# Image Classification using Deep Learning

CUNY 698 - Project Proposal

*Walt Wells, 2018*

A significant byproduct of Moore's law is how large scale computation is now considerably less expensive and is accessible to anyone with a credit card and a laptop. As a result, computationally intensive modeling techniques like deep learning that can accurately model highly dimensional data like images, video, and audio are becoming increasingly available.

## Problem

Deep learning techniques are one of the most viable approaches to building a machine learning system that can operate with accuracy and precision in a complex real-world environment [1]. This is important because for difficult problems with highly dimensional data like image classification, deep learning models like Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs) are among the best performers in benchmarked competitions and are even outpacing their human subject matter expert counterparts at image classification tasks [2]. But how do they work?

The final deliverable will seek to answer this question by exploring a variety of deep learning techniques for conducting image classification over a popular benchmarking dataset (ImageNet). The paper will focus on two tracks: a) *data engineering*, or managing the storage, computational infrastructure and backend for handling the dataset and models; and b) *deep learning modeling*, or training and optimizing neural networks (ANN, CNN, DNNs) to make classification predictions, and exploring methods for measuring, comparing, and improving model performance [3]-[7].

## Data Engineering

Managing the compute and storage used to explore deep learning techniques over the ImageNet dataset will require the use of modern big data techniques. Keras and Tensorflow are deep learning backend libraries that allow for a computational job to be broken out across multiple compute nodes. The project will explore the use of deep learning modeling libraries in Python (Scikit-learn) and R (darch, deepnet, caret). To faciltiate reproducability, the project will leverage Jupyter Notebooks and RStudio Server notebooks served from VMs or cluster headnodes, and TensorBoard to read logfiles. Github will be used to support the codebase. DataProc clusters and object storage will be used to manage the front and back ends on the Google Compute Platform. In addition, GCP has recently made their TPUs available to the public for rental so the project will explore using TPUs or GPUs [8].

The Data Engineering section of the final deliverable will explore and outline all of these options.

## Deep Learning Modeling

The project will focus on creating models that excel at the following image classification task as defined by the ImageNet 2012 challenge:

"For each image, algorithms will produce a list of at most 5 object categories in the descending order of confidence. The quality of a labeling will be evaluated based on the label that best matches the ground truth label for the image. The idea is to allow an algorithm to identify multiple objects in an image and not be penalized if one of the objects identified was in fact present, but not included in the ground truth [9]."

The deep learning modeling section of the final deliverable will compare the architecture, training, optimization and regularization methods for 3 common neural network architectures - ANNs, CNNs, and DNNs.

## The Dataset

The ImageNet dataset is a major touchstone for image classification and deep learning benchmarking. This project will explore the 2012 release and only utilize subsets related to classification, ignoring tasks like edge detection or bounding boxes.

"The validation and test data will consist of 150,000 photographs, collected from flickr and other search engines, hand labeled with the presence or absence of 1000 object categories. The 1000 object categories contain both internal nodes and leaf nodes of ImageNet, but do not overlap with each other. A random subset of 50,000 of the images with labels will be released as validation data included in the development kit along with a list of the 1000 categories. [9]"

- Training images - 138GB
- Validation images - 6.3GB
- Test images - 13GB

**For development / testing:** Tiny Imagenet has 200 classes. Each class has 500 training images, 50 validation images, and 50 test images. Overall, the dataset is approximately .8GB [10].

## References

[1] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, Cambridge, MA: MIT Press, 2016

[2] D. Ciresan, U. Meier, J. Schmidhuber, (June 2012). *Multi-column deep neural networks for image classification.* 2012 IEEE Conference on Computer Vision and Pattern Recognition: 3642–3649. doi: 10.1109/cvpr.2012.6248110

[3] T. Rashid, *Make Your Own Neural Network*, CreateSpace Independent Publishing Platform; 2016

[4] T. Beysolow III, *Introduction to Deep Learning Using R: A Step-by-Step Guide to Learning and Implementing Deep Learning Models Using R*, New York: Apres, 2017

[5] B. Lantz, *Machine Learning with R - Second Edition: Expert techniques for predictive modeling to solve all your data analysis problems*, Birmingham, UK: Packt Publishing, 2015

[6] A. Géron, *Hands on Machine Learning with Scikit-Learn & TensorFlow*, Sebastopol, CA: O'Reilly Media, 2017

[7] G. Hinton, *Neural Networks for Machine Learning*, University of Toronto: Coursera MOOC, 2013

[8] F. Lardonis. (2018, Feb 12). *Google's custom TPU machine learning accelerators are now available in beta*, [TechCrunch] Available: Techncrunch.com

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, IJCV, 2015

[10] L. Fei-Fei, J. Johnson, S. Yeung, *CS231n: Convolutional Neural Networks for Visual Recognition*, CA: Stanford University MOOC, 2017