**EXPLICIT IMAGE CLASSIFICATION WALK-THROUGH**

Instructions for Running / Training / Testing Image Classification with TensorFlow Using Inception on Windows 10

**Table of Contents:**

1) Install python and pip

2) Install TensorFlow (TS)

3) Clone Repository

4) Get and Prepare data

5) Separate out some test images

6) Run ***retrain.py***

7) Run ***test.py***

----------------------------------------------------------------------------------------

**1) Install python 3.**

Install python3 at least v3.6. and the pip package installer.

* <https://phoenixnap.com/kb/how-to-install-python-3-windows>

**2) Install pip and TensorFlow (TF).**

Install TF at least version 1.5.0, I used v1.5.0. To install it type the following command in your windows terminal:

* *pip install tensorflow==1.5.0*

**3) Clone the repository**

Clone it to a location you’d like to work in. I stored mine on my desktop:

C:\Users\NY\Desktop\explicit\_img\_detector

Type the following command in your terminal:

* git clone https://github.com/en-wai/explicit\_img\_detector.git

**4) Get and prepare your data.**

I scraped the web to make my own data which I have provided the link to. Download the zip file and extract the contents (NB: just copy or cut the training\_images & test\_images folders) into the cloned repository on the desktop.

* Link to data: https://drive.google.com/open?id=1MumPH7AGX3aNGSLafe9eNTembiSXzT0s
* Password is: angelique

Just in case you get confused or mess anything up. Note that that the file structure should look like this:

/explicit\_img\_detector

.

.

/training\_data

.

.

/explicit

/non-explicit

/test\_data

ie. The *explicit* and *not-explicit* folders are both in the *training\_data* folder. The *training\_data* is at the same level as the *test\_data* folder and both are to be in the *explicit\_img\_detector* folder.

**5) Separate data**

In downloading the images, I scraped explicit images into one folder and non-explicit images into another folder. These two together make the training data. In this classification task I only need two groups of items. It can me any number of categories depending on the task at hand. The next step is to separate our select some images randomly from both the explicit and non-explicit data sets. This is the data on which we will test how well our model generalizes (ie. how accurately it is able to label data it hasn’t seen before) so it is advisable to separate it before creating the model. I have already done this step for you.

**6) Train the model**

Now train the model by running the ***retrain.py*** script which should be in the *explicit\_img\_detector* folder. The file is called accurately called “retrain.py” because it is based on the large neural network (NN) that google has already trained and made open source. They used an algorithm called **inception**. I don’t know if you know what a neural network is but we leverage the power of transfer learning and build on top of it so instead of training the algorithm from scratch (which would require millions of images and some serious computing power which we can’t have) we just train the last layer. Just for some context the NN used to train inception is more than 14 layers deep. Simply put we borrowed what google has done and we built on it. I have refactored the original retrain.py file so that it performs some minor checks. It will print out meaning full errors if you mess anything up. I don’t expect anything to be out of place if you leave the file organizations as they are. The refactoring is for improved readability and also to make using the command line not necessary. At this point there will be a lot of output on your command line. Just let it run its course.

NB: In training the model, the script will connect to the internet to download the inception model so you need an internet connection. I could have included it but I was trying to keep the file size to a minimum so just connect to the internet and let the script do all the work for you.

**7) Run *test.py***

Test how well your model performs on the test data (new data). Most times when people run classifiers you get a result between 0 and 1 as the level of confidence. I took it a step further to make it more interesting so be patient the image in question will pop up and the confidence level will be displayed. Now I have placed some images in the test data folder (it requires at least 6 images). If you have less than the required number it will throw an error. This is just how I prefer to do things…. Why train a model and not test it on enough data?

**Done!!**

**Important notes**

If you’d like to make the accuracy better, the 2 general steps to accomplish this would be:

1. *Use more training images. You can download more images into the appropriate folders. Generally, the more data you train on the better the model performs.*
2. *In retrain.py, I set the how\_many\_training\_steps parameter to something higher than 500. Increasing this number can yield better results but it comes at the cost of time.*