

RUNNING THE PREDICTOR

After downloading and unzipping the project folder, download the following three models from Google Drive (all are not required) and place them in the project folder:

Filename: 'model_98percent.h5'

Type: 17 flower classifier, pretrained on Google's InceptionV3

List of flowers is in *predictor.py* (names) AND - <http://www.robots.ox.ac.uk/~vgg/data/flowers/17/>
<https://drive.google.com/file/d/0B-xwCRwClzyueXdZd1U2SnIXOWs/view?usp=sharing>

Filename: 'model_70percent.h5'

Type: 17 flower classifier, "from scratch" i.e. doesn't use pretrained models

List of flowers, same as above

<https://drive.google.com/file/d/0B-xwCRwClzyuRndEWHRvVk1pcms/view?usp=sharing>

Filename: '102_model.h5'

Type: 102 flower classifier, pretrained on Google's InceptionV3

List of flowers is in *predictor.py* (names_102) AND -

List of flowers at - <http://www.robots.ox.ac.uk/~vgg/data/flowers/102/categories.html>
<https://drive.google.com/file/d/0B-xwCRwClzyuVXA4bk5QTWs2aW8/view?usp=sharing>

After downloading the desired models, in a command prompt run:

python predictor.py

After starting, it may take a bit to load, when it gets to the following screen below enter in the model that you want to load. (If you have the non-GPU version of TensorFlow you won't get the lines that start with I)

```
Using TensorFlow backend.  
I c:\tf_jenkins\home\workspace\release-win\device\  
I c:\tf_jenkins\home\workspace\release-win\device\  
I c:\tf_jenkins\home\workspace\release-win\device\  
I c:\tf_jenkins\home\workspace\release-win\device\  
I c:\tf_jenkins\home\workspace\release-win\device\  
Please enter the model that you want to load:  
Available models are:  
    102_model.h5  
    model_98percent.h5  
    model_70percent.h5  
Model to load:  
```

Afterwards, enter in one of the three models that listed – these are the same models that were downloaded before this.

```
Model to load: 102_model.h5
Loading model 102_model.h5

E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
E c:\tf_jenkins\home\workspace\release-win\device\
```

Note: TensorFlow will most likely give you multiple errors – these are a bug with TensorFlow version 1.0.1 that will be fixed when the next version comes out – this doesn't affect the program.

It may take up to 30 seconds to load the model into memory – after it is successfully loaded you'll receive this message:

```
Model loaded successfully.

Please enter the full name of the image to predict.
Note: must be in directory predict/to_predict/
```

At this point you can enter in the full name of an image that you want to predict. *Make sure that all images to predict are within the **predict/to_predict/** directory.* For best results try to feed images that have a height and a width greater than 300 pixels – they don't have to be the same.

For this example, we will use the provided image *dandelion.jpg* located in *predict/to_predict/*:
(image was resized for this document)



Image courtesy of: <https://beediary.wordpress.com/2008/05/22/dandelion-days/>

Note: for this first prediction, it will take about 5 seconds or so, all subsequent predictions will be nearly instantaneous.

Predicted Result for *dandelion.jpg*

```
Please enter the full name of the image to predict.  
Note: must be in directory predict/to_predict/
```

```
dandelion.jpg
```

```
***** Top Five Predictions *****
```

Confidence	Flower Name
99.99%	common_dandelion
0.01%	colts_foot
0.0%	english_marigold
0.0%	globe_thistle
0.0%	carnation

```
Please enter the full name of the image to predict.  
Note: must be in directory predict/to_predict/
```

The predictor will give you the top five predictions. Each prediction comes with a confidence percentage – as we can see here it is 99.99% confident that the image was a *common_dandelion* (which it is), but it also gives its confidence for the next four best predictions.

Feel free to try more images from the *predict/to_predict/* directory. The ones labeled 102 in the beginning only work for the '102_model.h5' model. All will have their proper flower name in the filename, but not all will be predicted perfectly. This is to show where the model(s) fail due to having a small amount of data for training.

One sure way of getting it to mispredict is to pass an image that has a group of the flower in it because most of the training data was of a single flower.

To exit out either type 'exit' instead of a flower name or kill it however you want to.