Sergio Munguia

**Final Project: Cats vs Dogs**

By using computer vision, we can take advantage of machine learning techniques to detect objects of interest in images and classify or identify categories of objects.

In the project we extract features, and then use them to train a model to classify or learn patterns in the image data.

We use local detectors for locally “interesting points” in the image.

These image features: are collections of locally interesting points

* Combined to build classifiers

Standard image classification approach, Extract features

Using the features of a pre-trained network, we can achieve 90% accuracy in a minute-wise performance.

In the code we instantiate the convolutional part of the model, and everything up to the fully-connected layers.

If training a large network, can use regularization to defeat overfitting.

1st We take in a trained model of data, either a CNN pretrained or CNN trained from scratch on 25,000 cats & dogs

-Locating the pre-trained “AlexNet” in file location of the folder cats\_dogs\_starter\networks

(Loading the MatConvNet data into ConvNet, a series network object from NN toolbox, using helperImportMavConvNet in Computer Vision System Toolbox.

# We will use the series network object to inspect the network architecture, classify new data, and extract network activations from specific layers.

2nd Inspect the layers of the convnet.Layers

-The convolutional layers, interspersed with rectified linear units (ReLU) and max-pooling layers

-Following layers are the 3 Fully-connected layers

-the last layer is a classification layer

Convet.Layers(end) # inspect the last layer )

3rd Inspect the network weights for the second convolutional layer

* The first layer has learned filters for capturing blog and edge features.



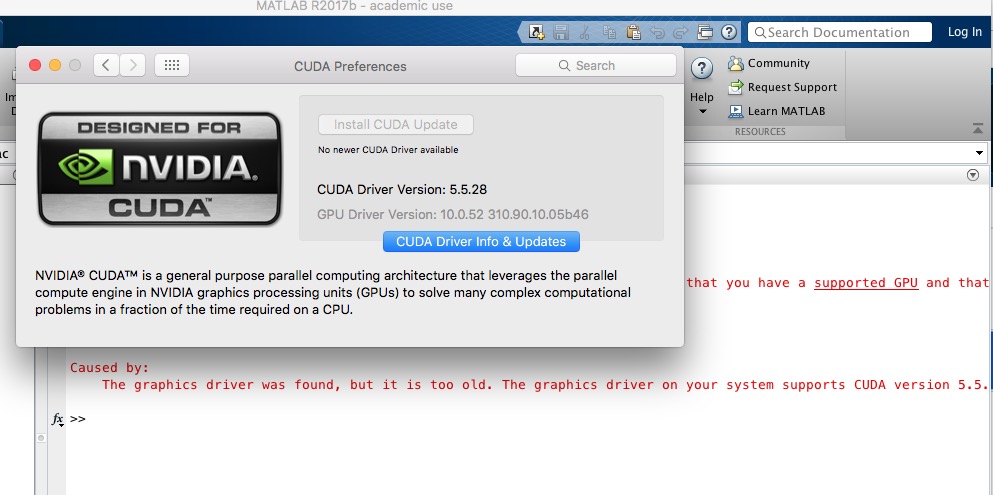
4th Inspect and extract features from one of the deeper layers using activations method.

5th Classify the training Features, training Labels

* CNN trained from scratch
* CNN using pre-trained data
* SVM Classifier
* TREE Classifier
* Naive Bayes Classifier
* K-Nearest Classifier

6th Show predications of 4 different classifiers % accuracy

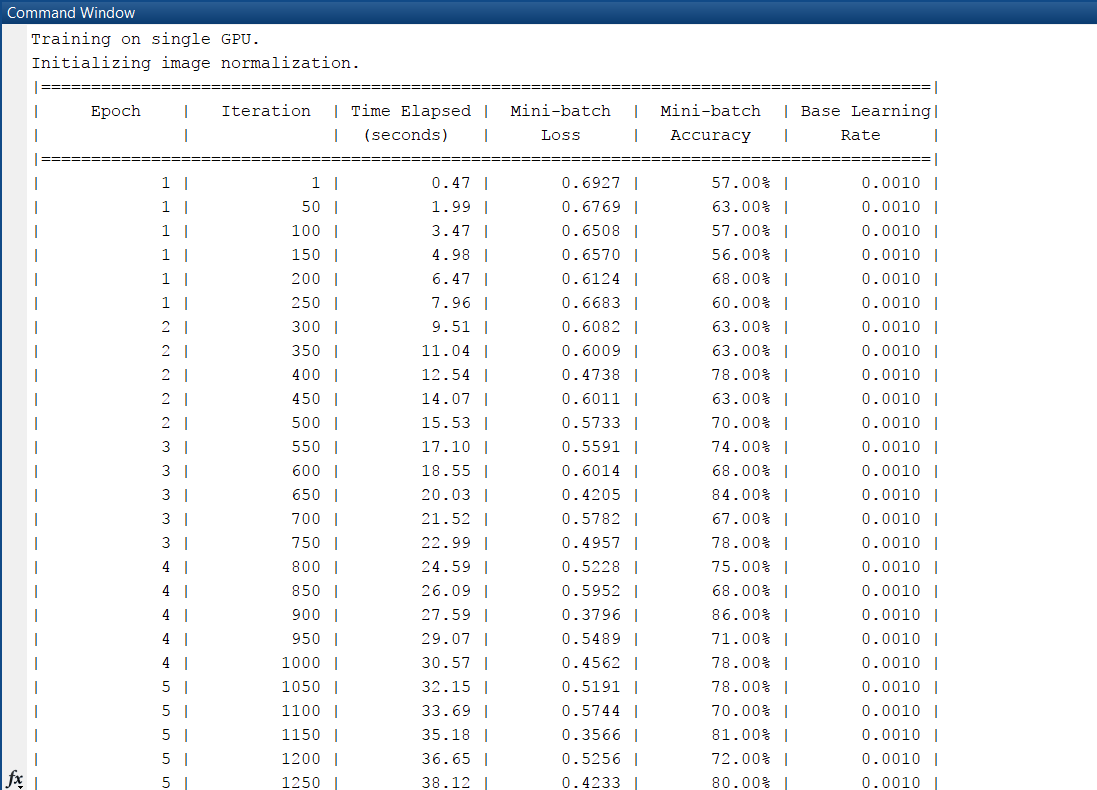
Note: At first had a bad GPU, an old computer and couldn’t perform the computer CUDA computations to train.

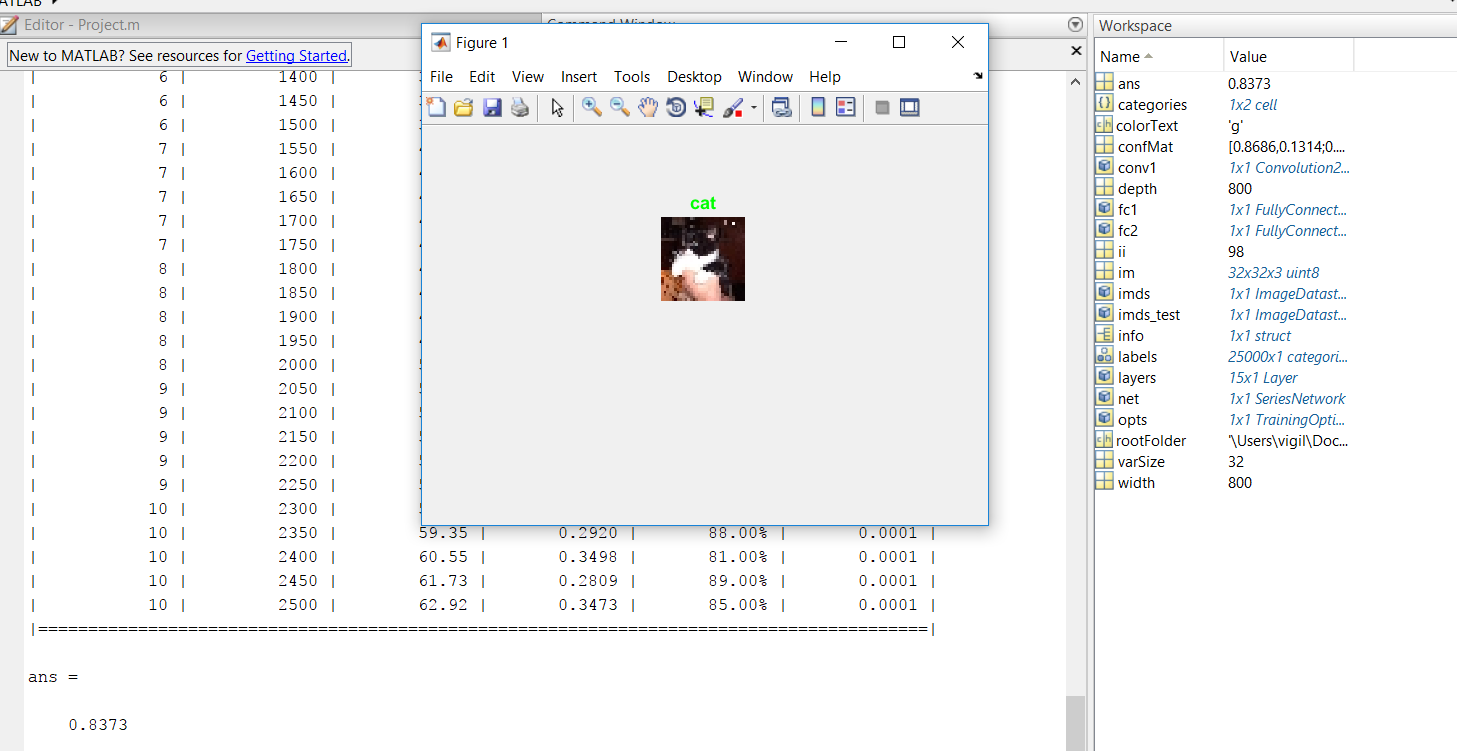


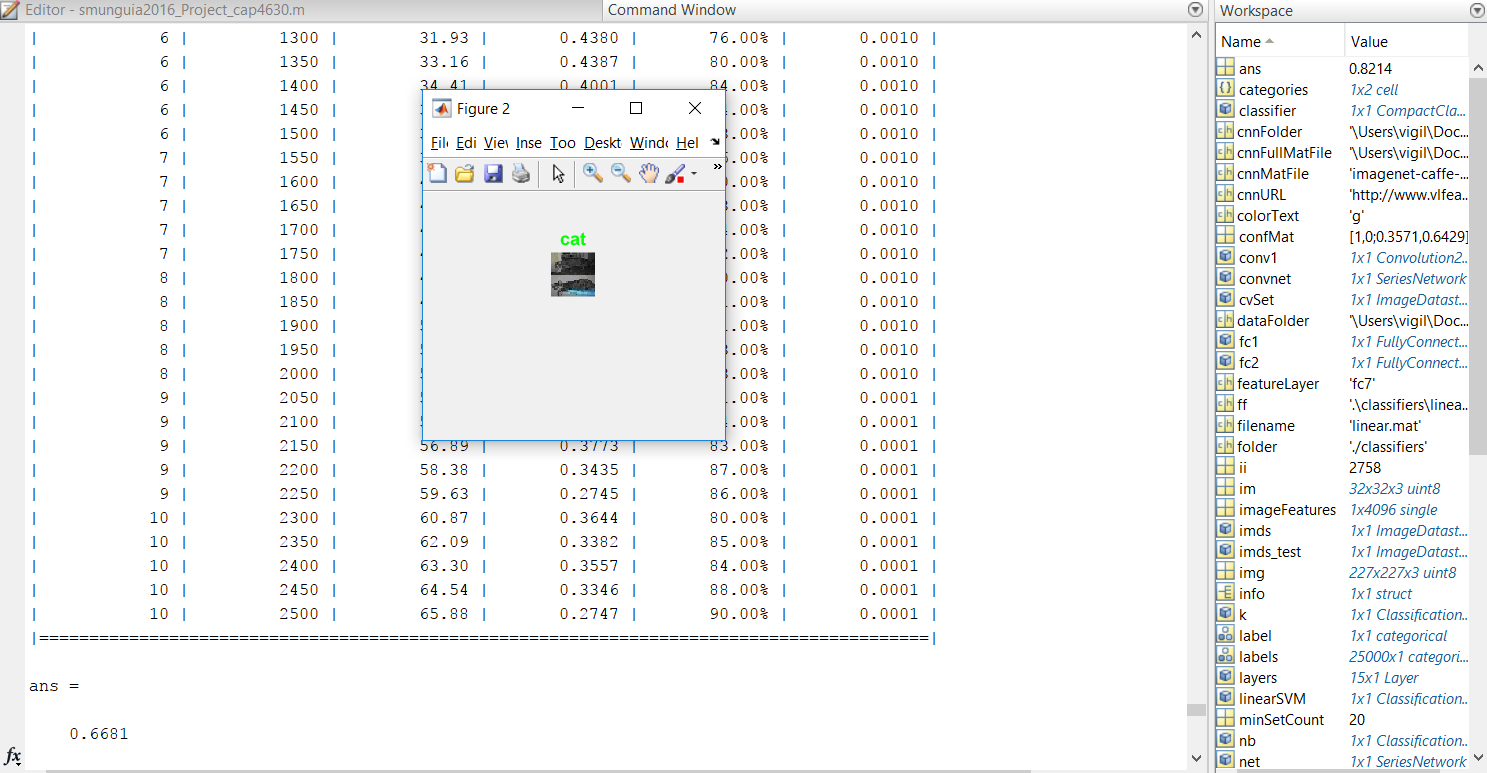
So, I went to Best Buy and purchased a new computer…

Here I used an NVIDIA GeForce GTX 1060 on an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 2801 Mhz, 4 Core(s), 8 Logical Processor(s) to train a CNN from of the 25,000 dogs and cats images data set from Kaggle.

, Then tested it on a new test set image.







Here we used the pre-trained CNN from AlexNet and compared Classifiers



