

## Part 2 - Building the Command Line Application

Now that you've built and trained a deep neural network on the flower data set, it's time to convert it into an application that others can use. Your application should be a Python script that run from the command line. For testing, you should use the saved Keras model you saved in the first part.

### Specifications

The project submission must include a `predict.py` file that uses a trained network to predict the class for an input image. Feel free to create as many other files as you need. Our suggestion is to create a module just for utility functions like preprocessing images. **Make sure to include all files necessary to run the `predict.py` file in your submission.**

The `predict.py` module should predict the top flower names from an image along with their corresponding probabilities.

### Basic usage:

```
$ python predict.py /path/to/image saved_model
```

### Options:

- `--top_k` : Return the top  $K$  most likely classes:

```
$ python predict.py /path/to/image saved_model --top_k K
```

- `--category_names` : Path to a JSON file mapping labels to flower names:

```
$ python predict.py /path/to/image saved_model --category_names map.json
```

The best way to get the command line input into the scripts is with the [argparse module](#) in the standard library. You can also find [a nice tutorial for argparse here](#).

### Examples

For the following examples, we assume we have a file called `orchid.jpg` in a folder named `/test_images/` that contains the image of a flower. We also assume that we have a Keras model saved in a file named `my_model.h5`.

## Basic usage:

```
$ python predict.py ./test_images/orchid.jpg my_model.h5
```

## Options:

- Return the top 3 most likely classes:

```
$ python predict.py ./test_images/orchid.jpg my_model.h5 --top_k 3
```

- Use a `label_map.json` file to map labels to flower names:

```
$ python predict.py ./test_images/orchid.jpg my_model.h5 --category_names label_map.json
```

# Workspace

## Install TensorFlow

We have provided a Command Line Interface workspace for you to run and test your code. Before you run any commands in the terminal make sure to install TensorFlow 2.0 and TensorFlow Hub using `pip` as shown below:

```
$ pip install -q -U "tensorflow-gpu==2.0.0b1"
```

```
$ pip install -q -U tensorflow_hub
```

## Images for Testing

In the Command Line Interface workspace we have we have provided 4 images in the `./test_images/` folder for you to check your `prediction.py` module. The 4 images are:

- `cautleya_spicata.jpg`
- `hard-leaved_pocket_orchid.jpg`
- `orange_dahlia.jpg`
- `wild_pansy.jpg`