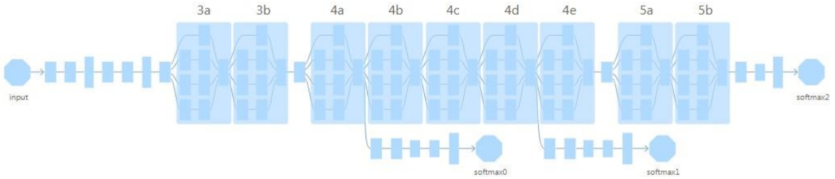
Google TensorFlow.js

浙江大学城市学院
ZHEJIANG UNIVERSITY CITY COLLEGE



深度学习应用开发

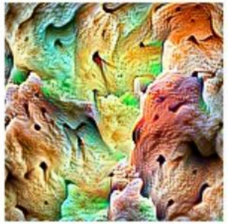
基于TensorFlow的实践

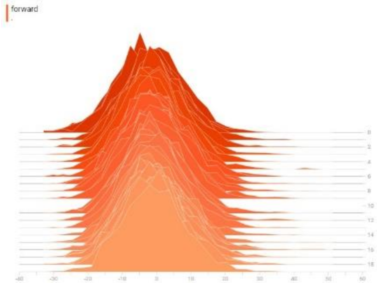
吴明晖 李卓蓉 金苍宏

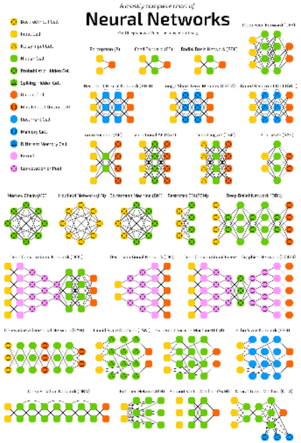
浙江大学城市学院

计算机与计算科学学院

Dept. of Computer Science
Zhejiang University City College







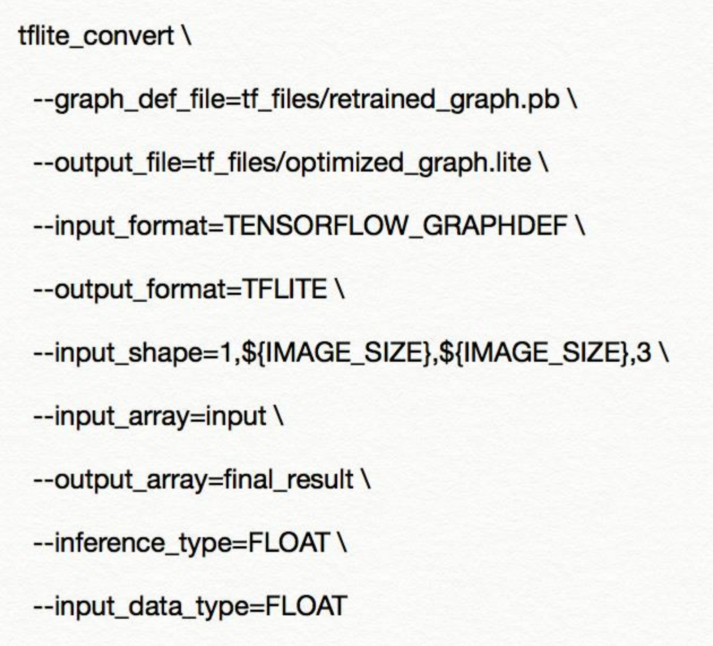
TFLite应用：手机识别花型

步骤4. 验证新模型

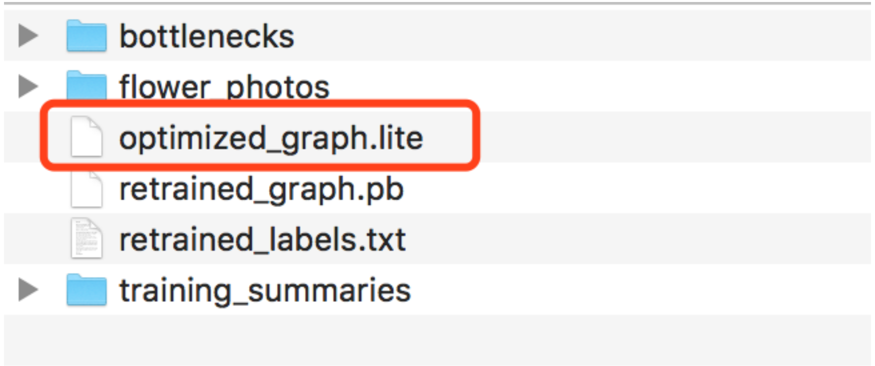
```
python3 label_image.py \

--graph=tf_files/retrained_graph.pb  \

--image=tf_files/flower_photos/daisy/21652746_cc379e0eea_m.jpg \

--input_layer=Placeholder
```

```
daisy 0.48343602
dandelion 0.34764665
sunflowers 0.138637
tulips 0.019344615
roses 0.010935791
```

步骤5. 转化TFLite模型

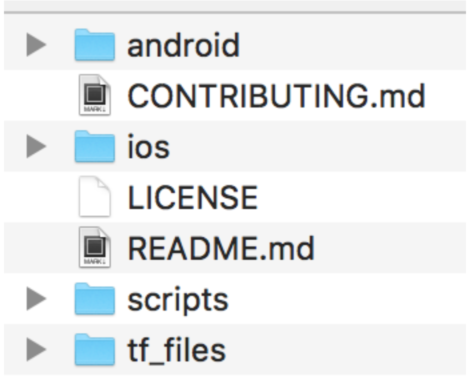


```
tflite_convert \
  --graph_def_file=tf_files/retrained_graph.pb \
  --output_file=tf_files/optimized_graph.lite \
  --input_format=TENSORFLOW_GRAPHDEF \
  --output_format=TFLITE \
  --input_shape=1,${IMAGE_SIZE},${IMAGE_SIZE},3 \
  --input_array=input \
  --output_array=final_result \
  --inference_type=FLOAT \
  --input_data_type=FLOAT
```

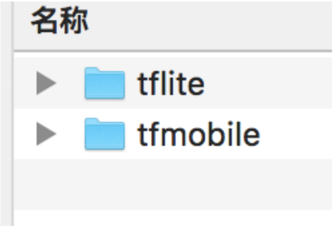


```
▶ bottlenecks
▶ flower_photos
  optimized_graph.lite
  retrained_graph.pb
  retrained_labels.txt
▶ training_summaries
```

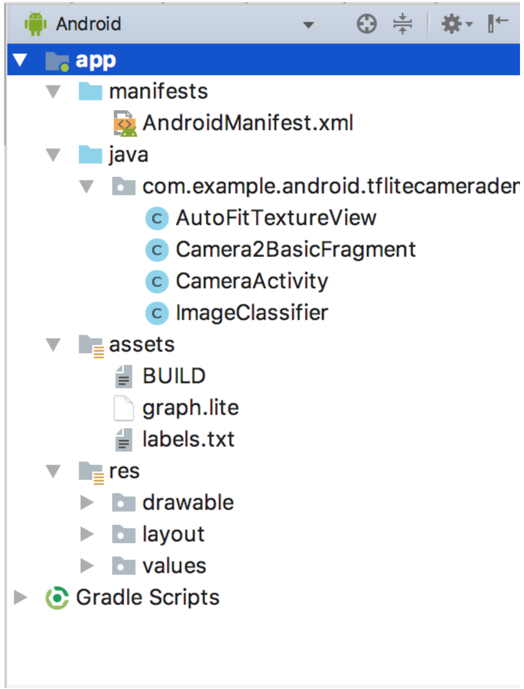
步骤6. 下载Android项目



```
▶ android
  CONTRIBUTING.md
▶ ios
  LICENSE
  README.md
▶ scripts
▶ tf_files
```

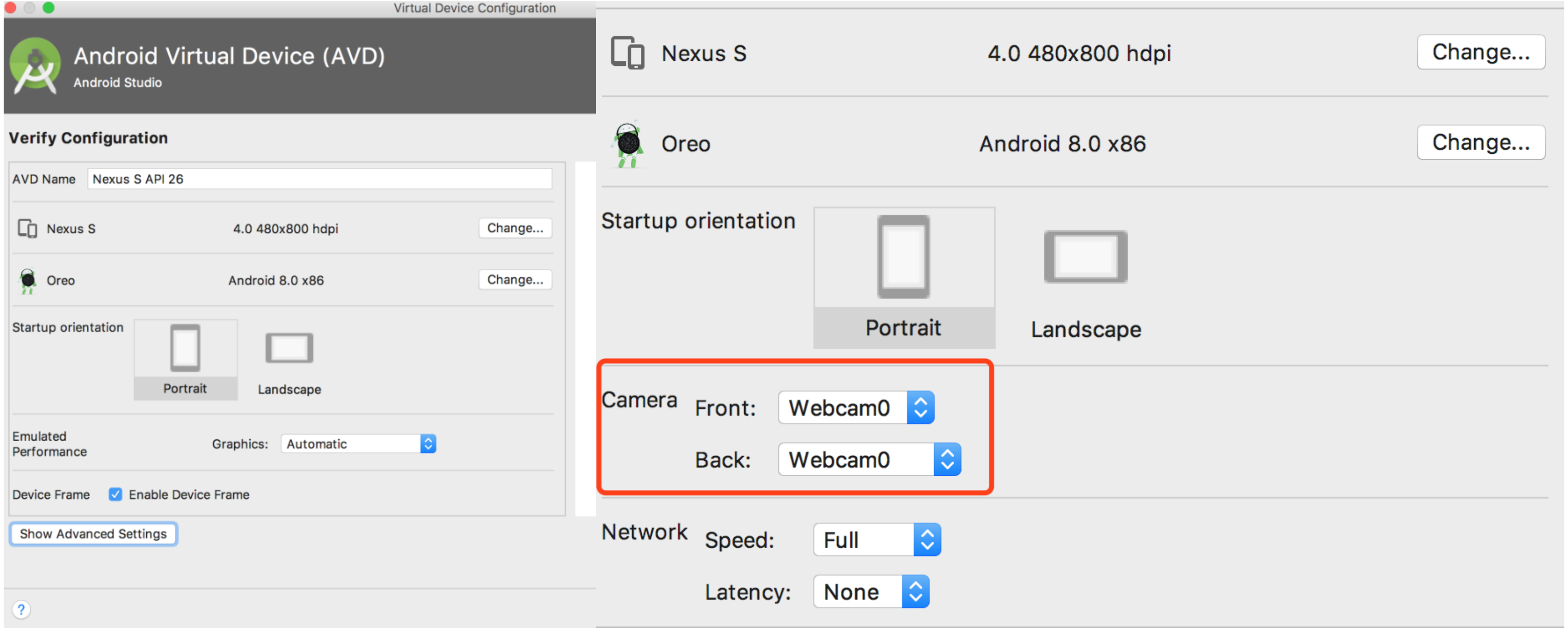


```
名称
▶ tflite
▶ tfmobile
```



```
Android
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.example.android.tflitecamerader
        AutoFitTextureView
        Camera2BasicFragment
        CameraActivity
        ImageClassifier
  ▼ assets
      BUILD
      graph.lite
      labels.txt
  ▼ res
    ▶ drawable
    ▶ layout
    ▶ values
▶ Gradle Scripts
```

步骤6. 设置AVD



Virtual Device Configuration

Android Virtual Device (AVD)
Android Studio

Verify Configuration

AVD Name: Nexus S API 26

Nexus S 4.0 480x800 hdpi Change...

Oreo Android 8.0 x86 Change...

Startup orientation: Portrait Landscape

Emulated Performance Graphics: Automatic

Device Frame ☑ Enable Device Frame

Show Advanced Settings

Nexus S 4.0 480x800 hdpi Change...

Oreo Android 8.0 x86 Change...

Startup orientation: Portrait Landscape

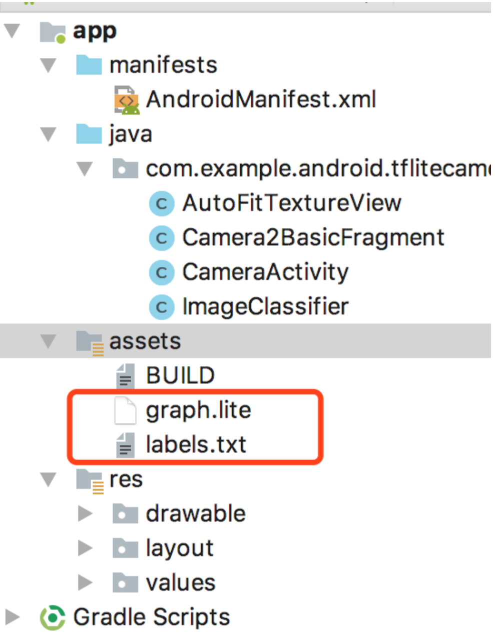
Camera Front: Webcam0
Back: Webcam0

Network Speed: Full
Latency: None

步骤7. 替换模型文件

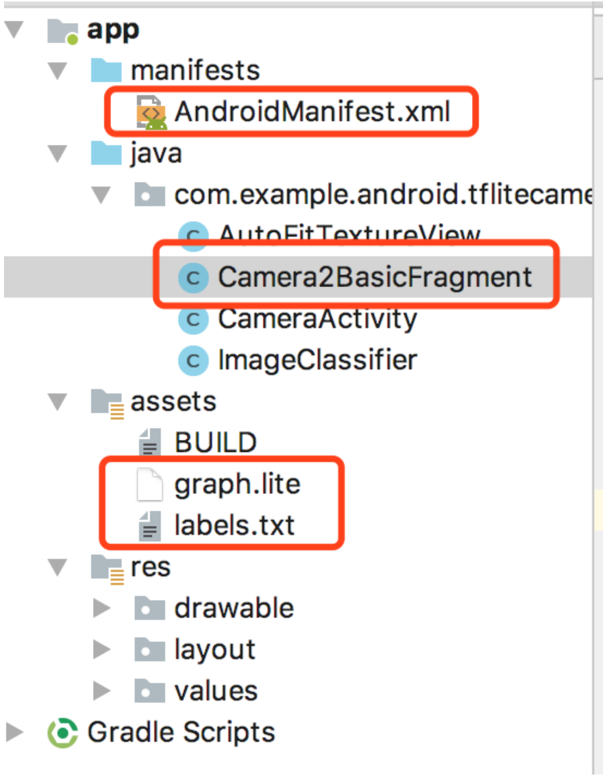
- cp tf_files/optimized_graph.lite android/tflite/app/src/main/assets/graph.lite

- cp tf_files/retrained_labels.txt android/tflite/app/src/main/assets/labels.txt



```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.example.android.tflitecam
        AutoFitTextureView
        Camera2BasicFragment
        CameraActivity
        ImageClassifier
  ▼ assets
      BUILD
      graph.lite
      labels.txt
  ▼ res
    ▶ drawable
    ▶ layout
    ▶ values
▶ Gradle Scripts
```

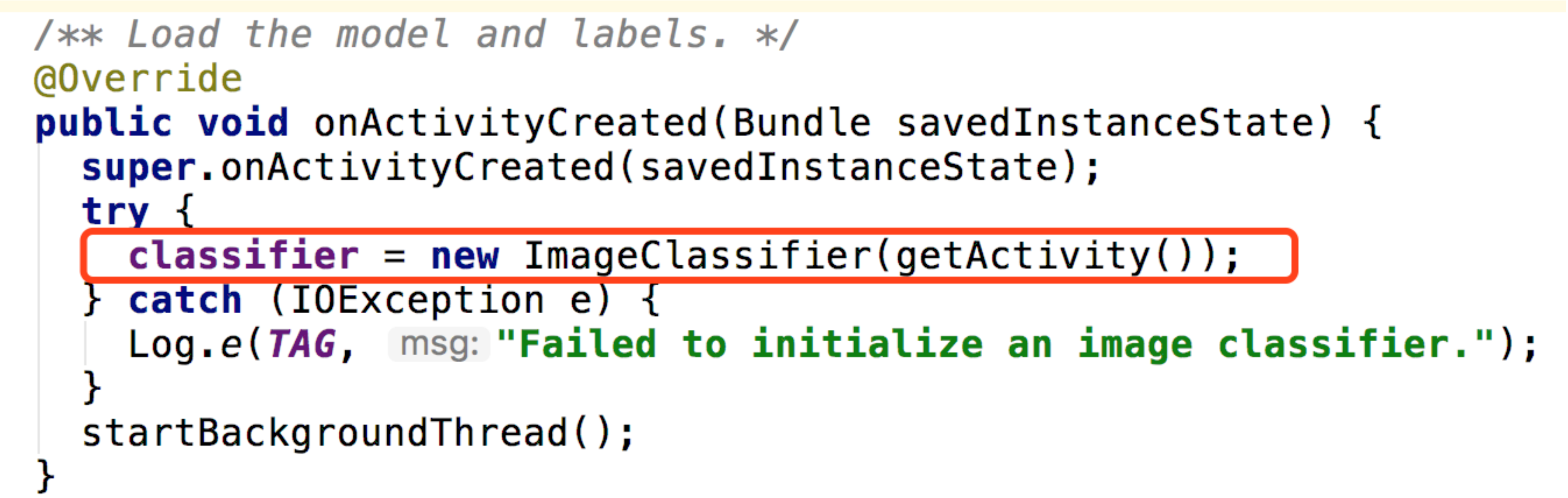
步骤8. 代码片段



```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.example.android.tflitecame
        AutoFitTextureView
        Camera2BasicFragment
        CameraActivity
        ImageClassifier
  ▼ assets
      BUILD
      graph.lite
      labels.txt
  ▼ res
    ▶ drawable
    ▶ layout
    ▶ values
▶ Gradle Scripts
```

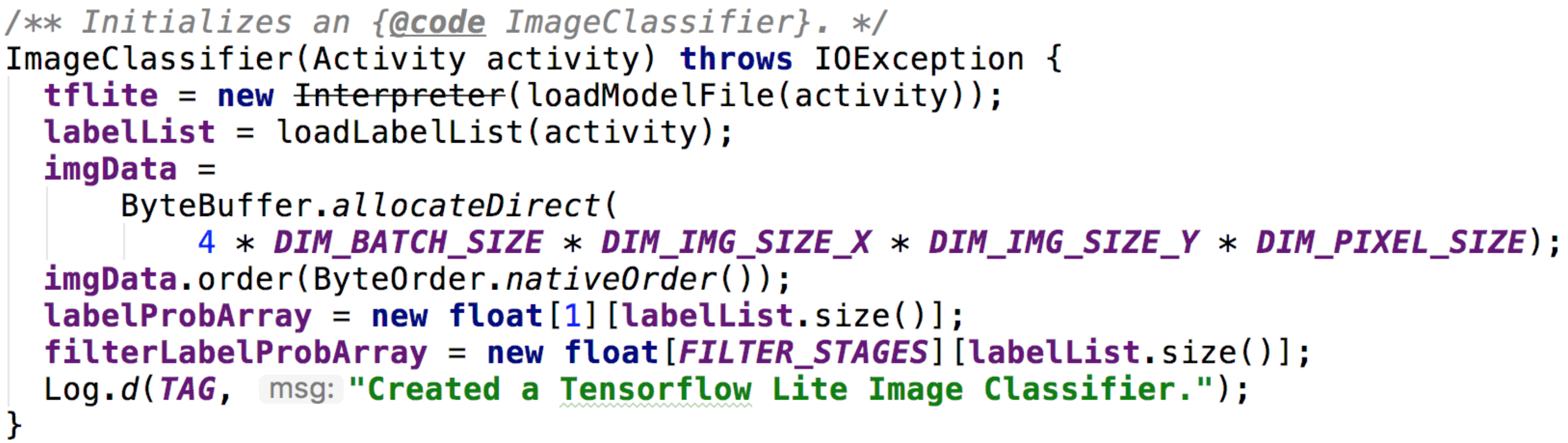
```
<uses-permission android:name="android.permission.CAMERA" />

<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<uses-sdk android:minSdkVersion="21" />
```

步骤8. 代码片段

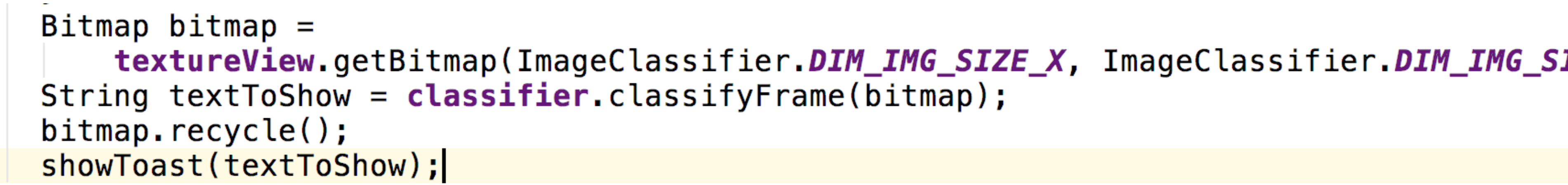


```
/** Load the model and labels. */
@Override
public void onActivityCreated(Bundle savedInstanceState) {
  super.onActivityCreated(savedInstanceState);
  try {
    classifier = new ImageClassifier(getActivity());
  } catch (IOException e) {
    Log.e(TAG, msg: "Failed to initialize an image classifier.");
  }
  startBackgroundThread();
}
```

步骤8. 代码片段

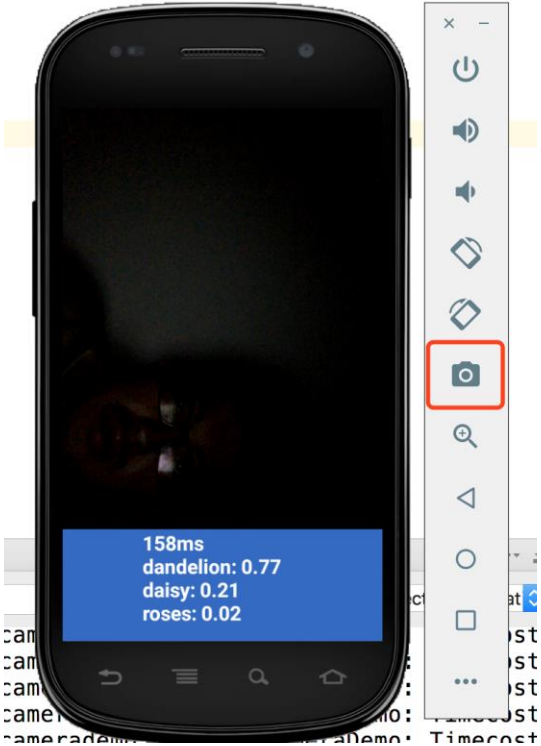


```
/** Initializes an {@code ImageClassifier}. */
ImageClassifier(Activity activity) throws IOException {
  tflite = new ~~Interpreter~~(loadModelFile(activity));
  labelList = loadLabelList(activity);
  imgData =
      ByteBuffer.allocateDirect(
          4 * DIM_BATCH_SIZE * DIM_IMG_SIZE_X * DIM_IMG_SIZE_Y * DIM_PIXEL_SIZE);
  imgData.order(ByteOrder.nativeOrder());
  labelProbArray = new float[1][labelList.size()];
  filterLabelProbArray = new float[FILTER_STAGES][labelList.size()];
  Log.d(TAG, msg: "Created a Tensorflow Lite Image Classifier.");
}
```

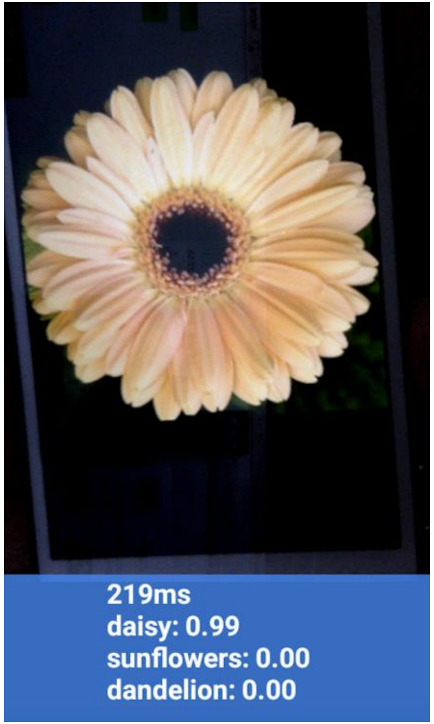


```
Bitmap bitmap =
    textureView.getBitmap(ImageClassifier.DIM_IMG_SIZE_X, ImageClassifier.DIM_IMG_SI
String textToShow = classifier.classifyFrame(bitmap);
bitmap.recycle();
showToast(textToShow);
```

步骤9. 模拟器运行



```
158ms
dandelion: 0.77
daisy: 0.21
roses: 0.02
```



```
219ms
daisy: 0.99
sunflowers: 0.00
dandelion: 0.00
```