**RETRAINING MOBILENET TO DETECT SYNTHETICALLY GENERATED GAUGE IMAGES USING TENSORFLOW:**

Reference: <https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0>

1. git clone <https://github.com/googlecodelabs/tensorflow-for-poets-2>

cd tensorflow-for-poets-2

**2. Set IMAGE\_SIZE and ARCHITECTURE variables.**

IMAGE\_SIZE=224

ARCHITECTURE="mobilenet\_0.50\_${IMAGE\_SIZE}"

**3. Run the retraining script with path to the gauge images folders**

python -m scripts.retrain \

--bottleneck\_dir=tf\_files/bottlenecks \

--how\_many\_training\_steps=5000 \

--model\_dir=tf\_files/models/ \

--summaries\_dir=tf\_files/training\_summaries/"${ARCHITECTURE}" \

--output\_graph=tf\_files/retrained\_graph.pb \

--output\_labels=tf\_files/retrained\_labels.txt \

--architecture="${ARCHITECTURE}" \

--image\_dir=tf\_files/tf\_gauges

**Note- tf\_gauges folder contains subfolders with gauge images for every class. Each sub-folder is named after one of the categories and contains only images from that category. The classification script uses the sub folder names as label names, and the images inside each folder should be pictures that correspond to that label.**

python -m scripts.label\_image \

--graph=tf\_files/retrained\_graph.pb \

--image=tf\_files/flower\_photos/daisy/21652746\_cc379e0eea\_m.jpg

**PORTING THE TRAINED MODEL TO ANDROID PHONE FOR REAL TIME GAUGE READING DETECTION:**

Reference: <https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0>

1. **Optimize the model using the following script.**

python -m tensorflow.python.tools.optimize\_for\_inference \

--input=tf\_files/retrained\_graph.pb \

--output=tf\_files/optimized\_graph.pb \

--input\_names="input" \

--output\_names="final\_result"

Note: Running this script creates a new file at tf\_files/optimized\_graph.pb.

**2. Add your model files to the project**

**The demo project is configured to search for a graph.pb, and a labels.txt files in the android/tfmobile/assets directory. Replace those two files with your versions.**

**The following command accomplishes this task:**

cp tf\_files/rounded\_graph.pb android/tfmobile/assets/graph.pb

cp tf\_files/retrained\_labels.txt android/tfmobile/assets/labels.txt

**3. Change the output\_name in ClassifierActivity.java**

**The TensorFlow Interface used by the app requires that you ask for your results by name. The app is currently set up to read the output of the baseline MobileNet, named "MobilenetV1/Predictions/Softmax". The output node for our model has a different name: "final\_result". Open ClassifierActivity.java and update the OUTPUT\_NAME variable as follows:**

ClassifierActivity.java

private static final String INPUT\_NAME = "input";

private static final String OUTPUT\_NAME = "final\_result";

**4. Run the android app to check real time gauge reading.**

Additional reference: <https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2-tflite/#0>

**RETRAINING OBJECT DETECTION MODELS TO DETECT SYNTHETICALLY GENERATED GAUGE IMAGES USING TENSORFLOW:**

1. Setting up the environment for tensorflow object detection API

Reference:

<https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/installation.md>

2. <https://towardsdatascience.com/detecting-pikachu-on-android-using-tensorflow-object-detection-15464c7a60cd>

3. <https://github.com/datitran/raccoon_dataset>