168.
- Real-time face interpretation engine for smart phones
  - Tracking user's 3D head orientation + facial expression
- Facial expression?
  - angry, disgust, fear, happy, neutral, sad, surprise
  - Intuition: shapes of triangles on face correspond to facial expression
  - Usage? Can be used in Mood Profiler app



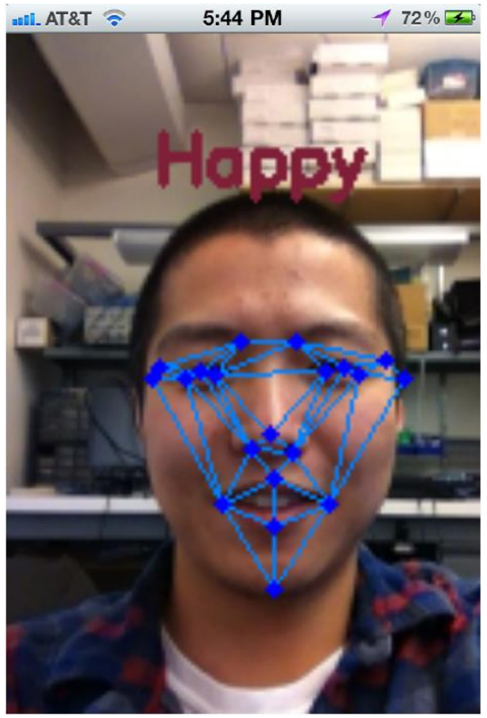
Facial Expression Inference

- Active appearance model
  - Describes 2D image as triangular mesh of landmark points
- 7 expression classes: angry, disgust, fear, happy, neutral, sad, surprise
- Extract triangle shape, texture features
- Classify features using Machine learning





Classification Accuracy



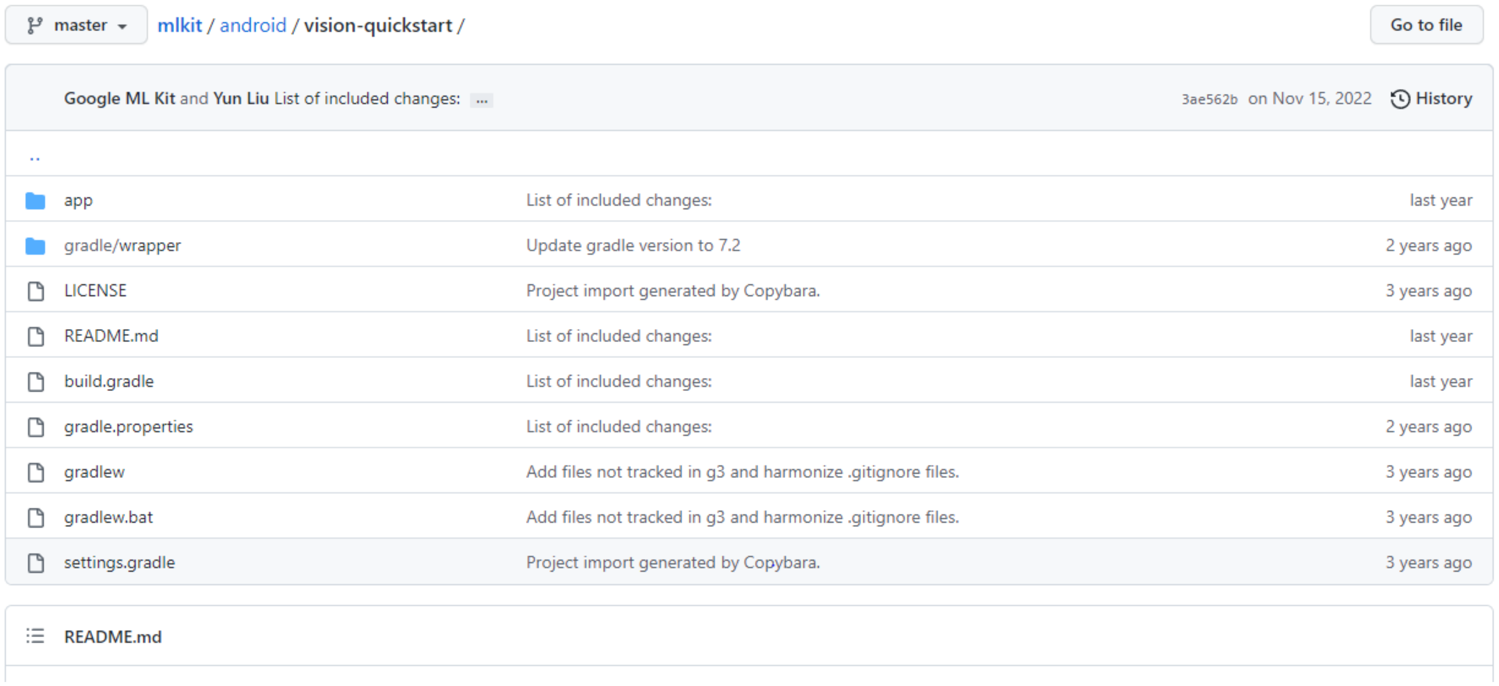
Expressions	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Accuracy(%)	82.16	79.68	83.57	90.30	89.93	73.24	87.52

Face Detection Using Google's Machine Learning (ML) Kit

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- ML kit can be used to detect faces in images and video
- Requires **Android API Level 19 or above**
- Download, study ML kit vision quickstart demos from github
  - https://github.com/googlesamples/mlkit/tree/master/android/vision-quickstart



Google ML Kit and Yun Liu List of included changes: ... 3ae562b on Nov 15, 2022

..

app	List of included changes:	last year
gradle/wrapper	Update gradle version to 7.2	2 years ago
LICENSE	Project import generated by Copybara.	3 years ago
README.md	List of included changes:	last year
build.gradle	List of included changes:	last year
gradle.properties	List of included changes:	2 years ago
gradlew	Add files not tracked in g3 and harmonize .gitignore files.	3 years ago
gradlew.bat	Add files not tracked in g3 and harmonize .gitignore files.	3 years ago
settings.gradle	Project import generated by Copybara.	3 years ago

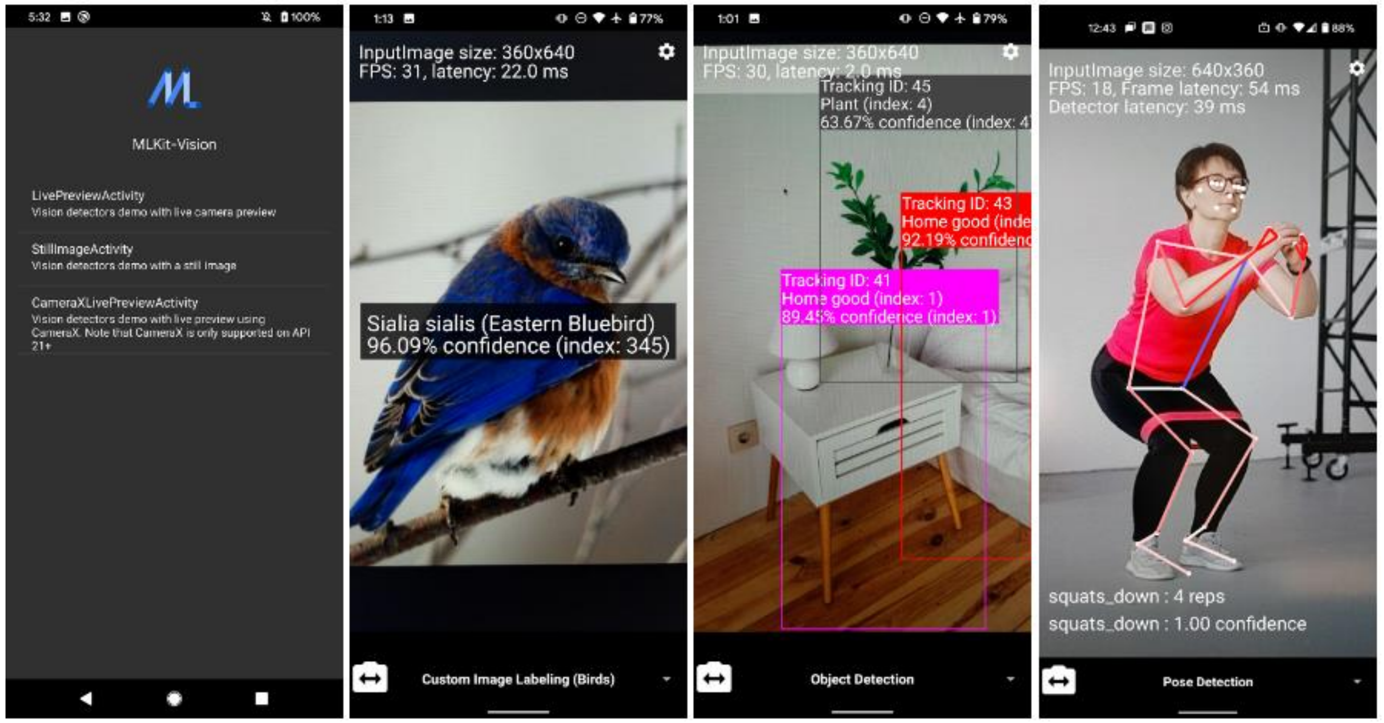
README.md

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- ML kit Quickstart app contains cool working demos
- Github site provides:
  - Instructions on running demo app
  - Documentation for ML Kit code
  - API reference, and
  - Link to stack overflow site:
    - You can review questions already asked, ask new questions.

- Object Detection - Detect, track, and classify objects in real time and static images
- Face Detection - Detect faces in real time and static images
- Face Mesh Detection - Detect face mesh in real time and static images
- Text Recognition - Recognize text in real time and static images
- Barcode Scanning - Scan barcodes in real time and static images
- Image Labeling - Label images in real time and static images
- Custom Image Labeling - Birds - Label images of birds with a custom TensorFlow Lite model.
- Pose Detection - Detect the position of the human body in real time.
- Selfie Segmentation - Segment people from the background in real time.



MLKit-Vision

LivePreviewActivity
Vision detectors demo with live camera preview

StillImageActivity
Vision detectors demo with a still image

CameraXLivePreviewActivity
Vision detectors demo with live preview using CameraX. Note that CameraX is only supported on API 21+

InputImage size: 360x640
FPS: 31, latency: 22.0 ms

Sialia sialis (Eastern Bluebird)
96.09% confidence (index: 345)

Custom Image Labeling (Birds)

InputImage size: 360x640
FPS: 30, latency: 2.0 ms
Tracking ID: 45
Plant (index: 4)
63.67% confidence (index: 4
Tracking ID: 43
Home good (inde
92.19% confidenc
Tracking ID: 41
Home good (index: 1)
89.45% confidence (index: 1)

Object Detection

InputImage size: 640x360
FPS: 18, Frame latency: 54 ms
Detector latency: 39 ms

squats_down : 4 reps
squats_down : 1.00 confidence

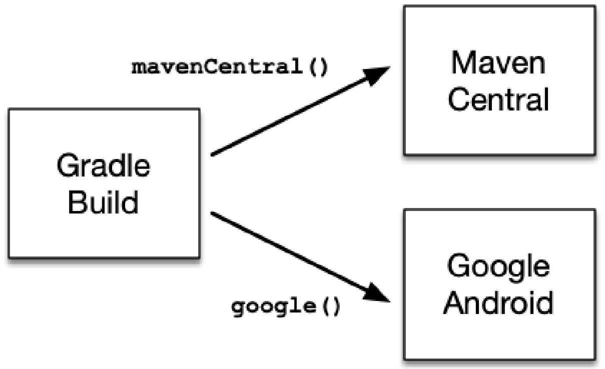
Pose Detection

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android
https://docs.gradle.org/current/userguide/dependency_management.html#sec:repositories

- Before starting with ML kit
  - Dependency resolution: process of downloading required libraries from the repositories holding them
  - Maven repository contains various support libraries required by Android apps
  - In project-level **build.gradle** file:
    - Include Google's Maven repository in both **buildscript** and **allprojects** sections

```
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
    }
}
```



Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- Add dependencies for ML Kit Android libraries to app-level gradle file (usually **app/build.gradle**) either
  a) Bundle model with app:

```
dependencies {
  // ...
  // Use this dependency to bundle the model with your app
  implementation 'com.google.mlkit:face-detection:16.1.5'
}
```

  b) Or use model in **Google Play Services** (broad set of Android SDKs. Once installed, Google can update SDK anytime)

```
dependencies {
  // ...
  // Use this dependency to use the dynamically downloaded model in Google Play Services
  implementation 'com.google.android.gms:play-services-mlkit-face-detection:17.1.0'
}
```

  c) If using **Google Play services**, add following declaration to **Android Manifest** to download model after app installed

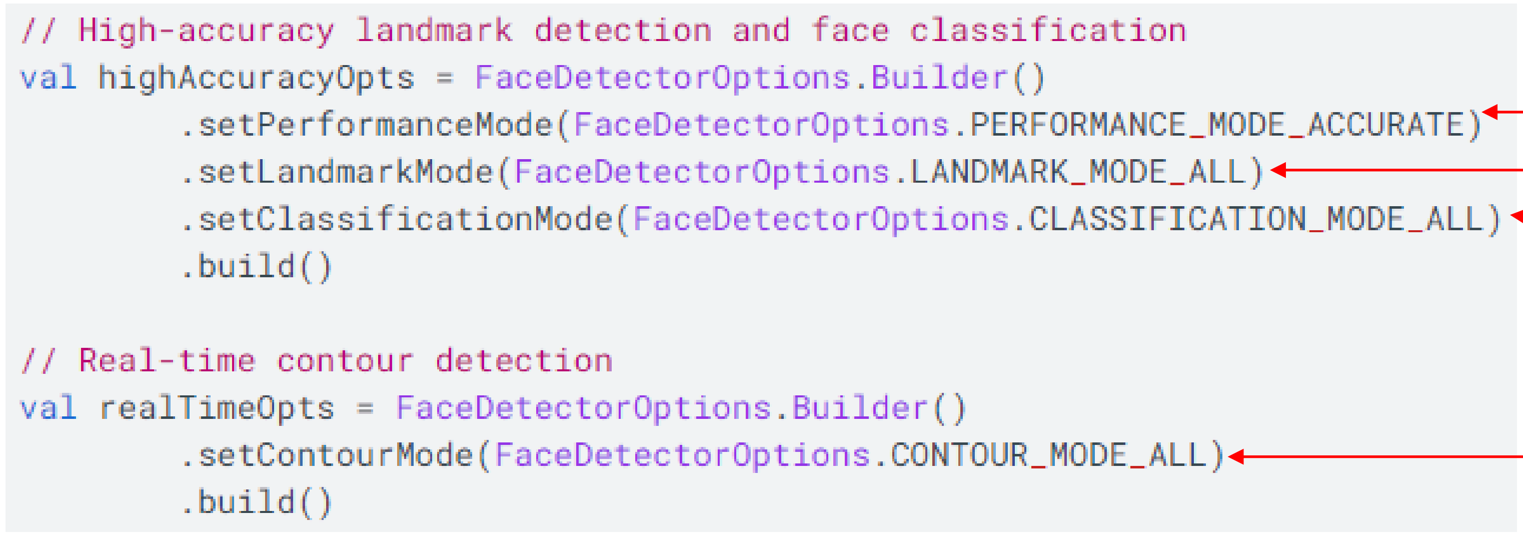
```
<application ...>
      ...
      <meta-data
          android:name="com.google.mlkit.vision.DEPENDENCIES"
          android:value="face" >
      <!-- To use multiple models: android:value="face,model2,model3" -->
</application>
```

Google Play Services

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- Some ML kit face detection rules:
  - Dimensions of image to analyzed at least 480 x 360 pixels
  - Each face to be detected must be “large enough” (at least 100 x 100 pixels)
  - To detect contours, each face must be at least 200 x 200 pixels
- First configure the face detector: In file **FaceDetectionActivity.kt** kotlin file



```kotlin
// High-accuracy landmark detection and face classification
val highAccuracyOpts = FaceDetectorOptions.Builder()
        .setPerformanceMode(FaceDetectorOptions.PERFORMANCE_MODE_ACCURATE)
        .setLandmarkMode(FaceDetectorOptions.LANDMARK_MODE_ALL)
        .setClassificationMode(FaceDetectorOptions.CLASSIFICATION_MODE_ALL)
        .build()

// Real-time contour detection
val realTimeOpts = FaceDetectorOptions.Builder()
        .setContourMode(FaceDetectorOptions.CONTOUR_MODE_ALL)
        .build()
```

Select accuracy over speed

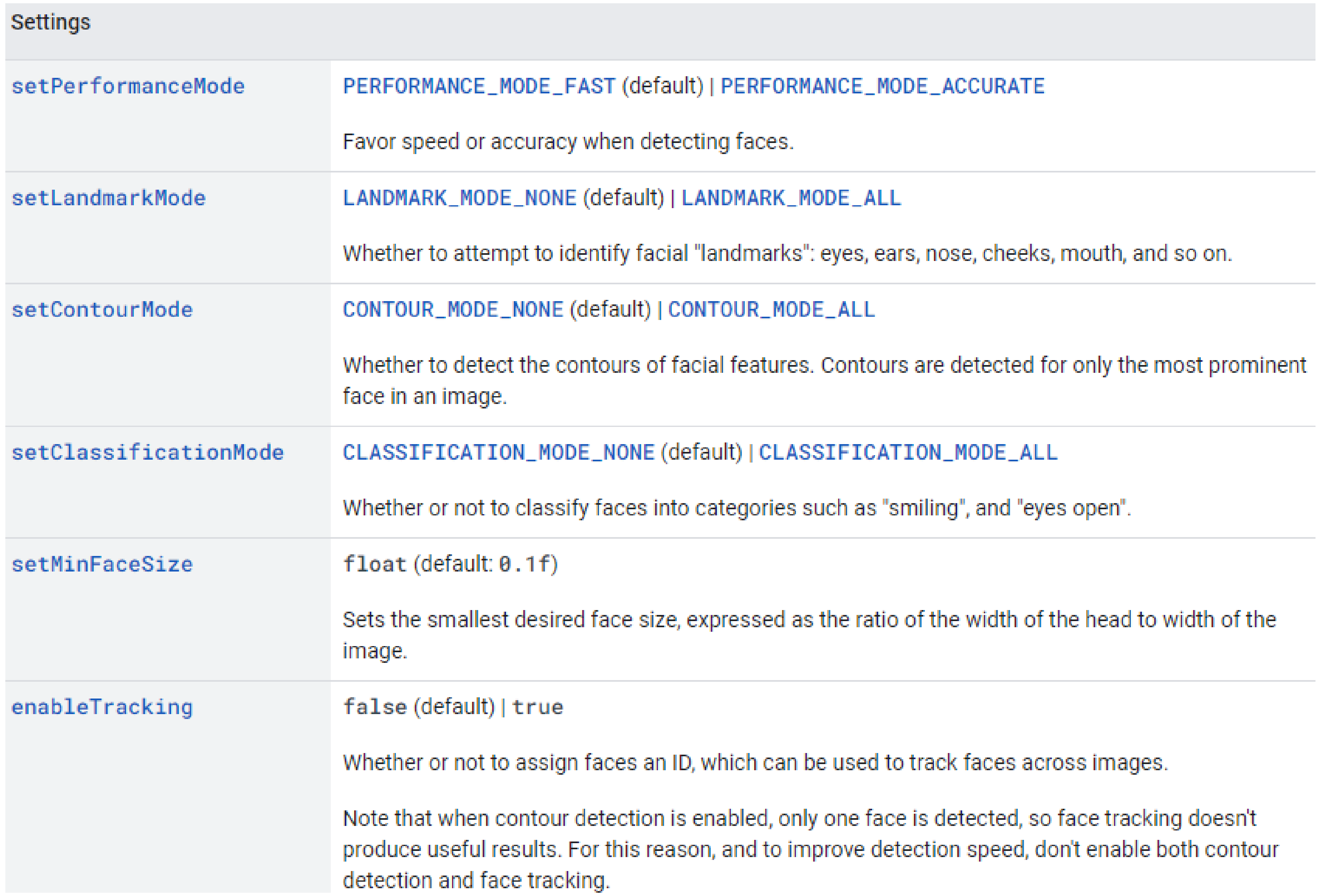
Detect all facial landmarks (eyes, nose, mouth, etc.)

Classify state of features (e.g. eyes open, mouth smiling)

Detect contours of most prominent face

Face Detection using ML Kit: Configurable Parameters

https://developers.google.com/ml-kit/vision/face-detection/android

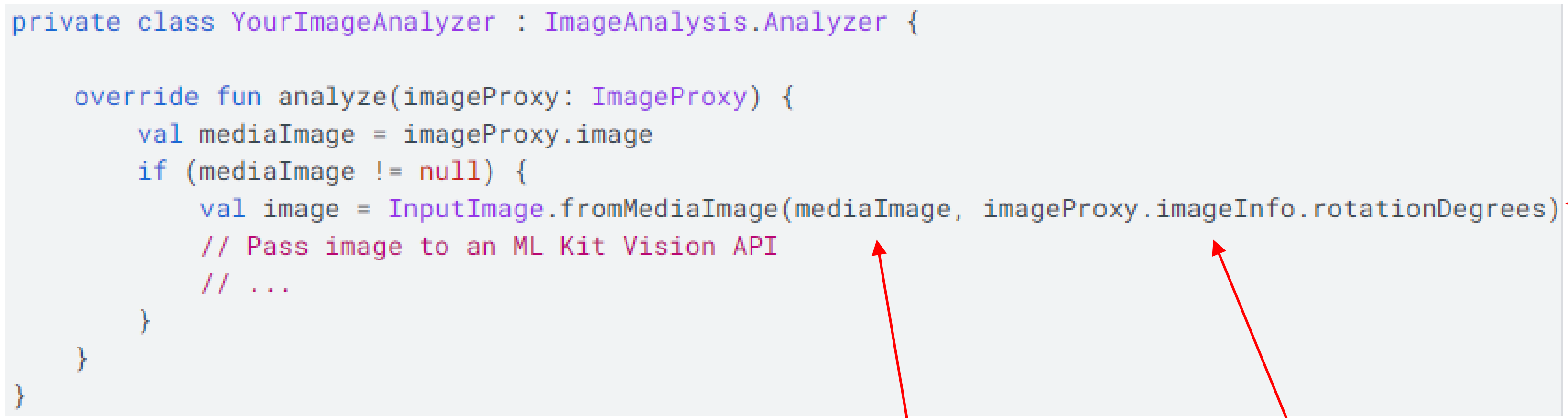


Settings	
`setPerformanceMode`	`PERFORMANCE_MODE_FAST` (default) | `PERFORMANCE_MODE_ACCURATE` Favor speed or accuracy when detecting faces.
`setLandmarkMode`	`LANDMARK_MODE_NONE` (default) | `LANDMARK_MODE_ALL` Whether to attempt to identify facial "landmarks": eyes, ears, nose, cheeks, mouth, and so on.
`setContourMode`	`CONTOUR_MODE_NONE` (default) | `CONTOUR_MODE_ALL` Whether to detect the contours of facial features. Contours are detected for only the most prominent face in an image.
`setClassificationMode`	`CLASSIFICATION_MODE_NONE` (default) | `CLASSIFICATION_MODE_ALL` Whether or not to classify faces into categories such as "smiling", and "eyes open".
`setMinFaceSize`	`float` (default: `0.1f`) Sets the smallest desired face size, expressed as the ratio of the width of the head to width of the image.
`enableTracking`	`false` (default) | `true` Whether or not to assign faces an ID, which can be used to track faces across images. Note that when contour detection is enabled, only one face is detected, so face tracking doesn't produce useful results. For this reason, and to improve detection speed, don't enable both contour detection and face tracking.

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- Prepare input image: Create **InputImage** object from Bitmap, media.Image, ByteBuffer or file on device
- Then pass **InputImage** object to **FaceDetector**'s **process** method
- To create **InputImage** from a **media.Image** object (e.g. when image is captured using camera)
  - Pass **media.Image** object and image's rotation to **InputImage.fromMediaImage( )**



```
private class YourImageAnalyzer : ImageAnalysis.Analyzer {

    override fun analyze(imageProxy: ImageProxy) {
        val mediaImage = imageProxy.image
        if (mediaImage != null) {
            val image = InputImage.fromMediaImage(mediaImage, imageProxy.imageInfo.rotationDegrees)
            // Pass image to an ML Kit Vision API
            // ...
        }
    }
}
```

Create InputImage from mediaImage (captured by camera)

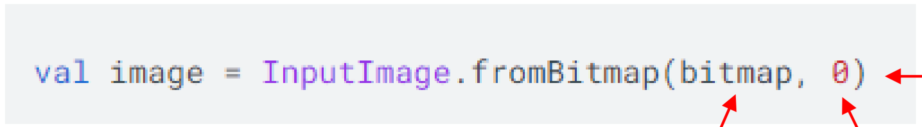
Media Image Object (image captured using Camera)

Rotation of image

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- To create **InputImage** object from a **Bitmap** object



```kotlin
val image = InputImage.fromBitmap(bitmap, 0)
```

Create InputImage from bitmap

Bitmap object

Rotation of image

- See documentation for how to create InputImage from file URI, ByteBuffer or ByteArray

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- Get instance of **FaceDetector**
  - Initialize either with preferred options, or use defaults

```
val detector = FaceDetection.getClient(options)
// Or, to use the default option:
// val detector = FaceDetection.getClient();
```

- Process the image: Pass the image to **process** method

```
val result = detector.process(image)
        .addOnSuccessListener { faces ->
            // Task completed successfully
            // ...
        }
        .addOnFailureListener { e ->
            // Task failed with an exception
            // ...
        }
```

Process image

Insert steps to be taken if process method succeeds

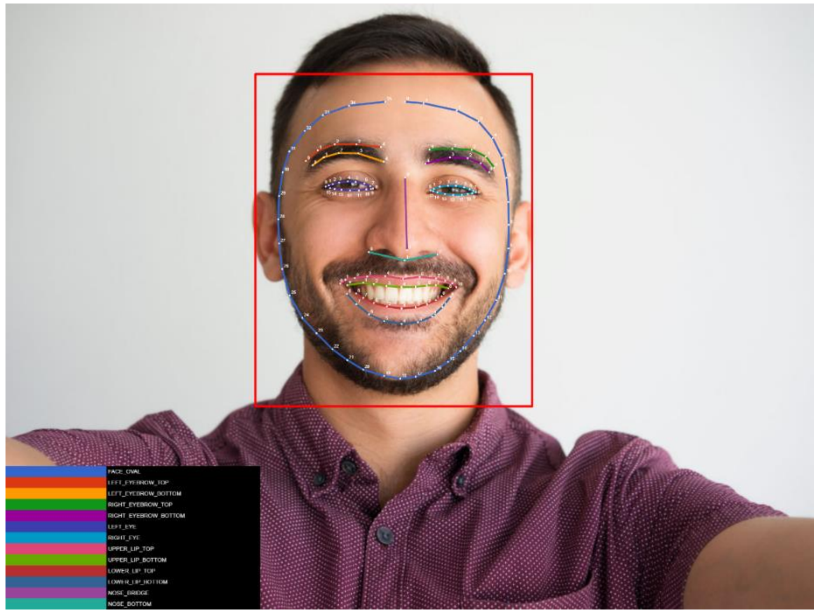
Insert steps to be taken if process method fails

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android

- Get information about detected faces from list of detected faces

```
for (face in faces) {
    val bounds = face.boundingBox
    val rotY = face.headEulerAngleY // Head is rotated to the right rotY degrees
    val rotZ = face.headEulerAngleZ // Head is tilted sideways rotZ degrees

    // If landmark detection was enabled (mouth, ears, eyes, cheeks, and
    // nose available):
    val leftEar = face.getLandmark(FaceLandmark.LEFT_EAR)
    leftEar?.let {
        val leftEarPos = leftEar.position
    }

    // If contour detection was enabled:
    val leftEyeContour = face.getContour(FaceContour.LEFT_EYE)?.points
    val upperLipBottomContour = face.getContour(FaceContour.UPPER_LIP_BOTTOM)?.points

    // If classification was enabled:
    if (face.smilingProbability != null) {
        val smileProb = face.smilingProbability
    }
    if (face.rightEyeOpenProbability != null) {
        val rightEyeOpenProb = face.rightEyeOpenProbability
    }

    // If face tracking was enabled:
    if (face.trackingId != null) {
        val id = face.trackingId
    }
}
```



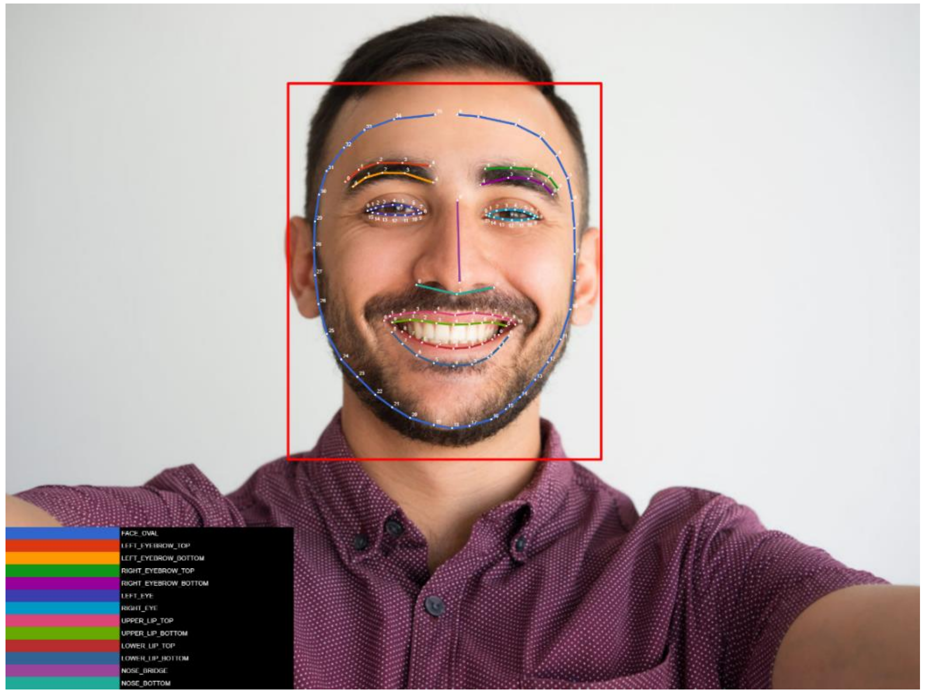
See ML kit website/documentation for how to detect faces in real-time application

Face Detection using ML Kit

https://developers.google.com/ml-kit/vision/face-detection/android
https://developers.google.com/ml-kit/vision/face-detection/face-detection-concepts#contours

- Different numbers of points used to represent shape of different facial contours

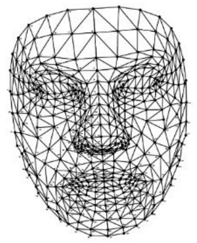
Face oval	36 points	Upper lip (top)	11 points
Left eyebrow (top)	5 points	Upper lip (bottom)	9 points
Left eyebrow (bottom)	5 points	Lower lip (top)	9 points
Right eyebrow (top)	5 points	Lower lip (bottom)	9 points
Right eyebrow (bottom)	5 points	Nose bridge	2 points
Left eye	16 points	Nose bottom	3 points
Right eye	16 points		
Left cheek (center)	1 point		
Right cheek (center)	1 points		

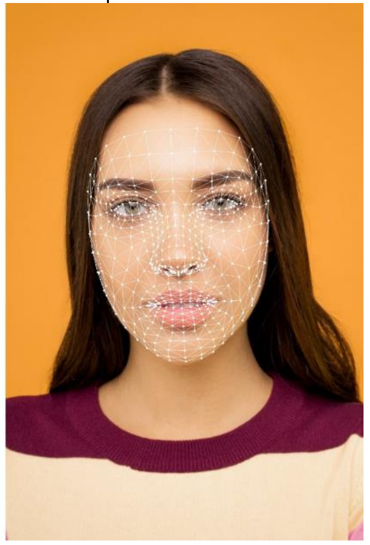


See ML kit website/documentation for how to detect faces in real-time application

Other ML Kit Modules

- Face mesh detection
  - https://developers.google.com/ml-kit/vision/face-mesh-detection/android


468 points



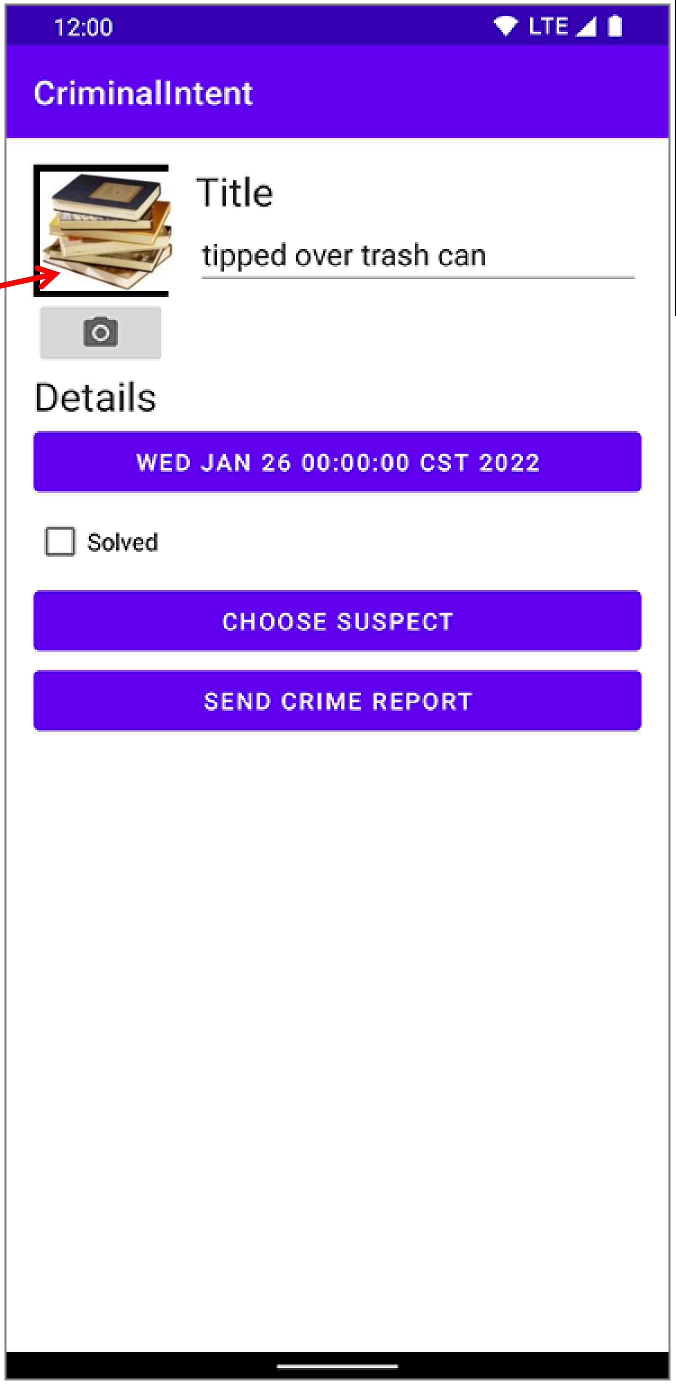
- Selfie segmentation
  - https://developers.google.com/ml-kit/vision/selfie-segmentation/android

Project 2

Project 2: Quick Walkthrough

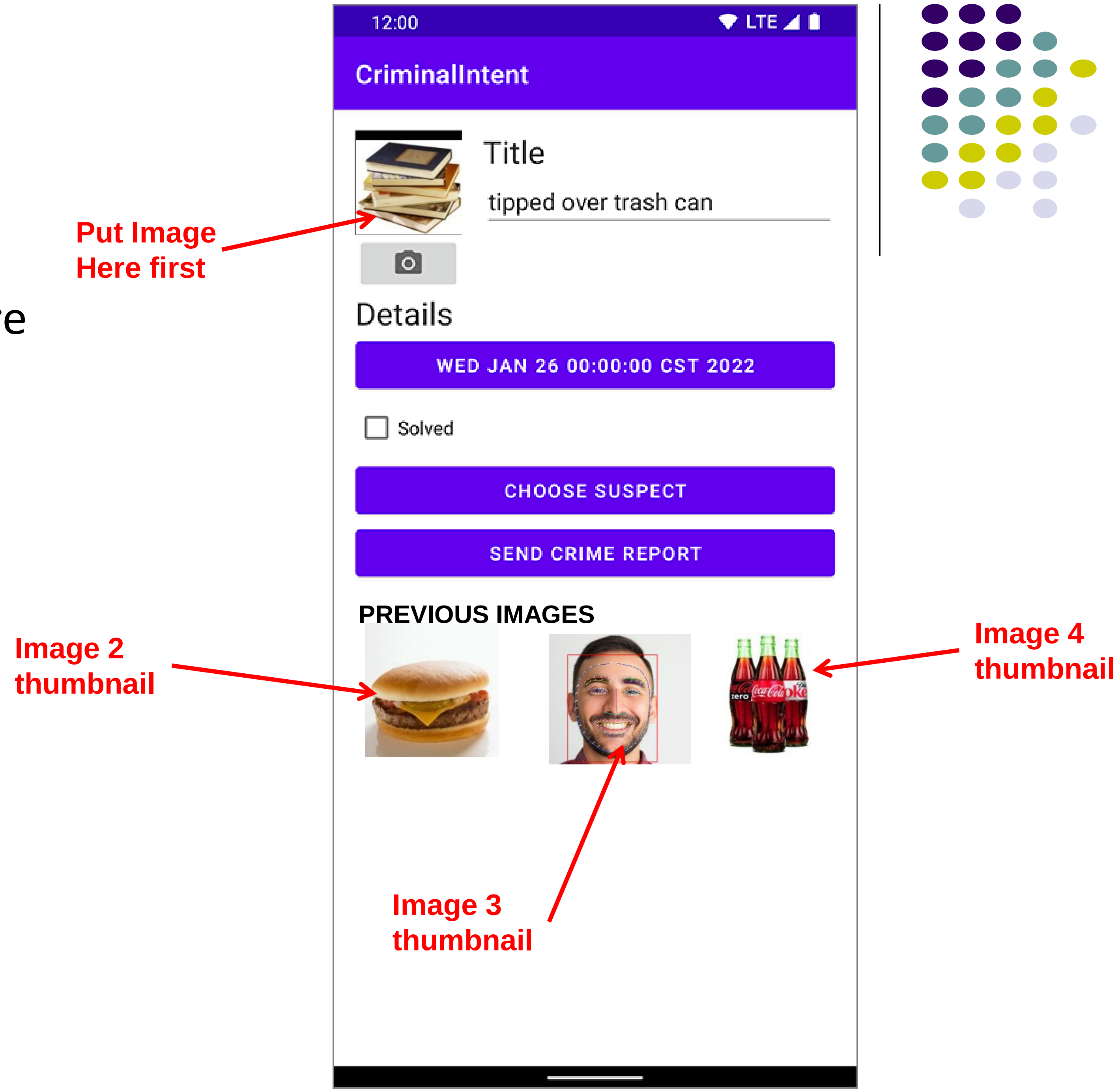
- Download and test out code for **CriminalIntent** (Chapter 19)
- Code already has Camera functionality
- Taking a picture inserts picture in the top left corner
- Taking picture 2
  - Picture 2 thumbnail replaces picture 1 thumbnail

Taking picture inserts picture here

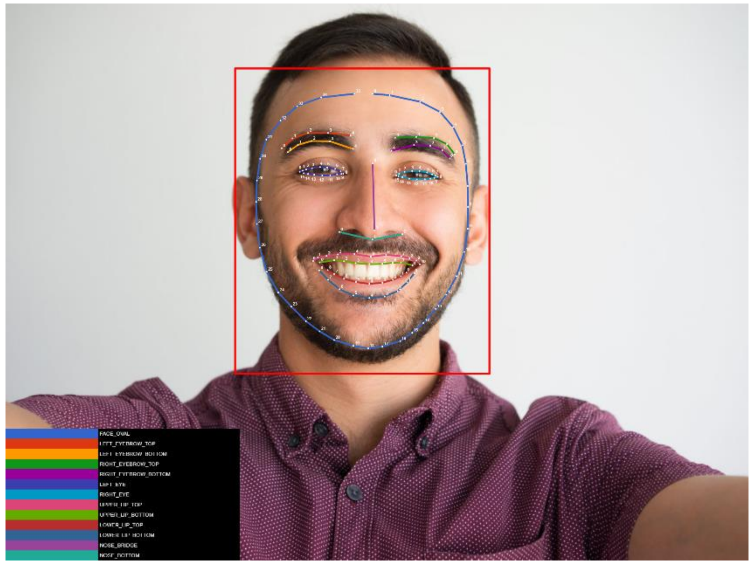


Project 2: Quick Walkthrough

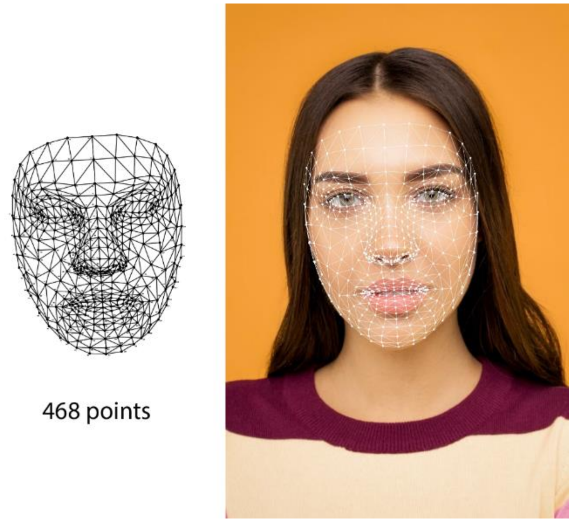
- First, make it possible to show thumbnails of 4 pictures (add 3 more thumbnails)
- After 4 thumbnails displayed, start replacing thumbnails from image 5
- Taking image 5
  - Picture 5 thumbnail replaces picture 1 thumbnail
- Taking image 6
  - Picture 6 thumbnail replaces picture 2 thumbnail
- Etc…



Recall: ML Kit has functionality for Cool Facial Detection Stuff



Face Detection
Face Contour Detection



Face Mesh Generation

Selfie Segmentation

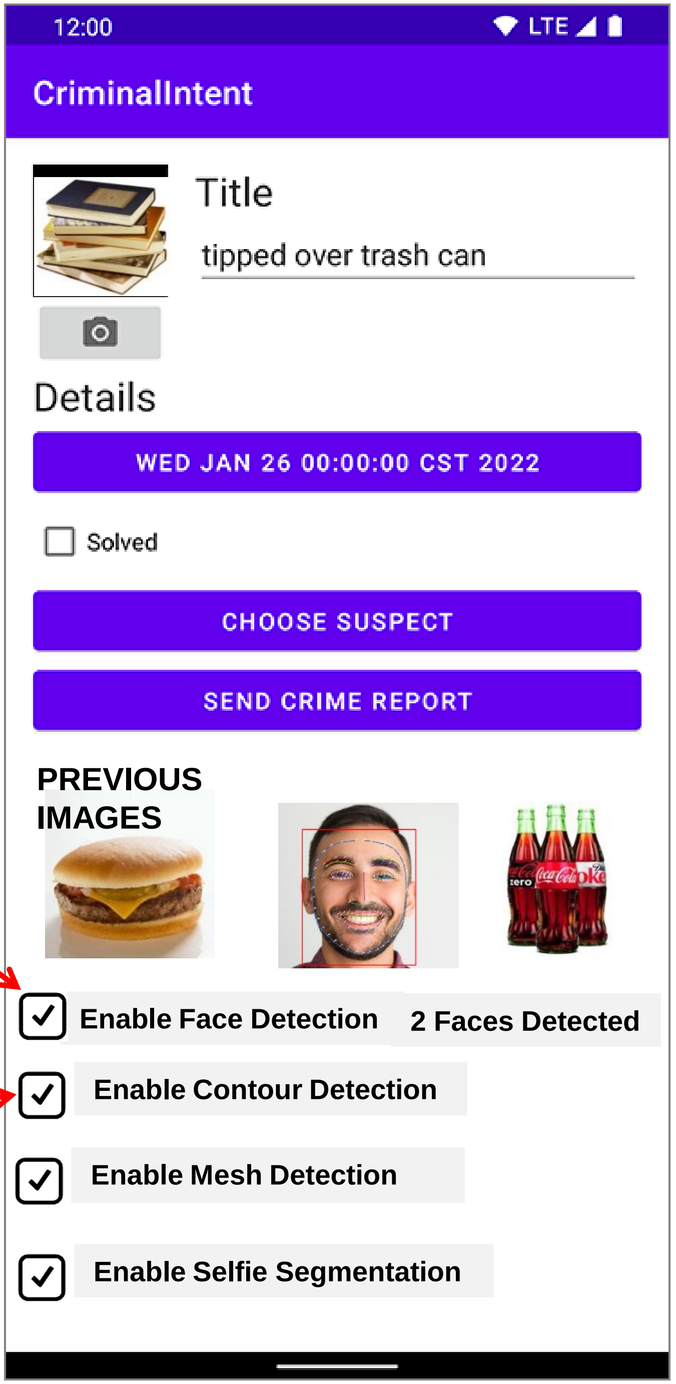
Project 2: Quick Walkthrough

- Using ML Kit, integrate:
  - Face detection + display number of faces in LAST picture
  - Face Contours detection
  - Mesh Detection
  - Selfie segmentation

- **Important Note:**
  - Code ALL PROJECTS in Kotlin where applicable (not Java) unless Kotlin is not an option

Checkbox to Enable/disable Face detection

Checkbox to Enable/disable Contour Detection



12:00 LTE

CriminalIntent

Title

tipped over trash can

Details

WED JAN 26 00:00:00 CST 2022

☐ Solved

CHOOSE SUSPECT

SEND CRIME REPORT

PREVIOUS IMAGES

☑ Enable Face Detection 2 Faces Detected

☑ Enable Contour Detection

☑ Enable Mesh Detection

☑ Enable Selfie Segmentation

References

- Android Nerd Ranch, 5th Edition
- ML Kit online