

TensorFlow for Poets

petewarden@google.com



Want to classify images, but don't have a PhD in machine learning?

This talk will show you how!

Don't worry about trying to write it all down, there's a full tutorial online at

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

Don't believe me?

Swiss trains may run on time, but American ones don't!



Don't believe me?

A non-specialist used this to tell when CalTrains were late, using Raspberry Pi's to spot when one was passing a particular spot. See <https://svds.com/introduction-to-trainspotting/>

Lots more examples, from spotting traffic cops to sorting trash and recycling.

What do you need?

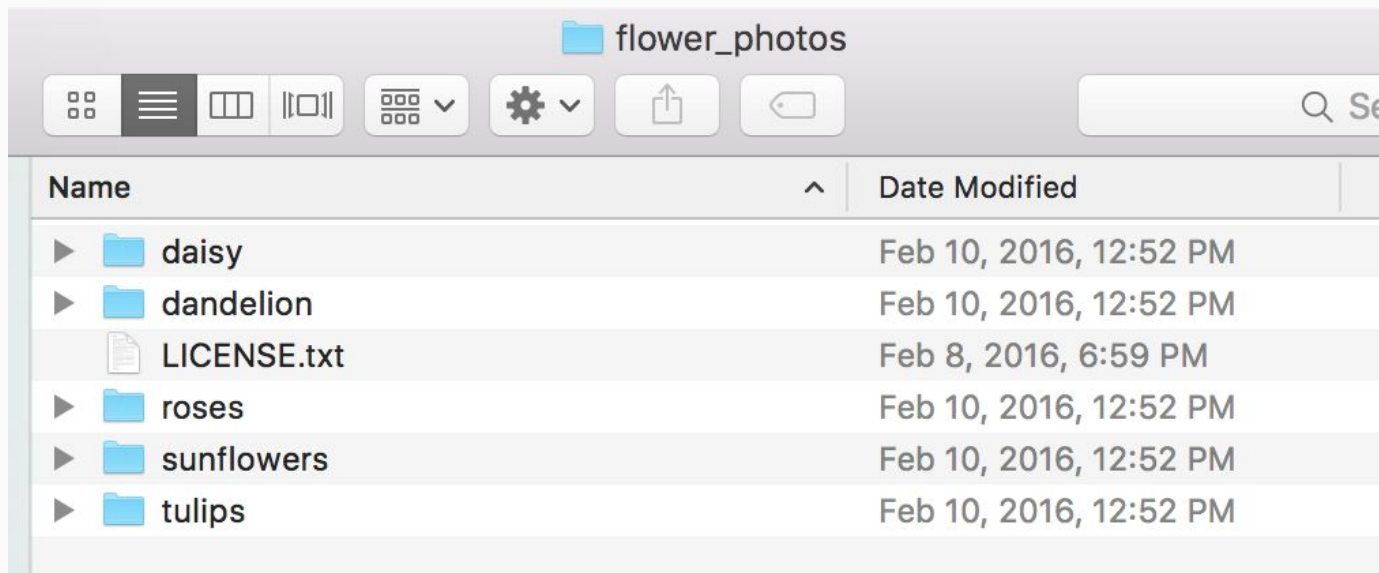
A laptop (OS X, Windows, or Linux).

A few hundred photos of the thing you want to identify, sorted into folders.

One hour.

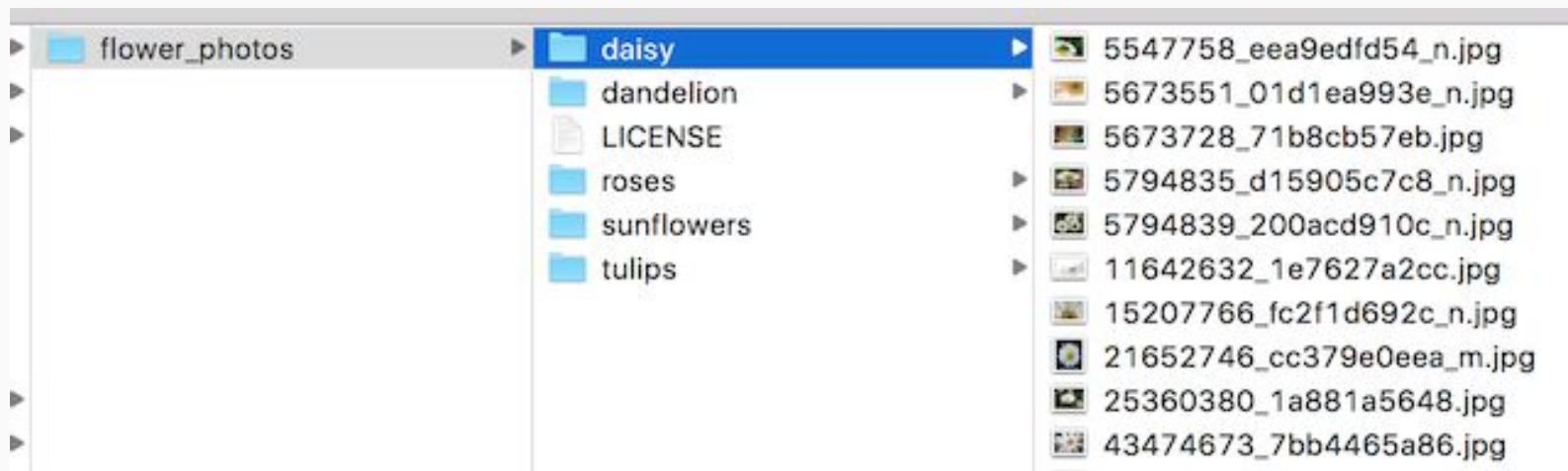
Your photos

Start with the prepackaged flowers data set.



Your photos

Start with the prepackaged flowers data set.



Your photos

Several hundred photos of each type of flower, organized into folders by their species.

You'll need to do the same with your images.

You can start off with fewer, if that's all you have.

Set up Docker

This makes it easy to deal with dependencies.

<https://docs.docker.com/docker-for-mac/>

```
docker run hello-world
```

```
docker run -it gcr.io/tensorflow/tensorflow:latest-devel
```

Share your photos folder with Docker

```
# ctrl-D if you're still in Docker and then:
```

```
cd $HOME
```

```
mkdir tf_files
```

```
cd tf_files
```

```
curl -O http://download.tensorflow.org/example\_images/flower\_photos.tgz
```

```
tar xzf flower_photos.tgz
```

```
docker run -it -v $HOME/tf_files:/tf_files gcr.io/tensorflow/tensorflow:latest-devel
```

Train the model

```
cd /tensorflow
git pull

python tensorflow/examples/image_retraining/retrain.py \
--bottleneck_dir=/tf_files/bottlenecks \
--model_dir=/tf_files/inception \
--output_graph=/tf_files/retrained_graph.pb \
--output_labels=/tf_files/retrained_labels.txt \
--image_dir /tf_files/flower_photos
```

Test the model

```
# ctrl-D to exit Docker and then:
```

```
curl -L https://goo.gl/tx3dqg > $HOME/tf_files/label_image.py
```

```
docker run -it -v $HOME/tf_files:/tf_files gcr.io/tensorflow/tensorflow:latest-devel
```

```
python /tf_files/label_image.py /tf_files/flower_photos/daisy/21652746_cc379e0eea_m.jpg
```

```
daisy (score = 0.99071)
```

```
sunflowers (score = 0.00595)
```

```
dandelion (score = 0.00252)
```

```
roses (score = 0.00049)
```

```
tulips (score = 0.00032)
```

Now use on your own images

--image_dir is the flag to point to your own images

Model is in ~/tf_files/retrained_graph.pb

You can easily load this into the Android, Python, or Raspberry Pi examples.

Advanced techniques

Not enough data? Try the random crop and sizing examples to augment.

Plenty of time and want more accuracy? Fine tune more than just the top layer:
<https://github.com/tensorflow/models/tree/master/inception#how-to-fine-tune-a-pre-trained-model-on-a-new-task>

Plenty of time, machines, and data? Retrain the whole network from scratch:
<https://github.com/tensorflow/models/tree/master/inception#how-to-train-from-scratch-in-a-distributed-setting>

How does it work?

Transfer learning.

Lets us take the knowledge gained by training on 1 million Imagenet photos, and transfer it to categories we care about.

Only a small data set and amount of retraining required.

Common pattern across many deep learning problems.

Great way for non-specialists to benefit from advances in ML.

Questions?

petewarden@google.com

@petewarden