

Documentation

This file specifies the working of object classification application and the mobilenet model which is used in the android app.

The java classes and their description is given below:

Class Summary

Class	Description
AutoFitTextureView	A TextureView that can be adjusted to a specified aspect ratio.
Camera2BasicFragment	Basic fragments for the Camera.
Camera2BasicFragment.ErrorDialog	Shows an error message dialog.
CameraActivity	Main Activity class for the Camera app.
GpuDelegateHelper	Helper class for GpuDelegate.
ImageClassifier	Classifies images with Tensorflow Lite.
ImageClassifierFloatInception	This classifier works with the Inception-v3 slim model.
ImageClassifierFloatMobileNet	This classifier works with the float MobileNet model.
ImageClassifierQuantizedMobileNet	This classifier works with the quantized MobileNet model.

The CameraActivity is the main Activity class which is launched when the app icon is clicked on the android device. Camera2BasicFragment takes the camera input and converts it into fragments on which the image classifier can acts on.

ImageClassifier class has the code that classifies object and put labels on it. The last three classes load the respective models, which had been converted using TFLiteConverter.

The `AutoFitTextureView` class:

```
com.example.android.tflitecamerademo
```

Class `AutoFitTextureView`

```
java.lang.Object  
    android.view.View  
        android.view.TextureView  
            com.example.android.tflitecamerademo.AutoFitTextureView
```

All Implemented Interfaces:

```
android.graphics.drawable.Drawable.Callback, android.view.accessibility.AccessibilityEventSource,  
android.view.KeyEvent.Callback
```

```
public class AutoFitTextureView  
    extends android.view.TextureView
```

A `TextureView` that can be adjusted to a specified aspect ratio.

This class sets the user interface accordingly to the device's aspect ratio and switch between interfaces for landscape and portrait mode.

The `Camera2BasicFragment` class:

```
com.example.android.tflitecamerademo
```

Class `Camera2BasicFragment`

```
java.lang.Object  
    android.app.Fragment  
        com.example.android.tflitecamerademo.Camera2BasicFragment
```

All Implemented Interfaces:

```
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,  
android.support.v13.app.FragmentCompat.OnRequestPermissionsResultCallback, android.view.View.OnCreateContextMenuListener
```

```
public class Camera2BasicFragment  
    extends android.app.Fragment  
    implements android.support.v13.app.FragmentCompat.OnRequestPermissionsResultCallback
```

Basic fragments for the Camera.

The CameraActivity(Main App Activity) Class:

com.example.android.tflitecamerademo

Class CameraActivity

```
java.lang.Object
    android.content.Context
        android.content.ContextWrapper
            android.view.ContextThemeWrapper
                android.app.Activity
                    com.example.android.tflitecamerademo.CameraActivity
```

All Implemented Interfaces:

android.content.ComponentCallbacks, android.content.ComponentCallbacks2, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2, android.view.View.OnCreateContextMenuListener, android.view.Window.Callback

```
public class CameraActivity
    extends android.app.Activity
```

Main Activity class for the Camera app.

The MobileNet class:

com.example.android.tflitecamerademo

Class ImageClassifierQuantizedMobileNet

```
java.lang.Object
    com.example.android.tflitecamerademo.ImageClassifier
        com.example.android.tflitecamerademo.ImageClassifierQuantizedMobileNet
```

```
public class ImageClassifierQuantizedMobileNet
    extends ImageClassifier
```

This classifier works with the quantized MobileNet model.

Field Summary

Fields inherited from class com.example.android.tflitecamerademo.ImageClassifier

imgData, tflite

This class loads the quantized mobilenet model.

The `getModelPath()` and `getLabelPath()` are the most important methods of any of the three classes which load the models for classification.

Method Detail

`getModelPath`

```
protected java.lang.String getModelPath()
```

Description copied from class: `ImageClassifier`

Get the name of the model file stored in Assets.

Specified by:

`getModelPath` in class `ImageClassifier`

Returns:

`getLabelPath`

```
protected java.lang.String getLabelPath()
```

Description copied from class: `ImageClassifier`

Get the name of the label file stored in Assets.

Specified by:

`getLabelPath` in class `ImageClassifier`

Returns:

The other two methods calculate the probability and run inference on the model, respectively.

`getNormalizedProbability`

```
protected float getNormalizedProbability(int labelIndex)
```

Description copied from class: `ImageClassifier`

Get the normalized probability value for the specified label. This is the final value as it will be shown to the user.

Specified by:

`getNormalizedProbability` in class `ImageClassifier`

Returns:

`runInference`

```
protected void runInference()
```

Description copied from class: `ImageClassifier`

Run inference using the prepared input in `ImageClassifier.imgData`. Afterwards, the result will be provided by `getProbability()`.

This additional method is necessary, because we don't have a common base for different primitive data types.

Specified by:

`runInference` in class `ImageClassifier`

The GpuDelegateHelper class contains the code for running the processes on gpu, if available.

Modifier and Type	Method and Description
static org.tensorflow.lite.Delegate	createGpuDelegate() Returns an instance of GpuDelegate if available.
static boolean	isGpuDelegateAvailable() Checks whether GpuDelegate is available.
Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Method Detail

isGpuDelegateAvailable

```
public static boolean isGpuDelegateAvailable()
```

Checks whether GpuDelegate is available.

createGpuDelegate

```
public static org.tensorflow.lite.Delegate createGpuDelegate()
```

Returns an instance of GpuDelegate if available.

The mobile app uses the android NNAPI created by tensorflow, for hardware acceleration. The application can run upto 10 threads simultaneously for faster results. As the number of cores available on any modern gpu is high, the computation can be parallelized on available cores which can yield even better performance.