SuperMap iMobile

Video Recognition

SuperMap Software Co., Ltd.

11, 2019 • Beijing

Legal Statement

SuperMap Software Co., Ltd.

Address: 6/F Unit E, Building 107, No. A10, Jiuxianqiao North Road,

Chaoyang District, Beijing, 100015, CHINA

Tel: +86-10-59896503

Fax: +86-10-59896666

Technical Support: globalsupport@supermap.com

Sales: request@supermap.com

Website: http://www.supermap.com

Your advice and suggestions are welcome!

Content

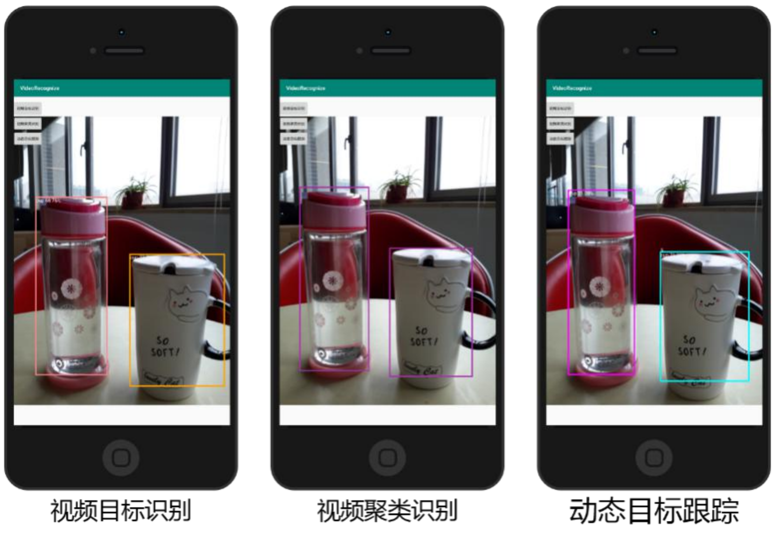
1. Overview

Artificial Intelligence (AI) is a new technology which is intelligence demonstrated by machines through simulating humans' intelligence. Applications of AI includes a lot of fields: speech recognition, image recognition, expert systems, robots, and so on.

SuperMap iMobile 10i (hereinafter referred to as iMobile) introduces the feature video recognition which can recognize targets and license plates on a video.

2. Target recognition

The feature of target recognition can recognize targets, trace targets, and detect clusters. It also can identify the specified element based on the trained model.



2.1. Required codes

（1）. Add class libraries

Add jar packages including com.supermap.ai.jar, com.supermap.data.jar, okhttp-3.9.0.jar, and okio-1.13.0.jar and so libraries including libimb2d_v1000.so, libeasypusher.so , libhyperlpr.so , libopencv_java3.so , libproffmpeg.so , libtensorflow_demo.so, libtensorflow_inference.so, libtensorflowlite_jni.so, libTxtOverlay.so, libUtils.so, libVideoCodecer.so, and libx264enc.so in the folder libs.

（2）. Add the following script

In the file build.gradle of your project add the following codes:

```
aaptOptions {
      noCompress "tflite"
      noCompress "lite"
}
```

（3）. Add the training file

You are allowed to add your own trained model in two ways.

- 1: Copy and paste your file in the directory ..\app\src\main\assets. The model files included in iMobile (detect.tflite and labelmap.txt) are located in the path /SampleCode\Android Studio\AndroidStudioSampleCode\AIdemo\ src\main\assets.
- 2: Copy and paste your model files into your mobile device.

AIDetectViewInfo. fileType controls how to load the model file, while AIDetectViewInfo. modeFile and AIDetectViewInfo. lableFile are responsible for the corresponding file.

(4). Add the control AIDetectView

```
<com.supermap.ai.AIDetectView
        android:id="@+id/test_arcontrol"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:visibility="visible"/>
```

2.2. Target recognition

The feature of target recognition can identify which are targets on a video.

Basic steps:

① Parameter setting: set all required parameters with AIDetectViewInfo.

② Initialization: perform the initialization and settings using the class AIDetectView.

③ Listener settings: add a listener through the funtion setDetectedListener.

Reference codes:

```
//①Sets the initialization parameters
AIDetectViewInfo aidetectViewInfo = new AIDetectViewInfo();
aidetectViewInfo.assetManager = getAssets();     //assetManager applies to save model files
aidetectViewInfo.fileType = AIDetectViewInfo.FileType.ASSETS_FILE;//The file type
aidetectViewInfo.modeFile = "detect.tflite";     //The path of the model file
aidetectViewInfo.lableFile = "labelmap.txt";   //The path of the label file
aidetectViewInfo.inputSize = 300;                 //the size of input
aidetectViewInfo.isQUANTIZED = true;             //Whether to quantify your model
//②Initializes
m_AIdetectView = (AIDetectView) findViewById(R.id.test_arcontrol);
m_AIdetectView.setDetectInfo(aidetectViewInfo); //Sets parameters
m_AIdetectView.init();                           //Initializes
m_AIdetectView.setDetectArrayToUse(m_vecStringUseTypes); //Sets the classification array
m_AIdetectView.setDetectInterval(0);             //Sets the detection time interval
//③Sets the detection result listener
m_AIdetectView.setDetectedListener(new AIDetectView.DetectListener() {
    @Override
    public void onDectetComplete(Map<String, Integer> map) {//The flow statistics
  listener
    }
    @Override
    public void onProcessDetectResult(List<AIRecognition> list) {//The image
recognition result listener
    }
    @Override
    public void onTrackedCountChanged(int i) {   //Tracked count result listener
    }
    @Override
    public void onAISizeChanged(AISize aiSize) {
    }
});
```

2.3. Cluster recognition

Based on the target recognition, you can use the function setisPolymerize() of the class AIDetectView to recognize clusters on a video.

Reference code:

```
if (m_AIdetectView.isPolymerize())    //Whether it is the cluster mode
{
    m_AIdetectView.setisPolymerize(false);   //Sets to the non-cluster mode
}
else
{
    m_AIdetectView.setisPolymerize(true);        //Sets to the cluster mode
    m_AIdetectView.setPolymerizeThreshold(400,400); //Sets the threshold of the cluster
mode
}
```

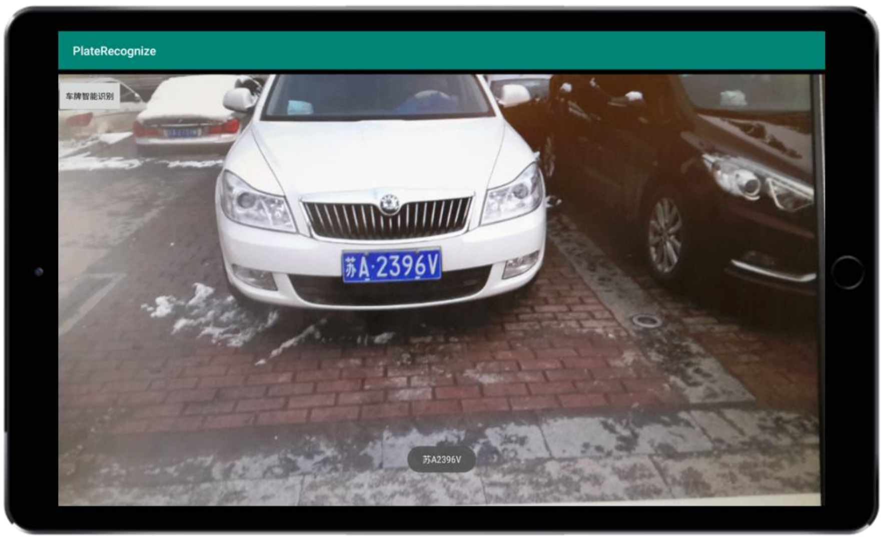
2.4. Dynamic target tracing

Based on the target recognition, you can use the functions startCountTrackedObjs() and stopCountTrackedObjs() of the class AIDetectView to control whether to trace targets.

```
m_AIdetectView.startCountTrackedObjs();     //Begins to trace and count
m_AIdetectView.stopCountTrackedObjs();      //Stops tracing and counting
```

3. Intelligent license plate recognition

License plate recognition is one kind of applications in the video recognition which can extract license plate information from complex backgrounds. Combining with other technologies including pre-processing images, feature extraction, character identification, and so on, this feature can identify the license plate and its colors of a car.



3.1. Required codes

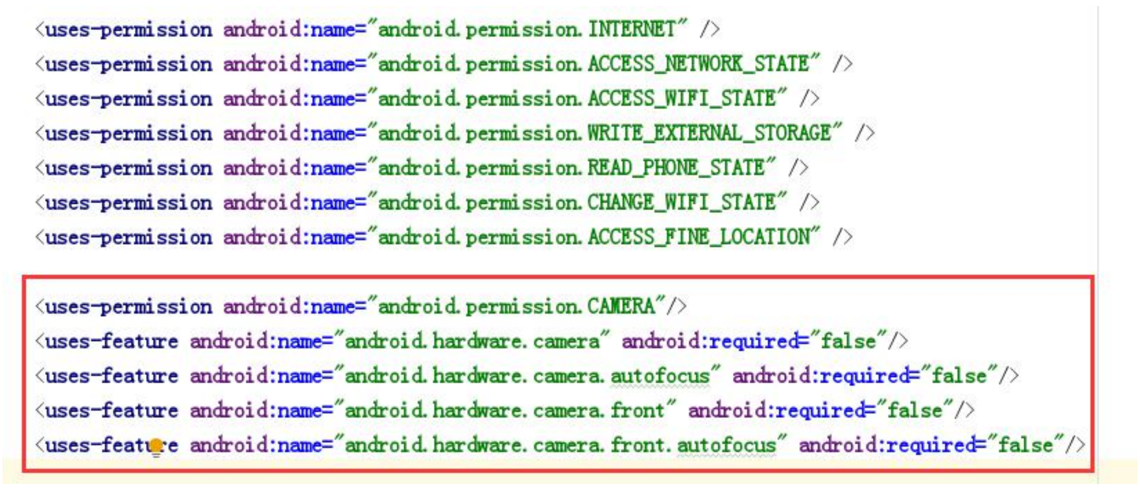
(1). Add class libraries

Add jar packages including com.supermap.ai.jar, com.supermap.data.jar, okhttp-3.9.0.jar, and okio-1.13.0.jar and so libraries including libimb2d_v1000.so, libeasypusher.so , libhyperlpr.so , libopencv_java3.so , libproffmpeg.so , libtensorflow_demo.so, libtensorflow_inference.so, libtensorflowlite_jni.so, libTxtOverlay.so, libUtils.so, libVideoCodecer.so, and libx264enc.so in the folder libs.

(2). Grant permissions

Apart from the permissions required by iMobile, the following permissions should be granted:

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera"
android:required="false"/>
<uses-feature android:name="android.hardware.camera.autofocus"
android:required="false"/>
<uses-feature android:name="android.hardware.camera.front"
android:required="false"/>
<uses-feature android:name="android.hardware.camera.front.autofocus"
android:required="false"/>
```

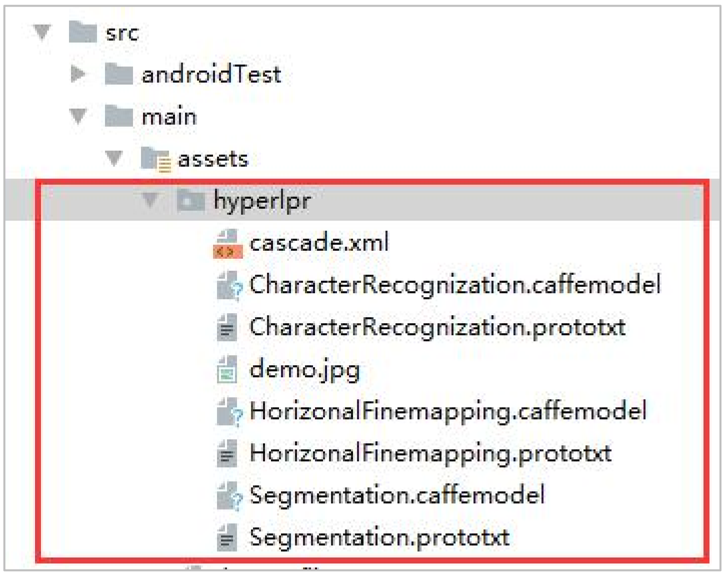


```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front.autofocus" android:required="false"/>
```

（3）． Add recognition files

Copy and paste the folder hyperlpr and all files located inside it to the project directory ..\app\src\main\assets. The folder hyperlpr is located in the path SampleCode\SampleCode_AndroidStudio\AndroidStudioSampleCode\hyperlpr\src\main\assets.



```
src
  androidTest
  main
    assets
      hyperlpr
        cascade.xml
        CharacterRecognization.caffemodel
        CharacterRecognization.prototxt
        demo.jpg
        HorizonalFinemapping.caffemodel
        HorizonalFinemapping.prototxt
        Segmentation.caffemodel
        Segmentation.prototxt
```

（4）． Add the control JavaCameraView

```
<org.opencv.android.JavaCameraView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/javaCameraView"
    android:visibility="invisible"/>
```

3.2. License plate recognition

Reference codes:

```
new MyAsyncTask(this).execute();
if( OpenCVLoader.initDebug() ) {
        System.loadLibrary("opencv_java3");
        System.loadLibrary("hyperlpr");
        m_handle = DeepAssetUtil.initRecognizer(MainActivity.this);
}else{
}
m_OpenCvCameraView = (CameraBridgeViewBase)
findViewById(R.id.javaCameraView);
m_OpenCvCameraView.setVisibility(CameraBridgeViewBase.VISIBLE);
m_OpenCvCameraView.setCvCameraViewListener(new
CameraBridgeViewBase.CvCameraViewListener() {
        @Override
        public void onCameraViewStarted(int i, int i1) {
        }
        @Override
        public void onCameraViewStopped() {
        }
        @Override
        public Mat onCameraFrame(Mat mat) {
            m_Rgba = mat;
            new MyAsyncTask(MainActivity.this).execute();
            return m_Rgba;
        }
});
m_OpenCvCameraView.enableView();
m_Rgba = new Mat(m_OpenCvCameraView.getHeight(),
m_OpenCvCameraView.getWidth(), CvType.CV_8UC4);

// Avoids memory leaks
  @SuppressLint("StaticFieldLeak")
  private static class MyAsyncTask extends AsyncTask<String, Integer, String> {
        private final WeakReference<MainActivity> weakActivity;
        MyAsyncTask(MainActivity myActivity) {
            this.weakActivity = new WeakReference<>(myActivity);
        }
        @Override
        protected String doInBackground(String... params) {
            try {
                return   DeepCarUtil.SimpleRecognization(m_Rgba.getNativeObjAddr(),
m_handle);
```

```
            } catch (Exception e) {
                return null;
            }
        }
        @Override
        protected void onPostExecute(String license) {//license is the result of license plate
recognition
            super.onPostExecute(license);
            MainActivity activity = weakActivity.get();
            if (activity == null
                    || activity.isFinishing()
                    || activity.isDestroyed()) {
                return;
            }
            Toast.makeText(activity,license ,Toast.LENGTH_SHORT).show();
        }
 }
```