

Install the TensorFlow. Follow the instruction from here:
<https://www.tensorflow.org/install/>

The screenshot shows the TensorFlow website's installation page. The header is orange with the TensorFlow logo and navigation links: Install, Develop, API (v1.8), Deploy, Extend, Community, Versions, and Ecosystem. A search bar and a link to GitHub are also present. The main content area is titled "Installing TensorFlow" and includes a sidebar with links to Python, Ubuntu, MacOS, Windows, From source, Transitioning to TensorFlow 1.0, and Other Languages (Java, Go, C). The main text provides instructions for installing TensorFlow on various operating systems (macOS 10.12.6 or later, Ubuntu 16.04 or later, Windows 7 or later) and languages (Python, Java, Go, C). It also mentions that the Python TensorFlow API changed from version 0.n to 1.0 and provides links to guides for migrating older applications. The footer contains links to Stay Connected (Blog, GitHub, Twitter, YouTube) and Support (Issue Tracker, Release Notes, Stack Overflow).

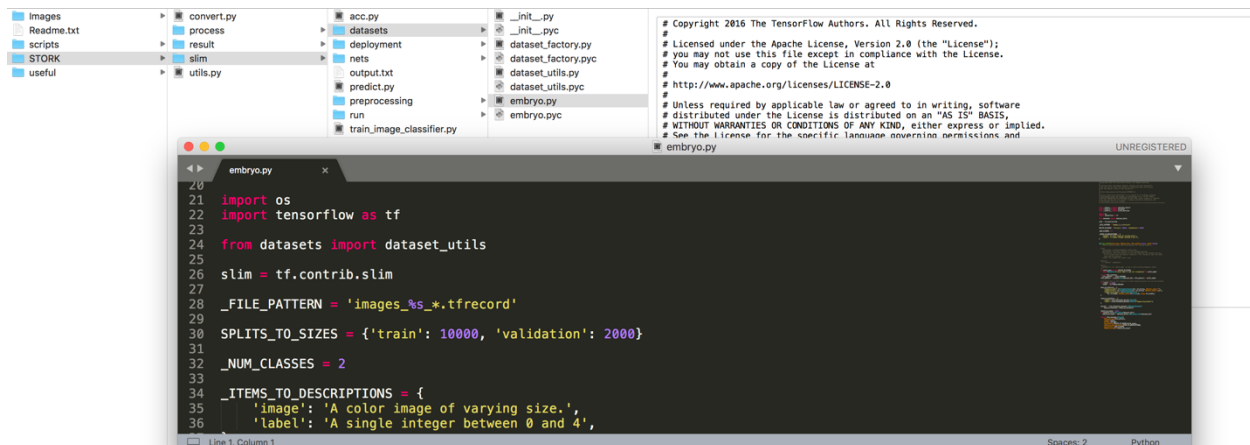
Pre-trained Models of CNN architectures should be downloaded from the "Pre-trained Models" part of <https://github.com/wenwei202/terngrad/tree/master/slim#pre-trained-models> and be located in your machine (e.g. GitHub_STORK/scripts/slim/run/checkpoint). The files for pre-trained models are available under the column named "Checkpoint".

The screenshot shows a file explorer window displaying the directory structure for pre-trained models. The left pane shows a tree view with folders: Images, Readme.txt, scripts, STORK, and useful. The right pane shows the contents of the 'checkpoint' folder, which includes a file named 'inception_v1.cpkt'. Other files visible in the tree include 'convert.py', 'process', 'result', 'slim', 'utils.py', 'acc.py', 'datasets', 'deployment', 'nets', 'output.txt', 'predict.py', 'preprocessing', 'run', and 'train_image_classifier.py'.

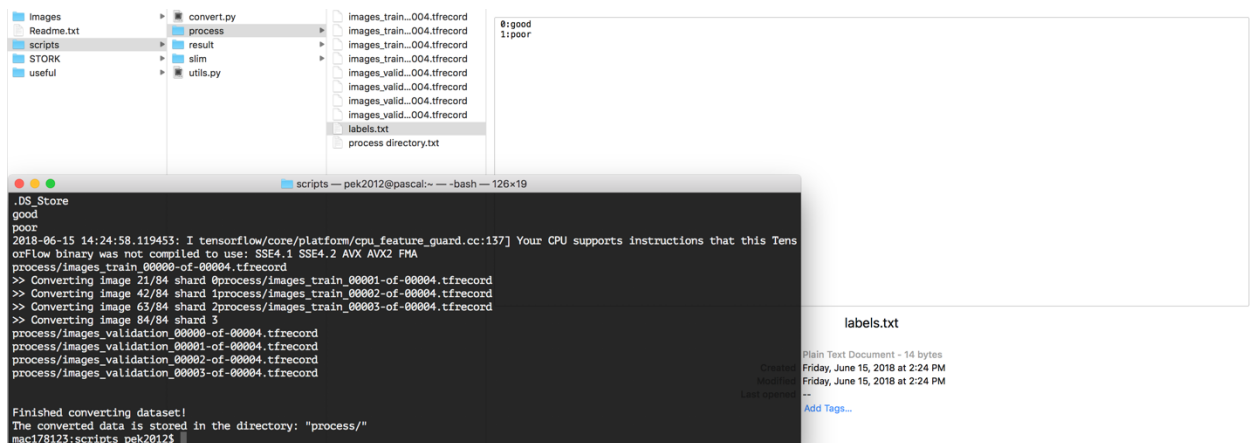
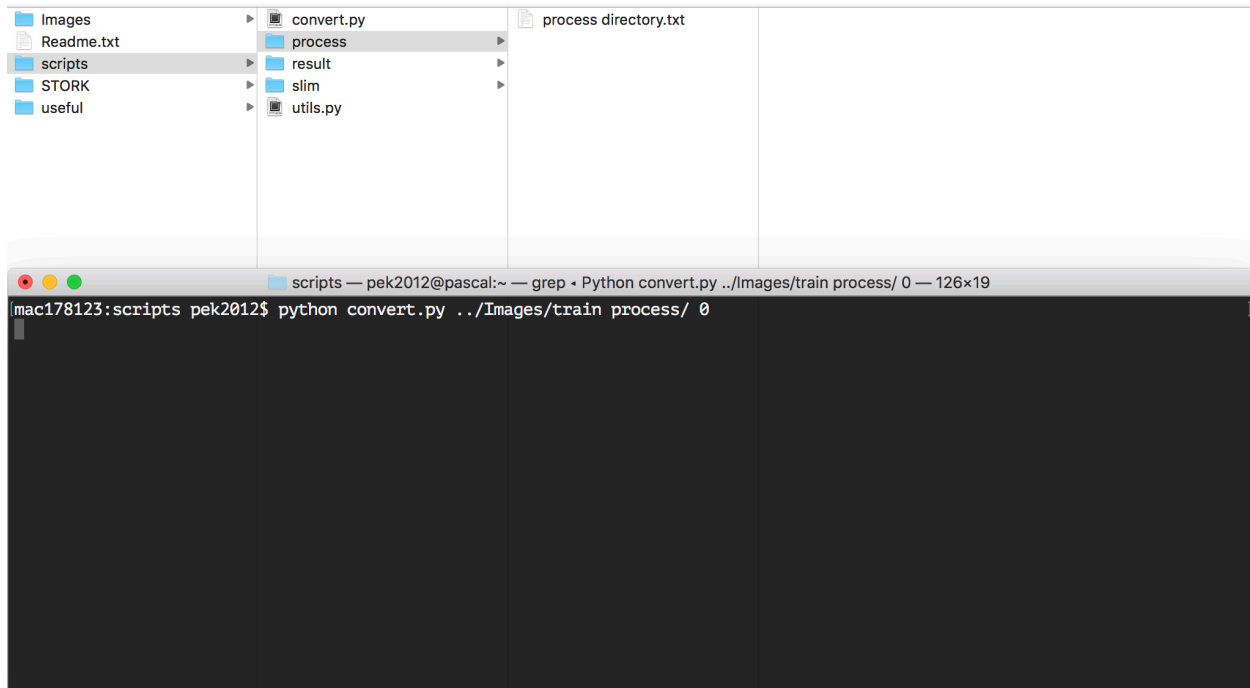
Divide the images with the original size into two or more classes based on the aim of classification (e.g., discrimination of good-quality and poor-quality embryos). 85% of images in each class will be selected as Train set (train and validation) and 15% for Test set.



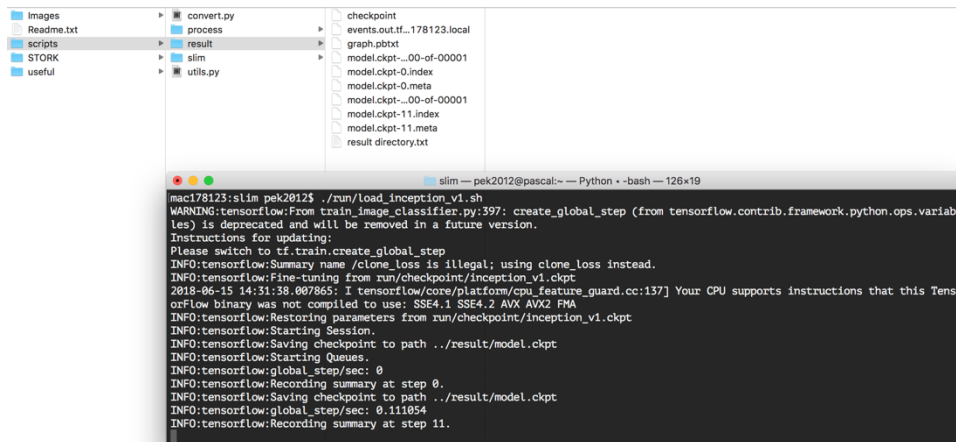
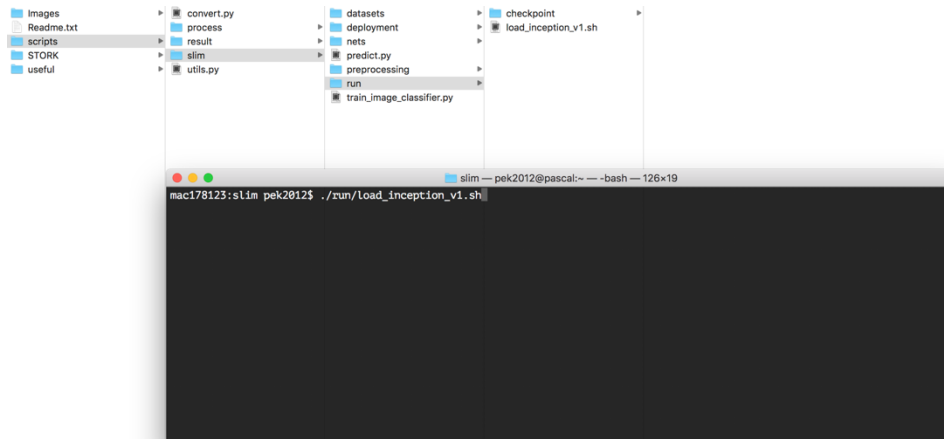
`_NUM_CLASSES` should be set in `embryo.py` (this script is located in `STORK/scripts/slim/datasets`).



Run the convert.py (it is located in the "STORK/scripts" directory) to allocate the suitable percentage of images to train and validation sets. the convert.py needs three arguments including: the address of images for training, the address of where the result will be located, and the percentage of validation images for the training step (e.g., \$ python convert.py ../Images/train process/ 0). It will save converted .tf records in the "process" directory.

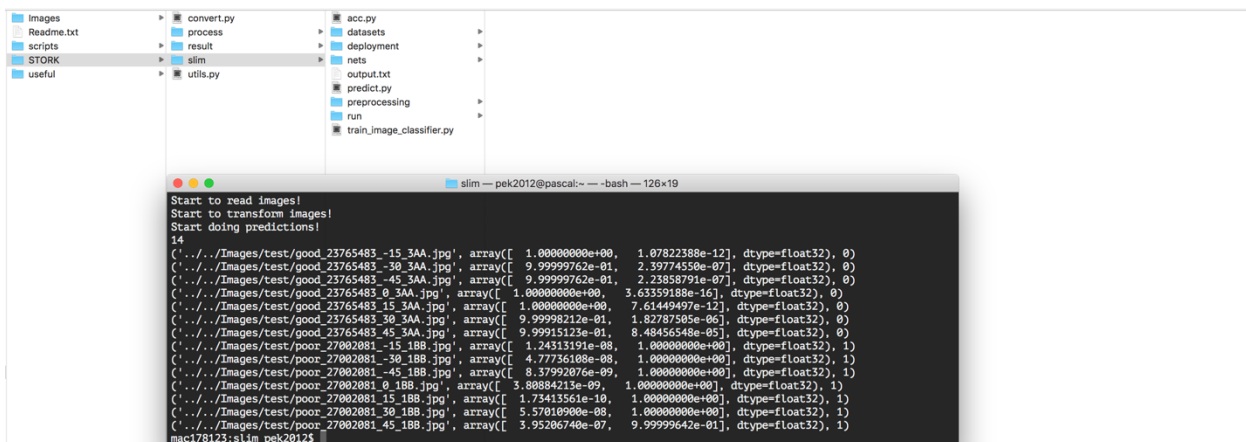
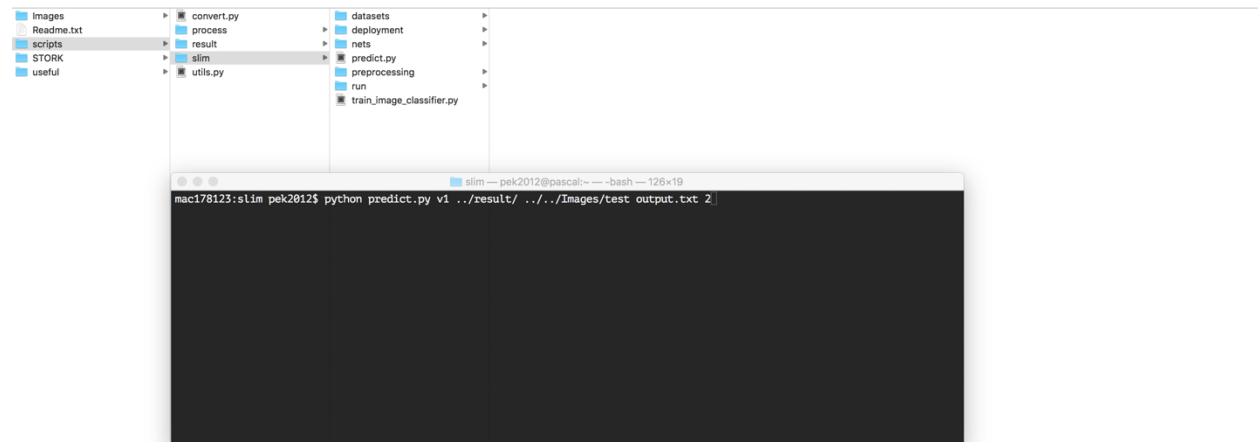


The Inception-V1 architecture should be run on the Train set images from the "STORK/scripts/slim" directory. First got the the following directory: STORK/scripts/slim. Then open load_inception_v1.sh located in "run/" directory and edit PRETRAINED_CHECKPOINT_DIR, TRAIN_DIR, and DATASET_DIR addresses. See the load_inception_v1.sh, for instance. Then, run the following command in shell script:
\$./run/load_inception_v1.sh



The trained algorithms should be tested using test set images. In folder "STORK/scripts/slim", predict.py loads a trained model on provided images. This code get 5 argu/resultments:

```
$ python predict.py v1 ../result/ ../Images/test output.txt 2
```



The accuracy can be measured using accuracy measurement codes ("acc.py") in "useful" folder. The output.txt file should be in the same folder that you are running acc.py. Then run the following code:

```
$ python acc.py
```

