**Project 3 - Using TensorFlow to help fruit grading**

**Name: Yu Wang**

**Section 0： YouTube URL**

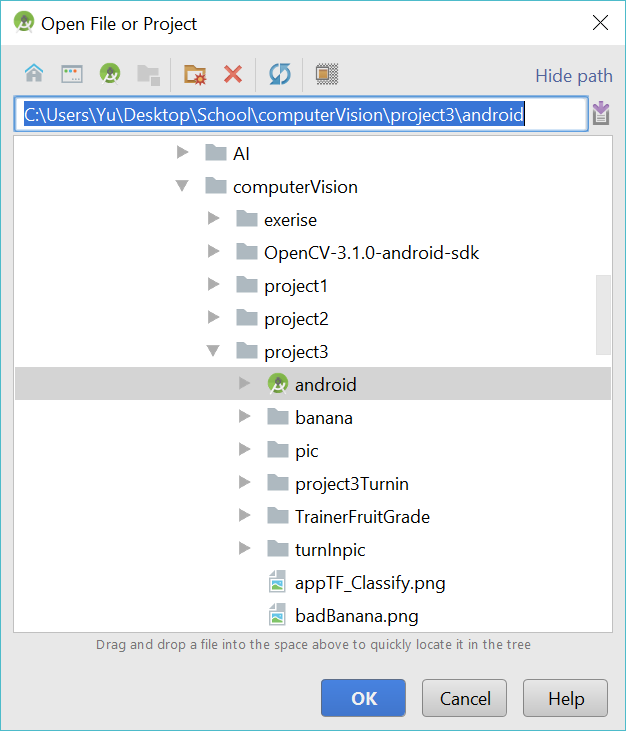
[**https://www.youtube.com/watch?v=bq43vdvVWnc**](https://www.youtube.com/watch?v=bq43vdvVWnc)

[**https://www.youtube.com/watch?v=bClPtZGUh-s**](https://www.youtube.com/watch?v=bClPtZGUh-s)

**Section 1 Execution Instructions:**

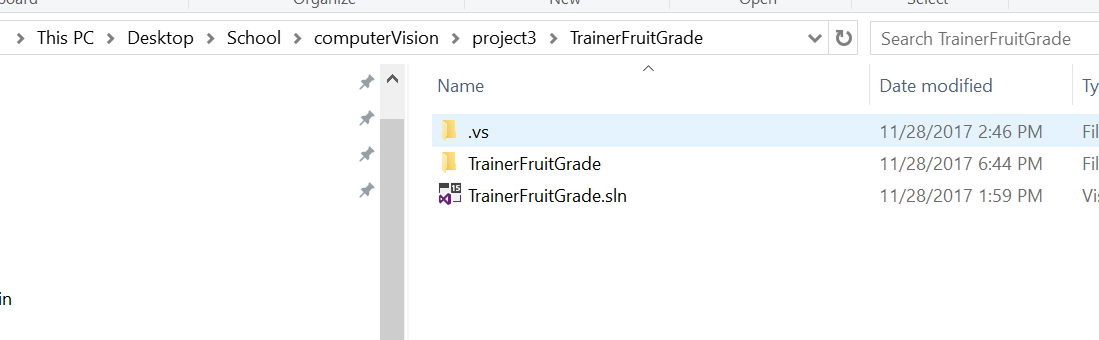
*Clone or download the files at GitHub.*

*Use Android Stuido: file-->open-->*

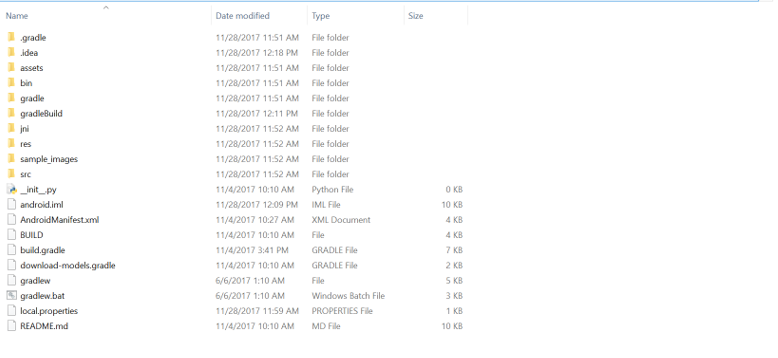


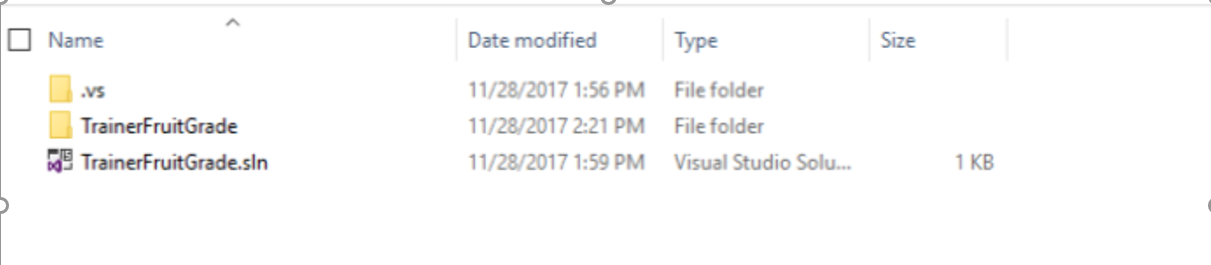
*If it does not work, just need to Rebuild project.*

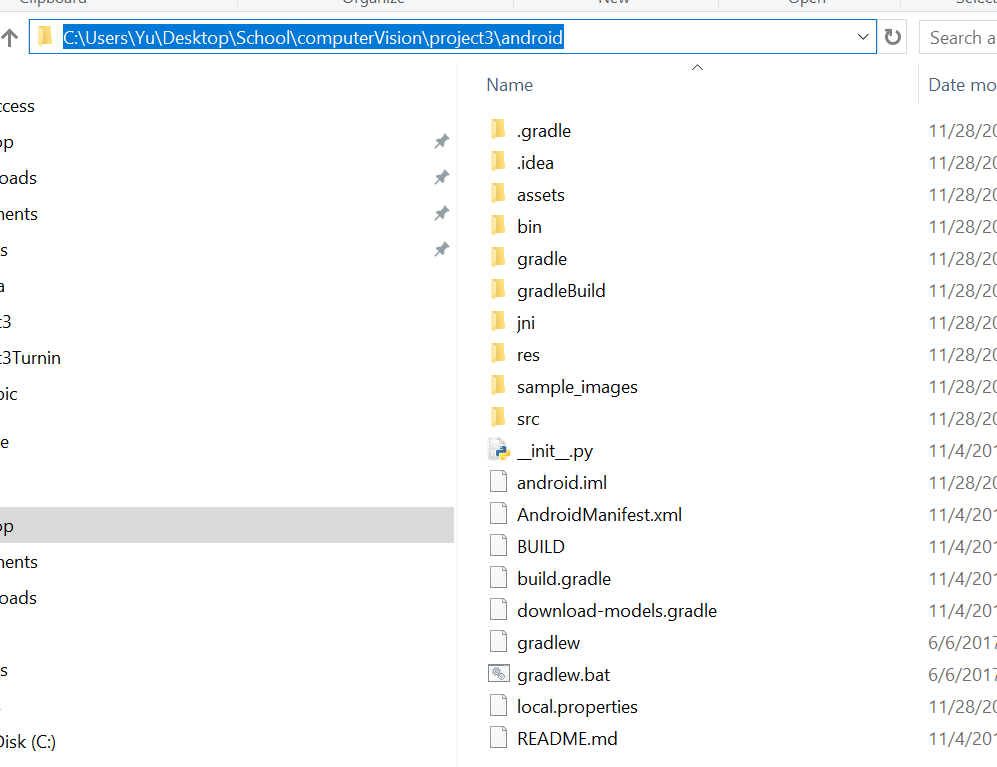
*Use visual studio to open the sln file*

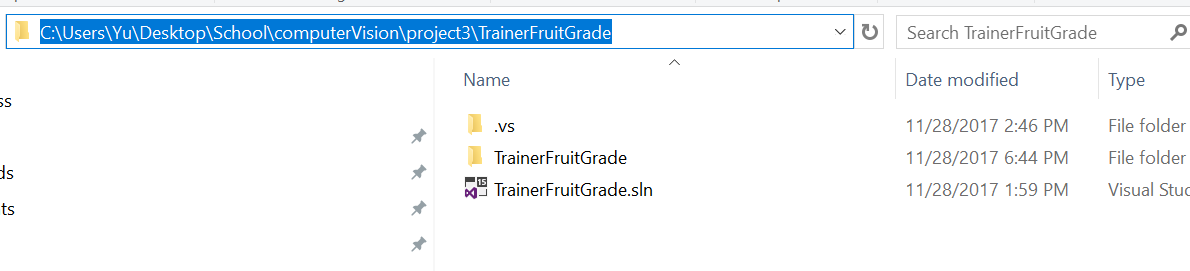


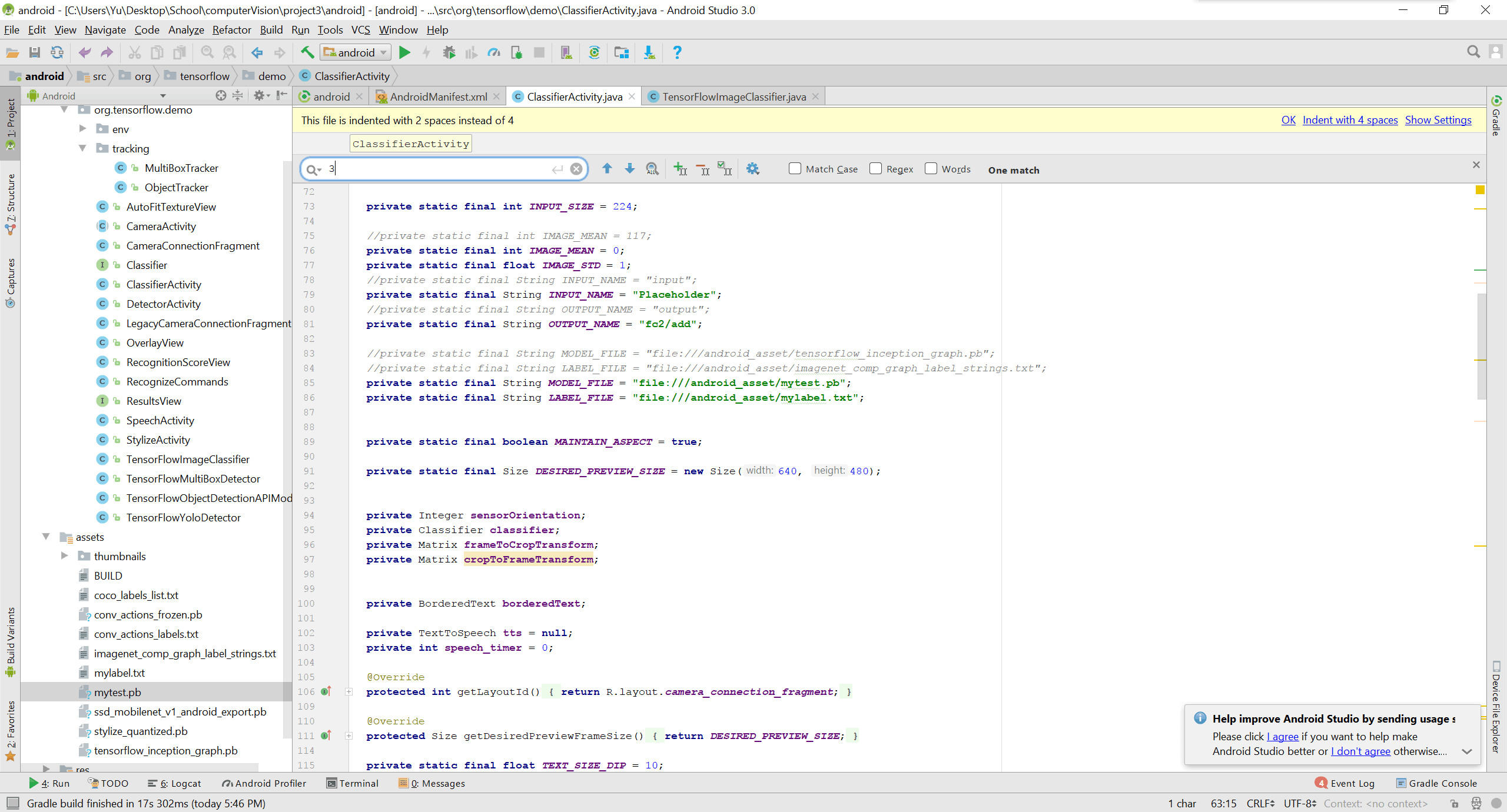
*The project contains two parts: the first part is the Android App Project with the fruit grading App, the second part is the Python Project with the TensorFlow model training.*







  
**screenshot 1.2 = 1. Directory of the Android project 3. Directory of the Python project 3**



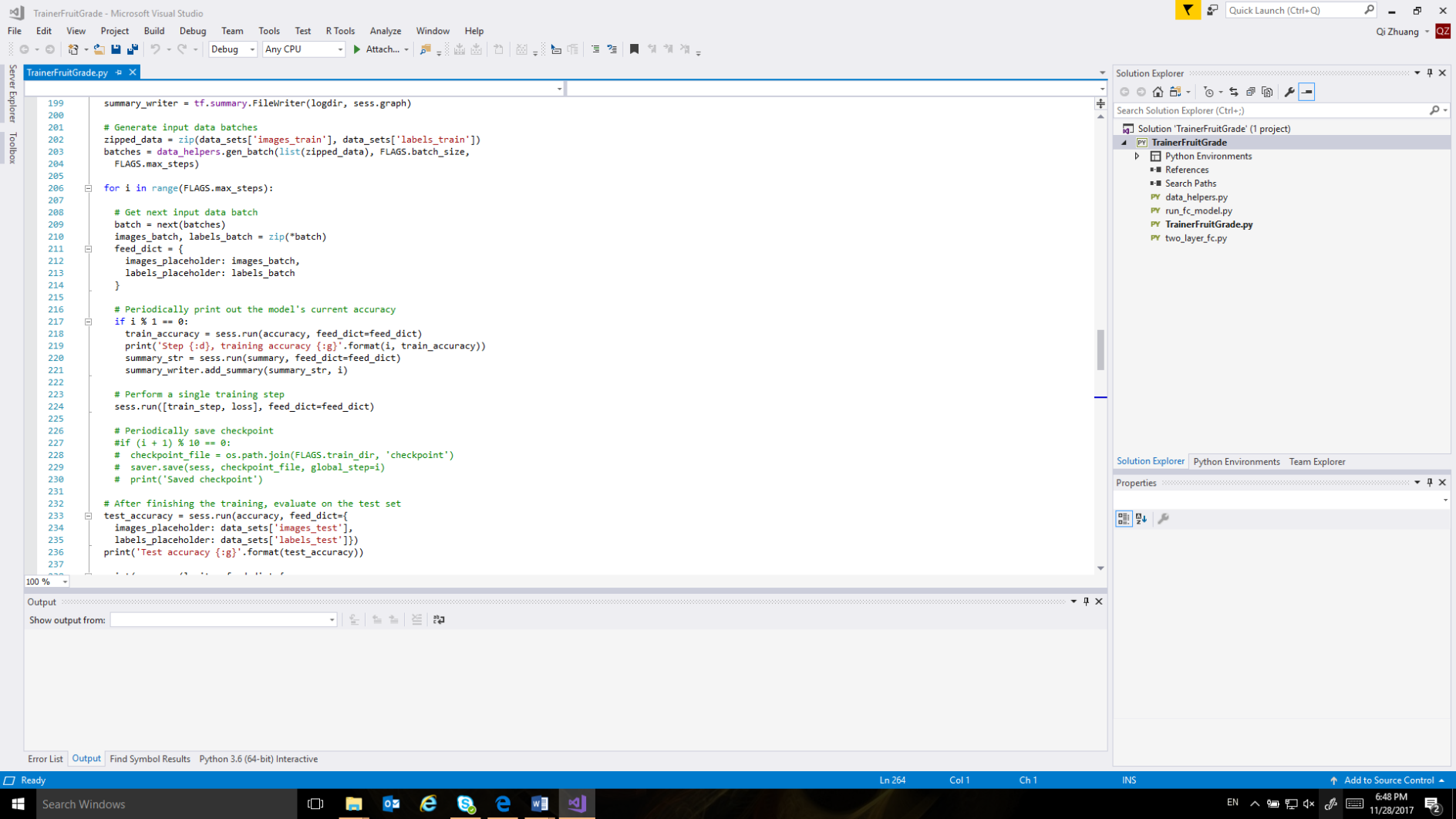


FIGURE HERE  
**screenshot 1.3 = 1. Opened Android project 3. Opened Python Project**

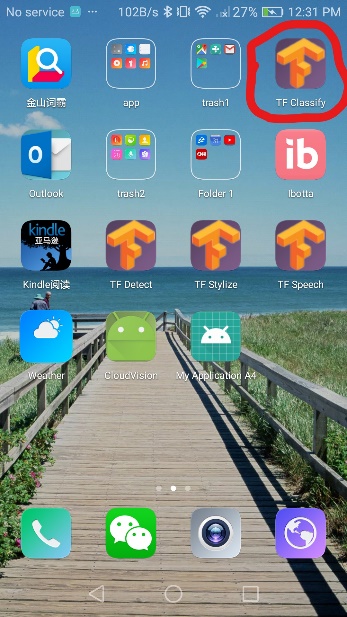
****

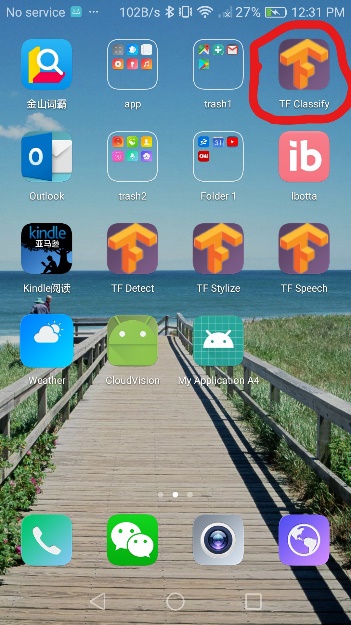
FIGURE HERE  
**screenshot 1.4 = 1. Running of the test on Android phone. 2. Training of the data.**

**Section 2 Code Description**

*A describing how code is structured and the state of how it works. Give a description for each filename listed.*

* In Android Project: The whole project is based on Google’s TensorFlow Android example. The main source files are ClassifierActivity.java and TensorFlowImageClassifier.java; the main model files are mytest.pb and mylabel.txt.
  + ClassifierActivity.java: Activity class. Initialized the tensorflow utility. Communicate with tensorflow utility. Handle result with TTS engine.
  + TensorFlowImageClassifier.java: the tensorflow utility. Communicate with the real tensorflow library. Prepare data and get result from the tensorflow library.
  + mytest.pb: model file used for tensorflow library.
  + mylabel.txt: dictionary file to translate number to the label.
* In Python Project: The training part is based on the TensorFlow example MNIST at <https://www.tensorflow.org/get_started/mnist/pros> . The banana images are from <http://image-net.org/> . There are two main source files in the project: TrainerFruitGrade.py and two\_layer\_fc.py.
  + TrainerFruitGrade.py: load images, build the network, save models.
  + Two\_layer\_fc.py: implementation of the layers for the neural network.

**Section 3 Testing**:

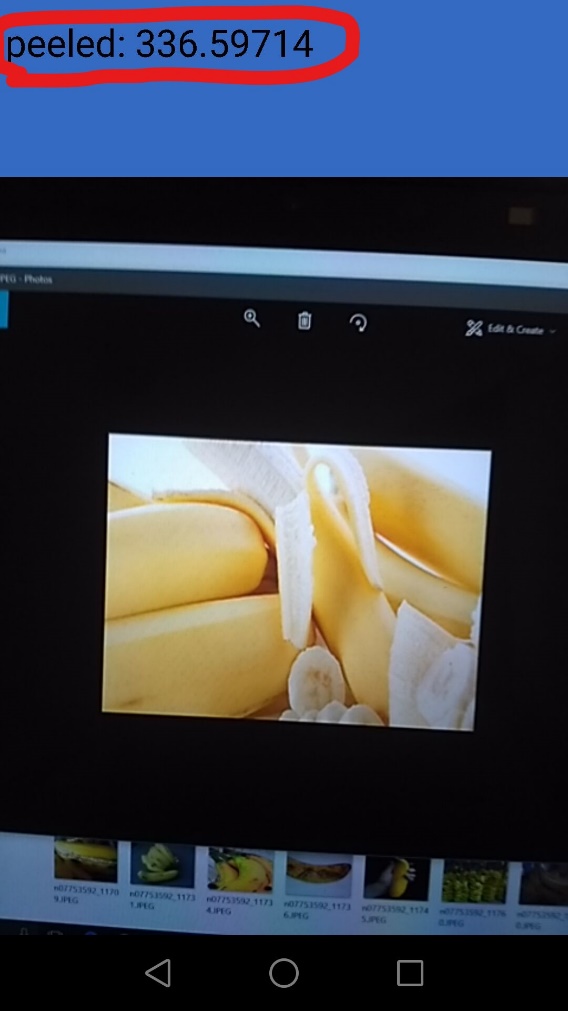
**section 3.1: starting application**  
  
  
screenshot 3.1a= showing icon and resulting starting GUI.

**section 3.2:**

****  
screenshot 3.2a = screen shot of a correct recognition of raw bananas. The current criteria is to leave the labels with top 3 high scores, and only keep positive scores. The TTS engine will speak the top score’s label.

****  
screenshot 3.2b = screen shot of showing recognition of bananas in good condition. Sometimes when the phone is shaking, the result changes a little.

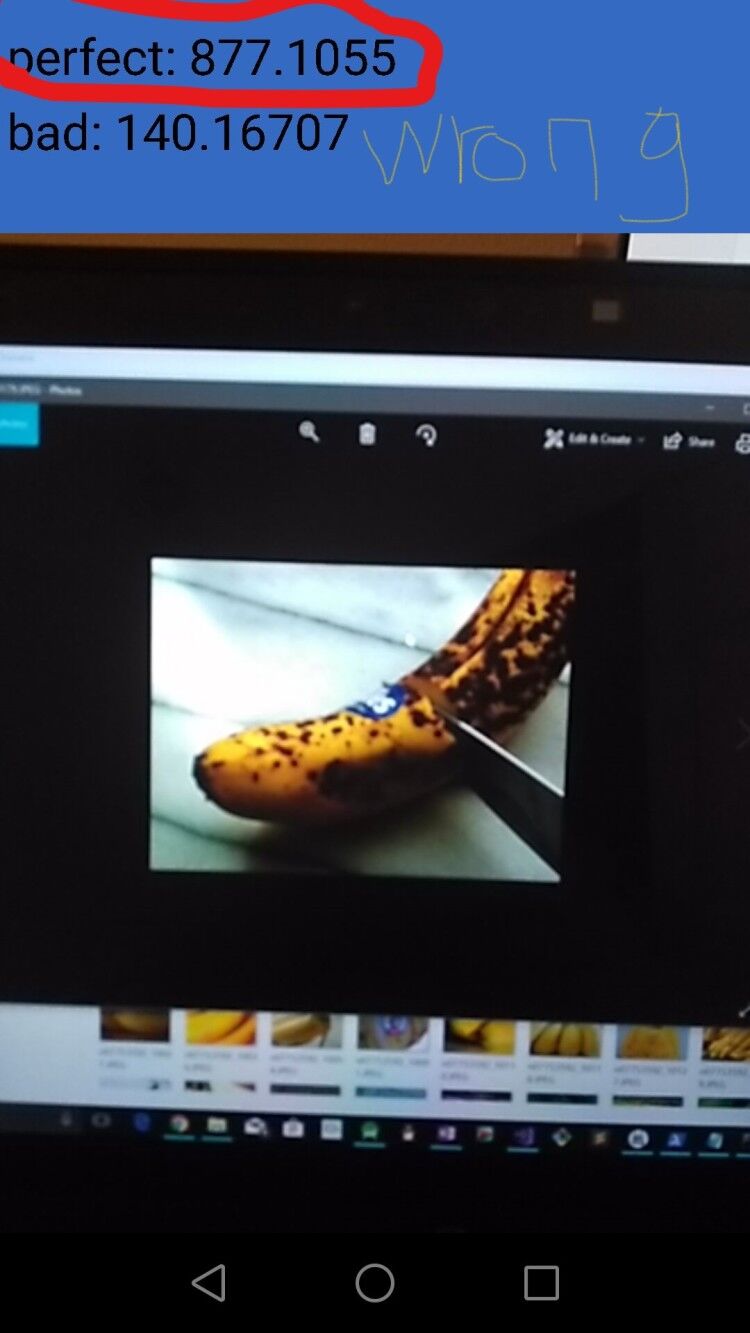
****  
screenshot 3.2c screen shot of a bad banana.



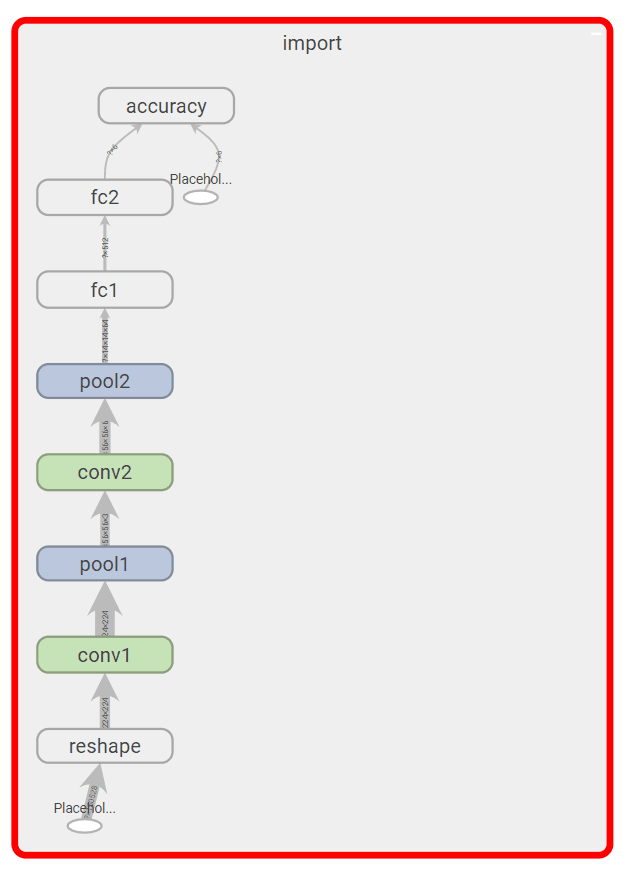
screenshot 3.2d screen shot of a peeled banana



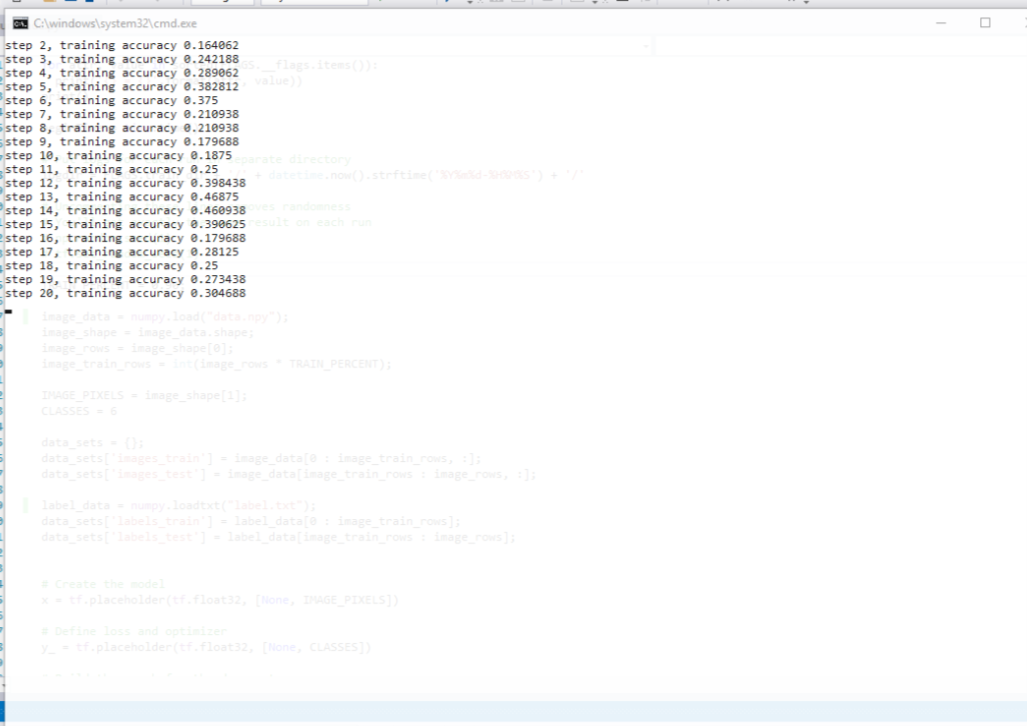
screenshot 3.2e screenshot of perfect and a little bad banana.



screenshot 3.2f sometimes the app will have wrong results.



**section 3.3: The graph for the trained model by tensorboard**



**section 3.4: The process of training**