Assignment #3
COMS 4995.06. Fall 2018.
Due Nov 5th.

Part 1: Develop a CNN-based model to classify QuickDraw images (30 points).

- Use the loader on CourseWorks to create an “animals” dataset. Develop a CNN-based classifier. Design and run an experiment to train and evaluate your model. Include the results of your evaluation in your submission.

- *Note: it is not necessary to use the full ~100K images / class. Feel free to choose a smaller number that gives reasonable accuracy.*

Part 2: Design and implement an AutoML-style image classifier and demonstrate it on the flowers dataset (70 points).

Download and extract the flowers dataset. Examine the directory structure.

- cd ~
- curl -O http://download.tensorflow.org/example_images/flower_photos.tgz
- tar xzf flower_photos.tgz
- ls

Notice there are five directories (daisy, dandelion, roses, etc). Each of these contains several hundred images (of varying sizes) of that flower.

Imagine this is training data provided by your user. Your program should provide a trained model that accurately classifies their images.

Use transfer learning to develop your model. Begin with a convolutional base (with pretrained weights). Add one or more Dense layers as necessary. Train your model, and include an evaluation in your submission. Include example predictions. If additional accuracy is needed, consider fine-tuning. If you wish, you may customize the complexity of the model you train based on the input dataset.

As always, your program should run end-to-end in a notebook with no user intervention necessary.

Your AutoML-style program should work well (e.g., train an accurate model, without overfitting) for several similar datasets, with between 50 and 500 images per class.

EC1: Recognize landmarks on the Columbia campus.

- Use the FFmpeg trick discussed in class and your AutoML program from part 2 to develop an image classifier that can accurately identify landmarks on the Columbia campus, with only a short video of each landmark used to train your model. As always, your program should run end-to-end without user intervention. (Host your training data on the Cloud). Use either your Columbia account, or a storage option provided by one of the major Cloud providers - most of which provide a free trial. If you would like cloud credits on GCP, drop a line.

- *Note: developing a more in depth version of this using Android, iOS, or TensorFlow.js would make a good course project.*

EC2: Visualize intermediate activations / convnet filters.

- Produce a clear and well-commented implementation of these two techniques from Deep Learning with Python notebook 5.4 in TensorFlow, using a GradientTape.

EC3: Visualizing class activation heatmaps.

- Same as EC2, except for this technique.