layout: post
title: "fast.ai Image classification"
date: 2019-09-23
excerpt: "Computer Vision "
tag:

- sample post
- post
- images

comments: true

fast.ai Course Image classification

About the course

While I w as looking for some good courses on Deep Learning, my search ended w hen I came across this course by fast.ai: Practical Deep Learning for Coders, v3. This is the part 1 of the tw o part series by fast.ai. Part 1 covers four applications:

1. Computer Vision
2. Natural Language Text
3. Collaborative filtering
4. Tabular Data

The course teaches how to train PyTorch models using the fastai library. The instructor Jeremy is the cofounder of the Fast.ai. The course helps the user set up the cloud service in the beginning before getting into the actual deep learning problems. In this course Jeremy has follow ed a unique approach of hands on experience in coding first and then getting into the concepts of w hat is going on under the hood. My experience in Computer Vision during my Capstone project through Master of Data Science program w as in Keras but I found the fastai commands quite straightforw ard.
The number of lines of codes in fastai is generally 1/6th those in Keras.

Motivation

The 1st tutorial video w as based on Pet detection w hich is classification of 25 breeds of dogs and 12 breeds of cats. Gone are the days w hen w e w ere fascinated w ith Dog and Cat classification, this problem has taken it to the next level by detecting even more intricate features detection hence the name fine grained image classification. Upon completion I decided to execute Image classification w hich w as more relevant to me. I had recently moved to Calgary and already fallen in love w ith the Banff area w hich is home to several beautiful lakes. Whenever some one show ed me a picture of a any of those lakes I could hardly tell w hich Lake that picture belonged to. I realized for me to be able to distinguish these lakes from each other I w ould need to identify more intricate differences similar to w hat Jeremy did for cat, dog breed identification. I decided to build a classification problem for these Lakes. I chose the lakes I had visited viz.

1. Lake Moraine
2. Lake Louise
3. Lakes Grassi
4. Lake Minnew anka
5. Lake Emerald

Data collection

The first task w as to collect the Images of these lakes to train my model. I used the add-on for Mozilla Firefox

Google Images Downloader
by Rushikesh Tade

Downloads full sized images & Links in Google Images Windows. It is different than other image downloaders since it goes one step inside the links & gets high quality images for you.
This addon automatically also filters websites not giving images.

This add-on allows you to directly download multiple pictures from google search.
With not much effort I collected around 100 images for each of these Lakes. Now that I had all the images I had to organize them in the way my I use them to train my model.

Before any work can be done a dataset needs to be converted into a fastai DataBunch object, and in the case of the computer vision data - specifically into an ImageDataBunch subclass.

To be able to use the Images to train the model I would need to organize my data in one of the formats the following functions handle.

1. *from_folder*

```
path\
  train\
    clas1\
    clas2\
    ...
  valid\
    clas1\
    clas2\
    ...
  test\
```

In this method we add the images in their specific class folders in the train, valid(validation) and if we wish in the test folders. For instance class1, class2 folders within the train folder will be considered for training the model and the images within the class1 folder will be assigned the label "class1" and so on.

2. *from_csv*

Create an ImageDataBunch from path by splitting the data in folder and labelled in a file csv_labels between a training and validation set. Use valid_pct to indicate the percentage of the total images to use as the validation set. An optional test folder contains unlabeled data and suffix contains an optional suffix to add to the filenames in csv_labels (such as '.jpg').

3. *from_df*

	name	label
0	train/3/7463.png	0
1	train/3/21102.png	0
2	train/3/31559.png	0
3	train/3/46882.png	0
4	train/3/26209.png	0

Reference:

1. The credit for this blogpost goes to Practical Deep Learning for Coders, v3 : course_url
2. The documentation by fast ai has been very helpful : doc_url