1 Face Dectection API

Huawei, Samsung acnounced AI-featured inteligeng cellphones recently. However, we could release the AI-powered ability for older cellphone too.

2 Updated

1. android studio, 3.4
2. gradle, 5.1.1
3. external kotlin plugin, 1.3.30
4. JDK-8 build 211

3 Steps

Create new kotlin project, **app: facedetectapp, company: ai.kotlin.io**

0. If want to tyr Firebase feature, a mobile app development platform, enable certain plugins from right buttom menu, [Configure] ➡ [Plugins] ➡ all plugins beginning with Firebase.
1. Android target SDK-28 (Pie) required, install it from

   ```
   [Tools] ➜ [SDK Manager] ➜ [Android SDK] ➜ [SDK platfor
   m] (☑ Show Packages Details)
            ▽ Android 9.0 Pie
             ☑ Android SDK Platform 28
   ```

   also install intel x86 Atom_64 System Image if want to test it on the fly.
2. create new project, App name: FaceDectectApp (Company name: ai.kotlin.io), as usual; and set the SDK conditions: **minSdkVersion 21, targetSdkVersion 28**.
3. modify [build.gradle (Project: FaceDectectApp)]:

```
buildscript {
    ...
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.4.0'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugi
n:$kotlin_version"
        classpath 'com.google.gms:google-services:4.2.0'
    }
}
allprojects {
    repositories {
        jcenter()
        maven { url "https://jitpack.io" }
        google()
    }
}
...
```

- modify [build.gradle (Module: app)]: here the entire list of dependent libraries:

```
...
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar']
)
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7
:$kotlin_version"

    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.0.0
'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.github.husaynhakeem:android-face-de
tector:v1.2'
    implementation 'com.otaliastudios:cameraview:1.6.0'
    implementation 'com.google.firebase:firebase-core:16.0.
5'
    implementation 'com.google.android.gms:play-services-vi
sion:11.8.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:run
ner:1.0.2'
    androidTestImplementation 'com.android.support.test.esp
resso:espresso-core:3.0.2'
}

apply plugin: 'com.google.gms.google-services'
```

and **sync** project.

- Set up [Firebase] of the project:
  - Click [Tools > Firebase] to open the Assistant window.
  - Click to expand one of the listed features (for example, Analytics), then click the provided tutorial link (for example, Log an Analytics event).
    - Click the Connect to Firebase button] to connect to Firebase and add the necessary code to your app.
    - register Firebase
      - [Firebase Console] [events] [Android] and download the file, `google-services.json` , and put it on the directory, `$Project/app/` .
- To make Face detector app is easy by Firebase ML Kit's face detection API. Here, we introduce how to create such app based on "android-face-detector", a library on top of Firebase ML Kit's face detection API. In brief, face-detect app is designed in three steps:
  - Add a FaceBoundsOverlay on top of your camera view:

```
<FrameLayout  ...>
 <CameraView
     ... />
 <husaynhakeem.io.facedetector.FaceBoundsOverlay
     ... />
</FrameLayout>
```

- Define a FaceDetection instance and connect it to camera o device:

```
private val faceDetector: FaceDetector by lazy {
 FaceDetector(facesBoundsOverlay)
}
...
cameraView.addFrameProcessor {
 faceDetector.process(Frame(
      data = it.data,
      rotation = it.rotation,
      size = Size(it.size.width, it.size.height),
      format = it.format,
      isCameraFacingBack = cameraView.facing))
}
```

- Setup firebase in this project.

4 Details

1 UI, acyivity_main.xml: use cameraview of otaliastudios:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/
res/android"
             xmlns:tools="http://schemas.android.com/tools"
             android:layout_width="match_parent"
             android:layout_height="match_parent"
             tools:context=".MainActivity">

    <com.otaliastudios.cameraview.CameraView
            android:id="@+id/cameraView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:keepScreenOn="true" />

    <husaynhakeem.io.facedetector.FaceBoundsOverlay
            android:id="@+id/facesBoundsOverlay"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    <android.support.design.widget.FloatingActionButton
            android:id="@+id/revertCameraButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|end"
            android:layout_margin="16dp"
            android:src="@drawable/camera" />

</FrameLayout>
```

Within the last UI object, FloatingActionButton, we add a icon picture, called camera.png, which was placed within $Res/drawable sub-folder.

- kotlin part: init faceDectector with cmeraview, startup after setup:

```
package io.kotlin.ai.facedetectapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import com.otaliastudios.cameraview.Facing
import husaynhakeem.io.facedetector.FaceDetector
import husaynhakeem.io.facedetector.models.Frame
import husaynhakeem.io.facedetector.models.Size
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    private val faceDetector: FaceDetector by lazy {
        FaceDetector(facesBoundsOverlay)
    }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setupCamera()
    }

    private fun setupCamera() {
        cameraView.addFrameProcessor {
            faceDetector.process(Frame(
                data = it.data,
                rotation = it.rotation,
                size = Size(it.size.width, it.size.height),
                format = it.format,
                isCameraFacingBack = cameraView.facing == F
acing.BACK))
        }
        // Toggles the facing value between Facing.FRONT an
d Facing.BACK.
        revertCameraButton.setOnClickListener {
            cameraView.toggleFacing()
        }
    }

    override fun onResume() {
        super.onResume()
        cameraView.start()
    }

    override fun onPause() {
        super.onPause()
        cameraView.stop()
    }

    override fun onDestroy() {
        super.onDestroy()
        cameraView.destroy()
    }
}
```

Happy ending ...

Face Dectection API

Huawei, Samsung acnounced AI-featured inteligeng cellphones recently. However, we could release the AI-powered ability for older cellphone too.

Updated

1. android studio, 3.4
2. gradle, 5.1.1
3. external kotlin plugin, 1.3.30
4. JDK-8 build 211