nimble

ML-powered Facial Recognition Scanner V2

Tobias

Growth Session #21 - January 24-25 2019

Services

Facial Recognition Scanner

- **Raspberry Pi**
  - Small computer
- **Android Things**
  - IoT operating system
- **Firebase MLKit**
  - ML SDK for mobile platform
- **Tensorflow**
  - Machine learning framework

Services

Facial Recognition Scanner

- **~~Raspberry Pi~~**
  - ~~Small computer~~
- **~~Android Things~~**
  - ~~IoT operating system~~
- **~~Firebase MLKit~~**
  - ~~ML SDK for mobile platform~~
- **Tensorflow**
  - Machine learning framework

ML Firebase VS Tensorflow

ML Kit for Firebase

PRE-MADE MODELS

Text recognition

Image labeling

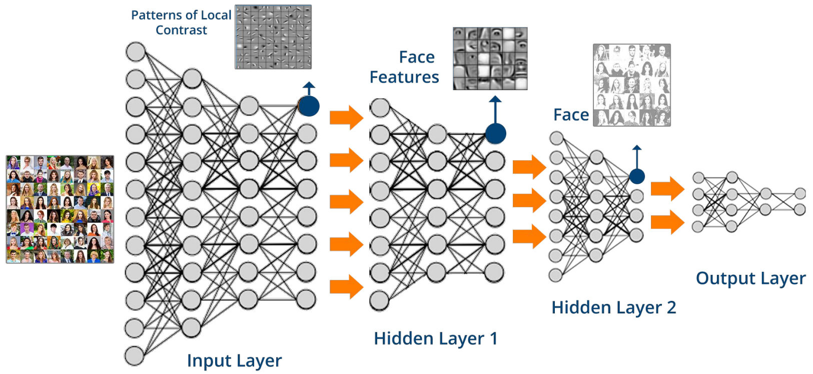
Barcode scanning

Face detection

Landmark recognition

TensorFlow

CUSTOM MODELS



ML Firebase VS Tensorflow

ML Kit
for Firebase

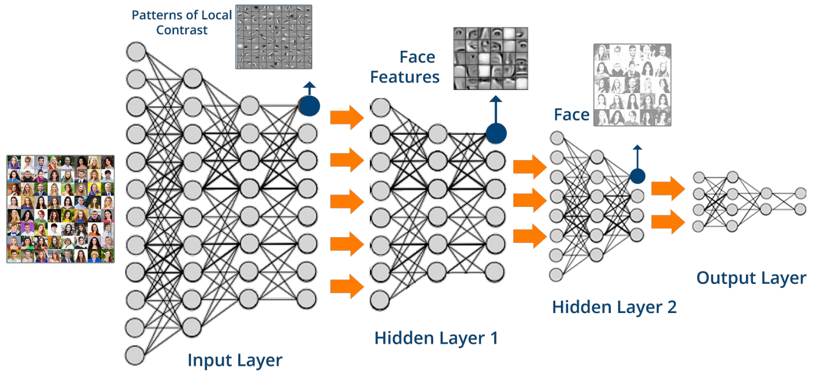
PRE-MADE MODELS

Text recognition · Image labeling · Barcode scanning · Face detection · Landmark recognition

FACE RECOGNITION

TensorFlow

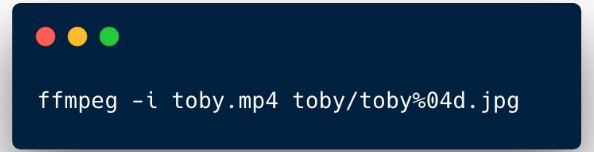
CUSTOM MODELS



Steps, how does it work?

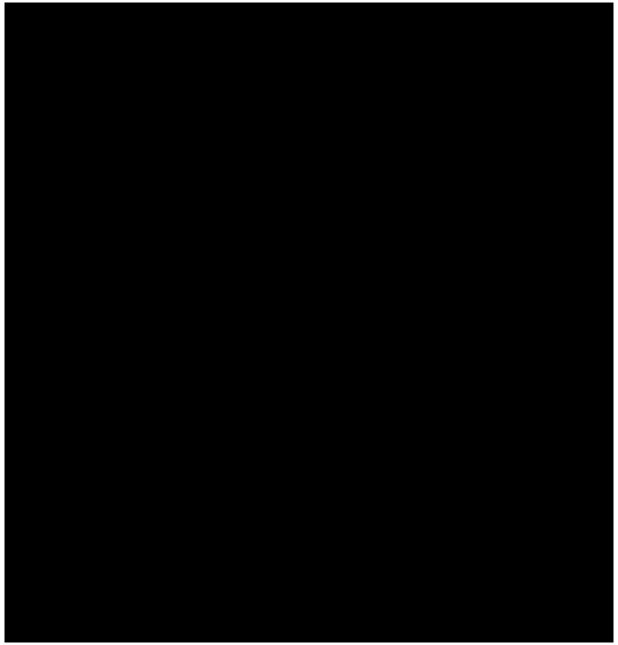
1. Create a model based on a **video**

2. Slice the video up in to multiple images



```
ffmpeg -i toby.mp4 toby/toby%04d.jpg
```

3. Continue doing this for all other models



Steps, how does it work?

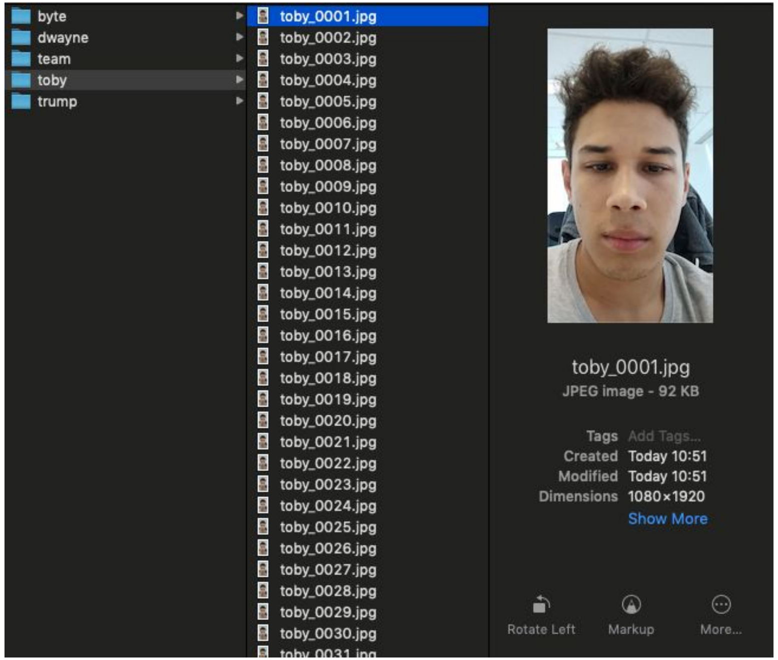
1. Create models based on **videos**

2. Slice the video up in to multiple images

```
ffmpeg -i toby.mp4 toby/toby%04d.jpg
```

3. Continue doing this for all other models

4. Now we have our **INPUTS**



byte
dwayne
team
toby
trump

toby_0001.jpg
toby_0002.jpg
toby_0003.jpg
toby_0004.jpg
toby_0005.jpg
toby_0006.jpg
toby_0007.jpg
toby_0008.jpg
toby_0009.jpg
toby_0010.jpg
toby_0011.jpg
toby_0012.jpg
toby_0013.jpg
toby_0014.jpg
toby_0015.jpg
toby_0016.jpg
toby_0017.jpg
toby_0018.jpg
toby_0019.jpg
toby_0020.jpg
toby_0021.jpg
toby_0022.jpg
toby_0023.jpg
toby_0024.jpg
toby_0025.jpg
toby_0026.jpg
toby_0027.jpg
toby_0028.jpg
toby_0029.jpg
toby_0030.jpg

toby_0001.jpg
JPEG image - 92 KB

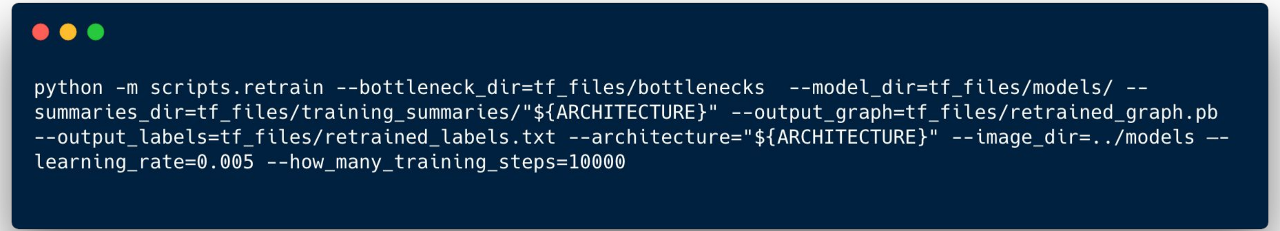
Tags	Add Tags...
Created	Today 10:51
Modified	Today 10:51
Dimensions	1080×1920

Show More

Steps, how does it work?

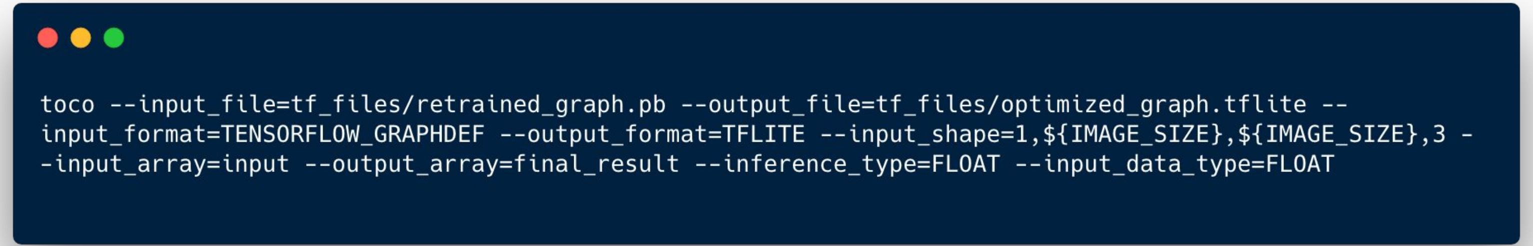
5. Re-train an existing pre-trained-model (= Mobilenet)

   → Used for classification, detection, …



```
python -m scripts.retrain --bottleneck_dir=tf_files/bottlenecks  --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=../models —-
learning_rate=0.005 --how_many_training_steps=10000
```

6. Export the **OUTPUT** to a more mobile-friendly format (= Optimization)



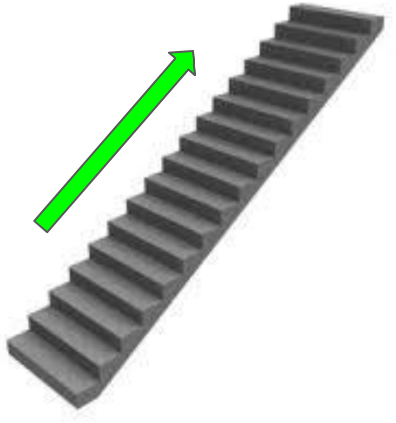
```
toco --input_file=tf_files/retrained_graph.pb --output_file=tf_files/optimized_graph.tflite --
input_format=TENSORFLOW_GRAPHDEF --output_format=TFLITE --input_shape=1,${IMAGE_SIZE},${IMAGE_SIZE},3 -
-input_array=input --output_array=final_result --inference_type=FLOAT --input_data_type=FLOAT
```

DEMO

Next Steps

What could I have done to improve this?

- Integrate with **Firebase**: Having the models stored in the cloud VS locally
- Combine with **Face-detection** instead scanning the whole image
- Create more **models** = more **probabilities** = more **accuracy**



Thanks!

Contact Nimble

nimblehq.co
hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit #2402-03, Klong Toei, Wattana, Bangkok 10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

nimble