

20161025-tensorflow 的安装、图像识别应用、训练自己的图像识别模型

一、tensorflow 的安装

github 地址:

<https://github.com/tensorflow/tensorflow>

以源码方式安装, 官方文档:

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/get_started/os_setup.md

Installing from sources 这一节:

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/get_started/os_setup.md#installing-from-sources

```
$ cd ~
```

```
$ git clone https://github.com/tensorflow/tensorflow
```

安装 bazel:

<https://www.bazel.io/versions/master/docs/install.html#ubuntu>

1. Install JDK 8

```
$ sudo add-apt-repository ppa:webupd8team/java
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install oracle-java8-installer
```

2. Add Bazel distribution URI as a package source

```
$ echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list
```

```
$ curl https://bazel.io/bazel-release.pub.gpg | sudo apt-key add -
```

3. Update and install Bazel

```
$ sudo apt-get update && sudo apt-get install bazel
```

如果下载失败需要用迅雷下载安装包 (目前的是 `bazel_0.3.2_amd64.deb`) 手动安装即可。

安装依赖项:

```
$ sudo apt-get install python-numpy python-dev python-wheel
```

安装 CUDA:

(没有显卡, 没装这个)

配置安装:

```
$ ./configure
```

google cloud 和 hadoop 都选 N, 其他默认即可。

Create the pip package and install:

```
$ bazel build -c opt //tensorflow/tools/pip_package:build_pip_package
```

(如果有 GPU 的话则用下面这个命令:

```
$ bazel build -c opt --config=cuda //tensorflow/tools/pip_package:build_pip_package )
```

```
$ bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

这个安装文件的具体名字要到 `/tmp/tensorflow_pkg` 中查看, 用实际的更换:

```
$ sudo pip install /tmp/tensorflow_pkg/tensorflow-0.11.0rc1-py2-none-any.whl
```

Train your first TensorFlow neural net model

```
$ cd tensorflow/models/image/mnist
```

```
$ python convolutional.py
```

正常情况下即可看到开始进行训练了：

```
lifei@lifei-desktop: ~/tensorflow/tensorflow/models/image/mnist
Minibatch loss: 2.502, learning rate: 0.009025
Minibatch error: 0.0%
Validation error: 1.1%
Step 2500 (epoch 2.91), 135.8 ms
Minibatch loss: 2.483, learning rate: 0.009025
Minibatch error: 0.0%
Validation error: 1.1%
Step 2600 (epoch 3.03), 145.3 ms
Minibatch loss: 2.453, learning rate: 0.008574
Minibatch error: 0.0%
Validation error: 1.2%
Step 2700 (epoch 3.14), 139.4 ms
Minibatch loss: 2.481, learning rate: 0.008574
Minibatch error: 1.6%
Validation error: 1.1%
Step 2800 (epoch 3.26), 135.5 ms
Minibatch loss: 2.414, learning rate: 0.008574
Minibatch error: 0.0%
Validation error: 1.1%
Step 2900 (epoch 3.37), 135.3 ms
Minibatch loss: 2.470, learning rate: 0.008574
Minibatch error: 3.1%
Validation error: 1.2%
```

二、tensorflow 在图像识别中的应用

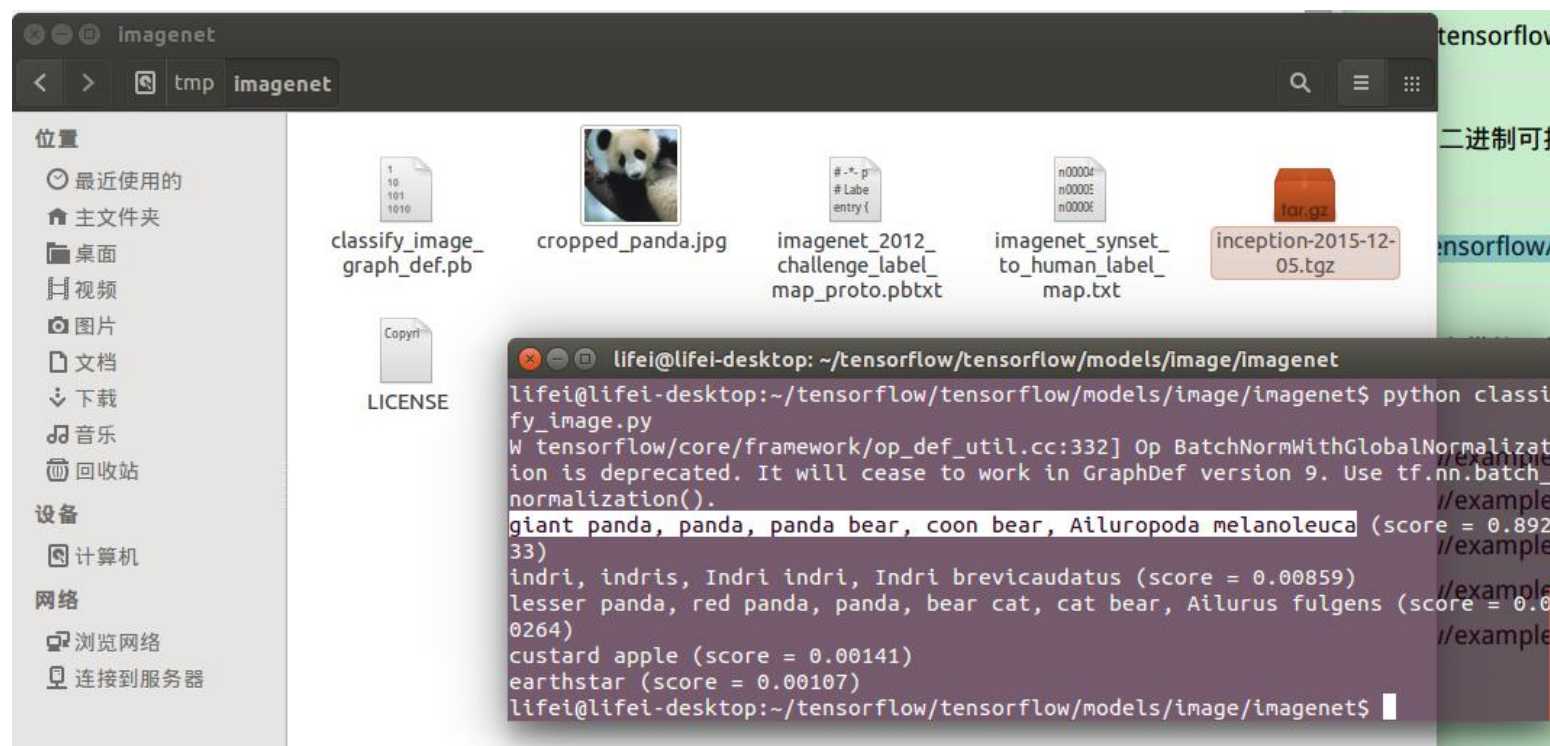
1. python API 的用法

```
$ cd ~/tensorflow/tensorflow/models/image/imagenet
```

```
$ python classify_image.py
```

该脚本会自动下载压缩文件 inception-2015-12-05.tgz 到/tmp/imagenet 文件夹中并解压，同样的，很有可能下载失败，因此需要在迅雷中手动下载然后放置到该文件夹中，再次运行脚本即可。

该压缩文件中包含分类模型、分类标签和一个张大熊猫的照片，并对该照片进行识别：



2. C++ API 的用法

下载数据集:

```
$ cd ~/tensorflow
```

```
$ wget https://storage.googleapis.com/download.tensorflow.org/models/inception_dec_2015.zip -O tensorflow/examples/label_image/data/inception_dec_2015.zip
```

```
$ unzip tensorflow/examples/label_image/data/inception_dec_2015.zip -d tensorflow/examples/label_image/data/
```

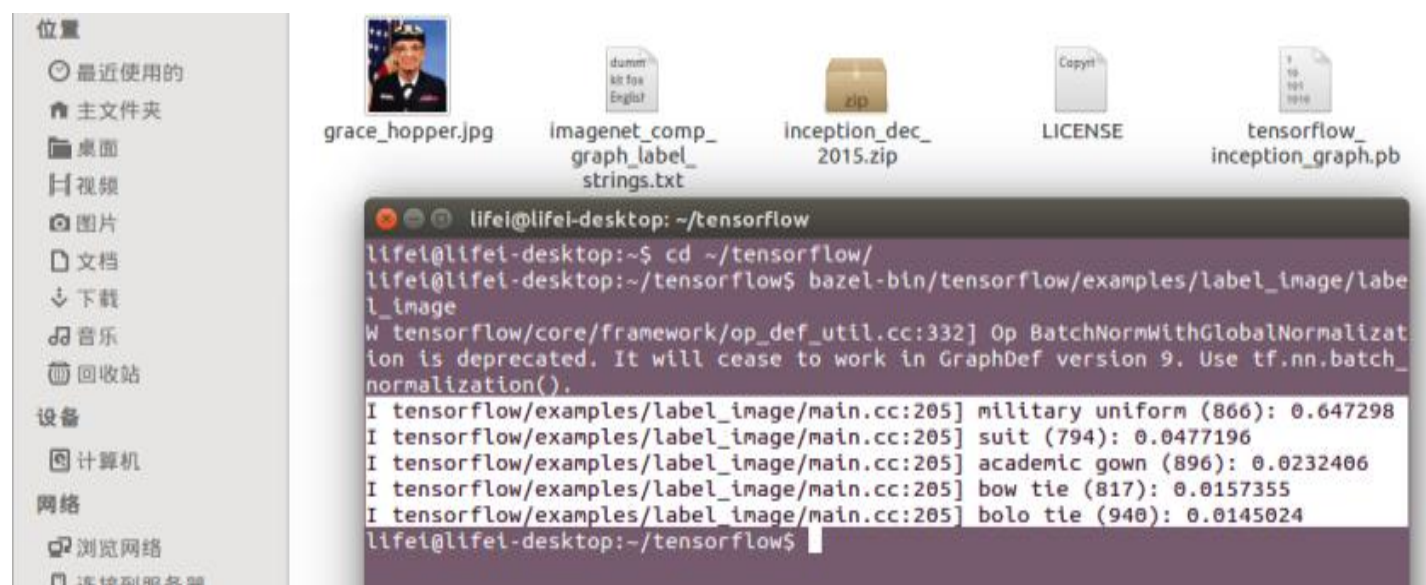
编译例程 label_image 的 C++代码:

```
$ bazel build tensorflow/examples/label_image/...
```

运行:

```
$ bazel-bin/tensorflow/examples/label_image/label_image
```

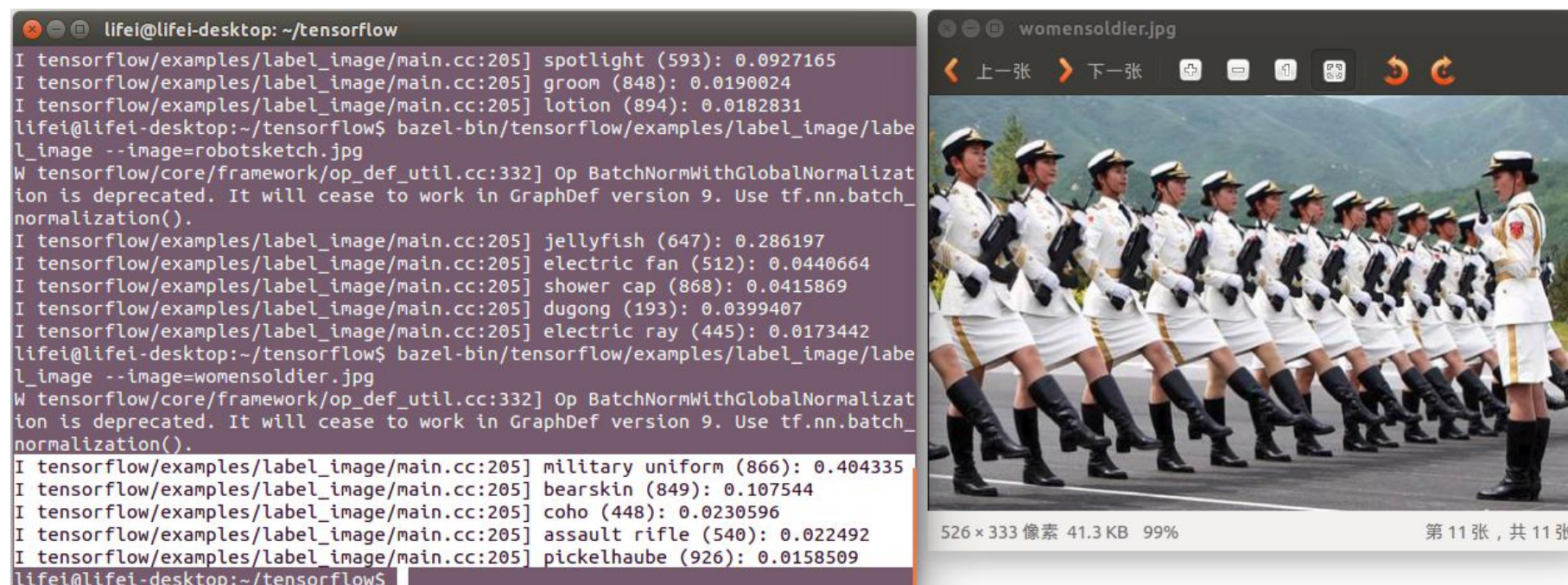
该例程对默认图像的识别结果如下:



可见，该例程正确的识别出了画面中的多种元素。

还可以用来识别自己提供的图像，将图像放在~/tensorflow 目录下：

```
$ bazel-bin/tensorflow/examples/label_image/label_image --image=womensoldier.png
```



可见，仍然识别出了图像中的主要元素。

三、训练自己的图像识别模型

使用了 image_retrain 例程，编译：

```
$ cd ~/tensorflow
```

```
$ bazel build tensorflow/examples/image_retraining:retrain
```

在 python 脚本注释中可以查看对自行添加图像的目录放置要求：

```
~/tensorflow/tensorflow/examples/image_training/retrain.py
```

我收集了网络上一些手势图像，第一类是心形动作，其他手势都放入 other 分类。图像文件放置的位置是：

```
~/gestrue_photos/heart
```

```
~/gestrue_photos/other
```

图像名字简单为好，例如：h1.jpg、h2.jpg.....

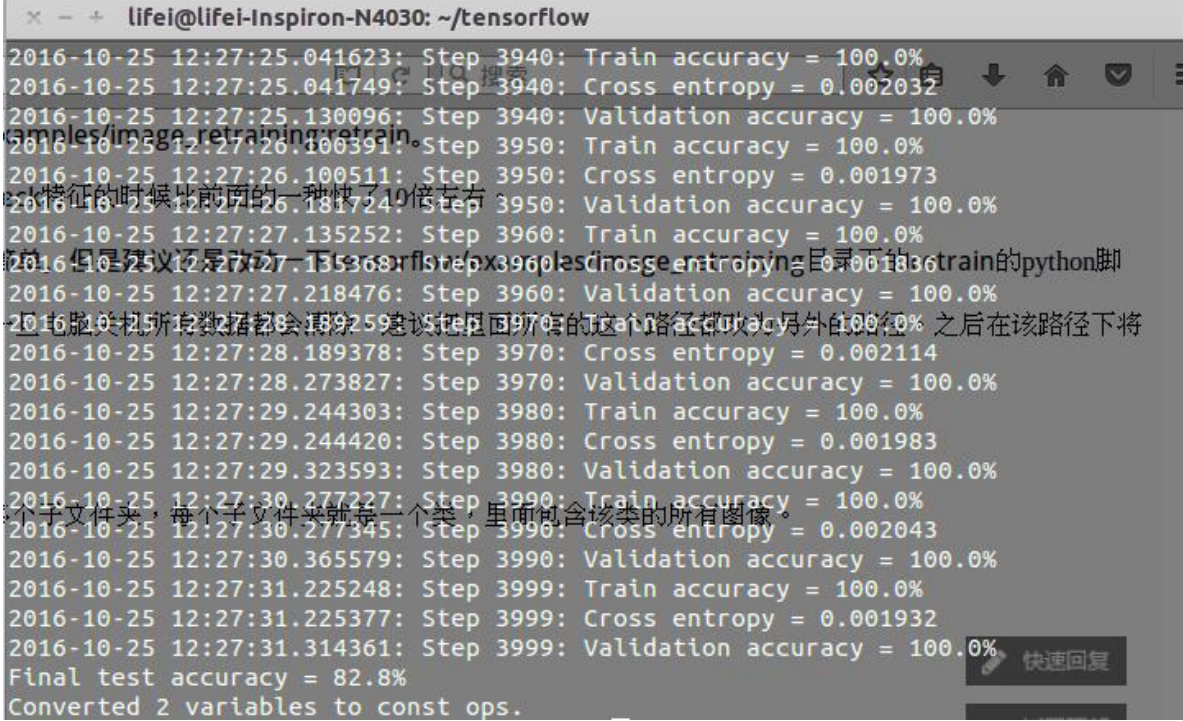
子目录名就是分类的名字。

运行：

```
$ bazel-bin/tensorflow/examples/image_retraining/retrain --image_dir ~/gestrue_photos
```

该脚本首先会下载官方数据集。同上，若下载不成功则需要手动下载，链接在上述 retrain.py 文件中，下载默认位置为/tmp/imagenet，数据集为 inception-2015-12-05.tgz。

训练过程：



训练完毕后，在/tmp 下生成两个文件 output_graph.pb 和 output_labels.txt，分别是训练后的图结构和到分类标签的映射。

得到这两个文件后即可用于对自己的图像进行识别，使用茄子大头贴和笔记本摄像头拍摄自己的心形手势照片：

```
$ cd ~/tensorflow
```

```
$ bazel-bin/tensorflow/examples/label_image/label_image \
```

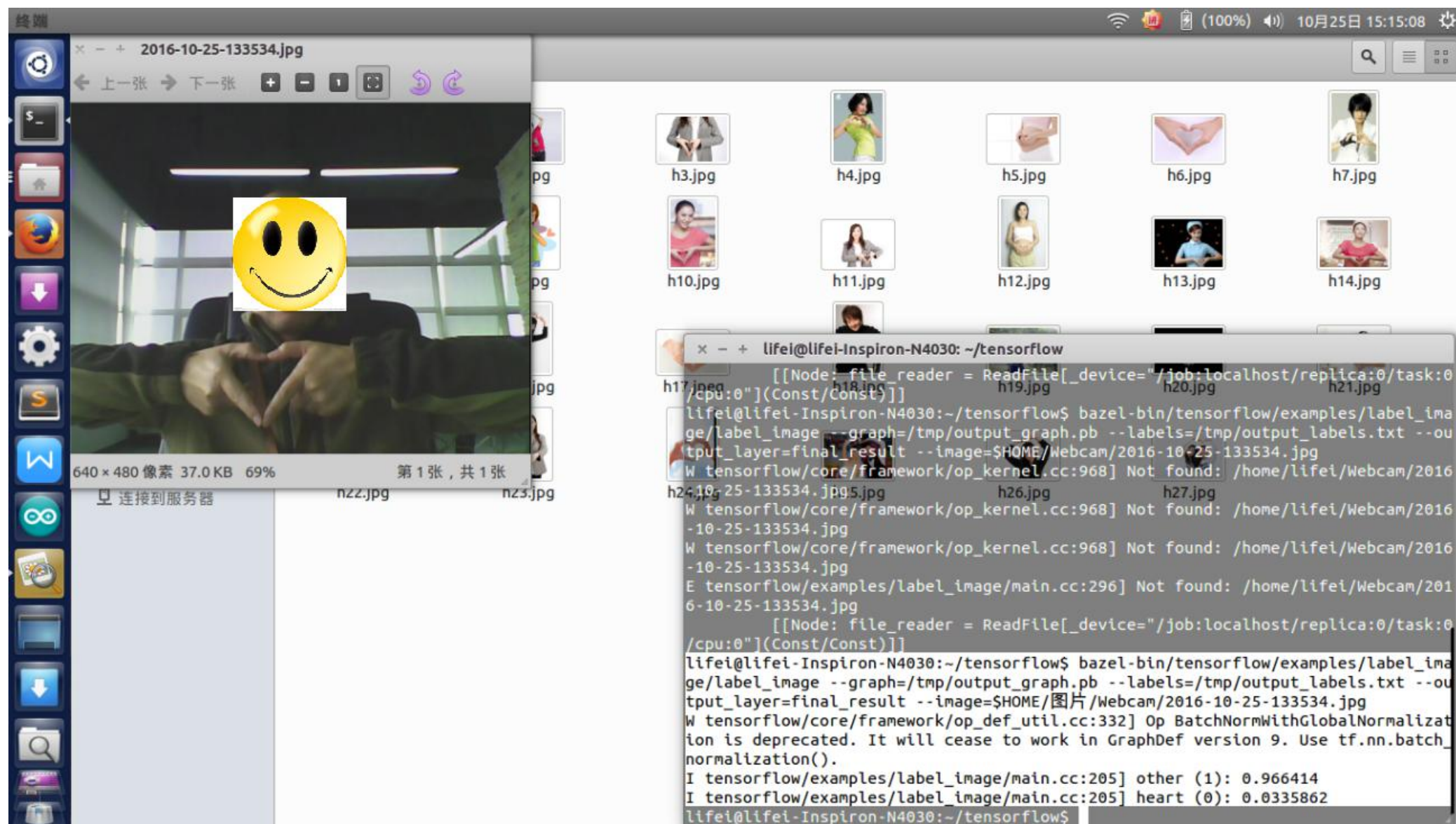
```
--graph=/tmp/output_graph.pb \
```

```
--labels=/tmp/output_labels.txt \
```

```
--output_layer=final_result \
```

```
--image=$HOME/图片/Webcam/2016-10-25-133534.jpg
```

识别结果：



可见，完全识别错了.....

原因是，深度学习对输入训练数据要求较高：特征鲜明、样本数量大、图像要有代表性，每一个类别至少要有 100 张输入。

四、训练自己的图像识别模型（官方的花卉识别数据集）

下载数据集：

```
$ cd ~
```

```
$ curl -O http://download.tensorflow.org/example_images/flower_photos.tgz
```

```
$ tar xzf flower_photos.tgz
```

该数据集包含了 daisy、dandelion、roses、sunflowers、tulips 五种花卉，每种提供了 600~900 张照片。

训练：

```
$ bazel-bin/tensorflow/examples/image_retraining/retrain --image_dir ~/flower_photos
```

在旧 DELL 笔记本（CPU 为 i5 M480）上的训练时间约为 24 小时。

训练完毕后即可识别：

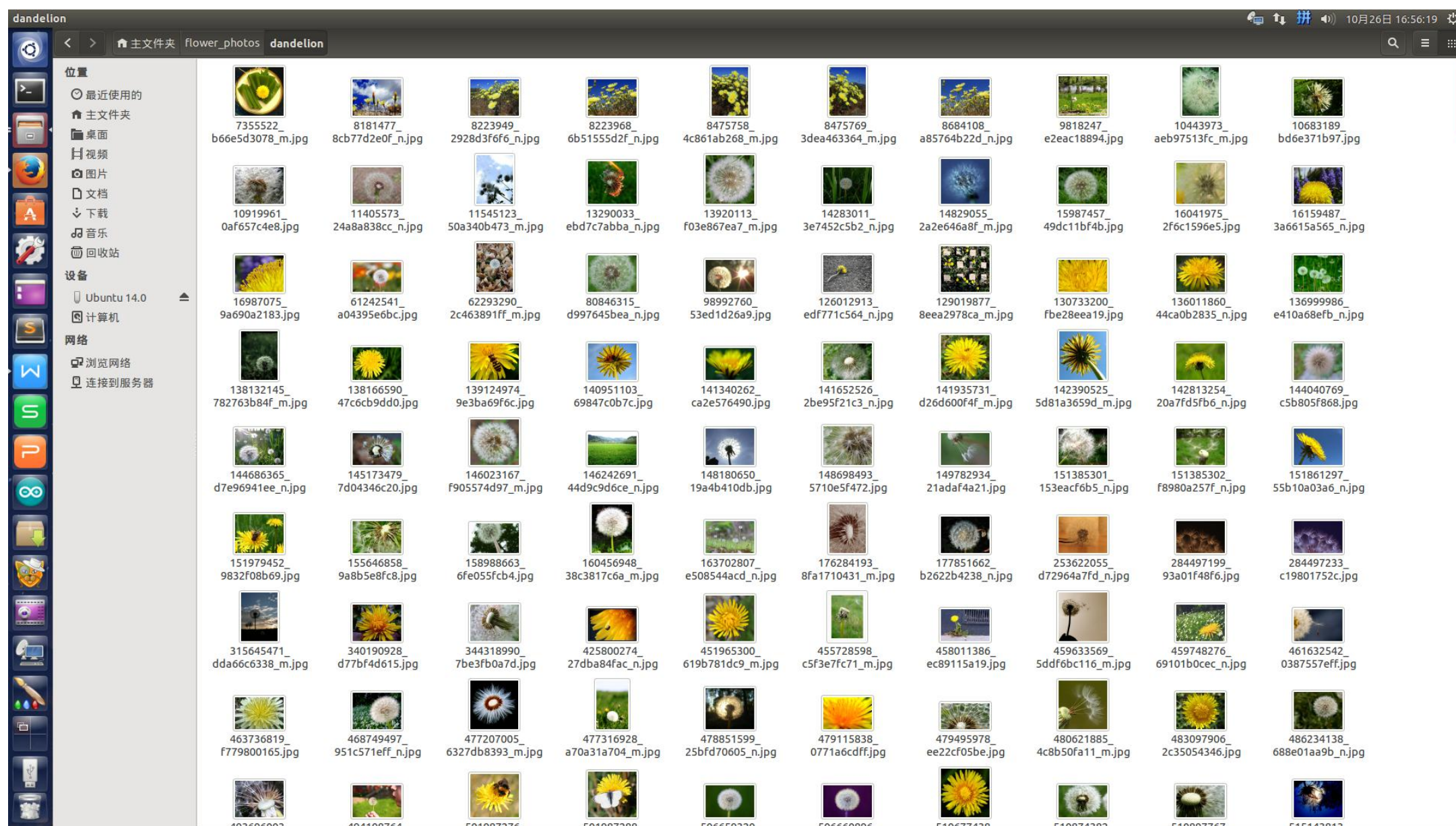
```
$ bazel-bin/tensorflow/examples/label_image/label_image \
--graph=/tmp/output_graph.pb \
--labels=/tmp/output_labels.txt \
--output_layer=final_result \
--image=$HOME/flower_photos/daisy/21652746_cc379e0eea_m.jpg
```

识别结果：

```
lifei@lifei-Inspiron-N4030:~/tensorflow$ bazel-bin/tensorflow/examples/label_image/label_image \
--graph=/tmp/output_graph.pb --labels=/tmp/output_labels.txt \
--output_layer=final_result \
--image=$HOME/flower_photos/daisy/21652746_cc379e0eea_m.jpg
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization().
I tensorflow/examples/label_image/main.cc:205] daisy (4): 0.997822
I tensorflow/examples/label_image/main.cc:205] sunflowers (3): 0.0015008
I tensorflow/examples/label_image/main.cc:205] dandelion (2): 0.000432897
I tensorflow/examples/label_image/main.cc:205] tulips (0): 0.00016656
I tensorflow/examples/label_image/main.cc:205] roses (1): 7.79277e-05
lifei@lifei-Inspiron-N4030:~/tensorflow$
```

可见，能够很好的识别出 daisy。

Tensorflow 基本的使用规则就是，需要大量的训练数据：



李飞

sune.cn@aliyun.com

2016 年 10 月 26 日