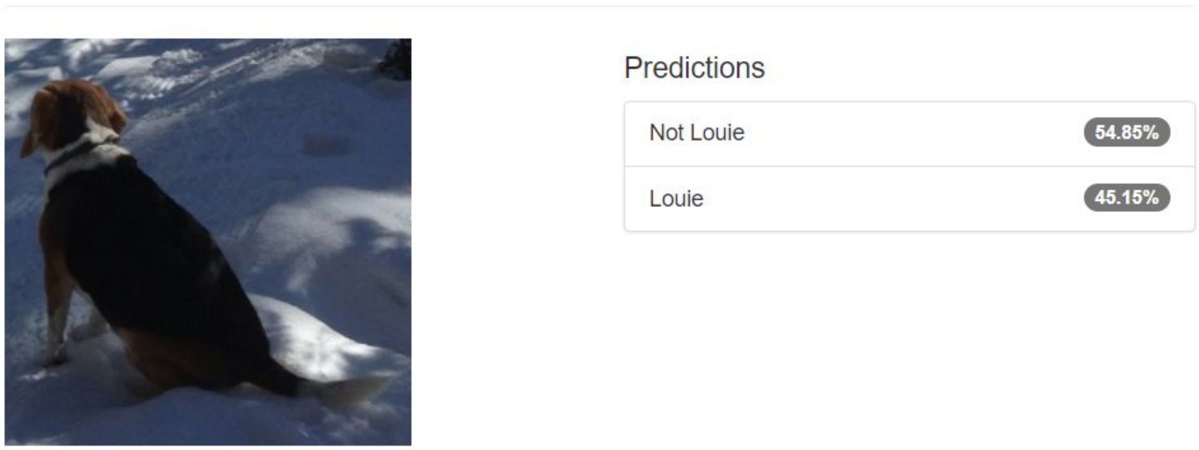
Assignment 2 Part 1
Summary of NVIDIA Fundamentals of Deep Learning for Computer Vision!

GPU Task 1: Image Classification

Training a neural network to classify images. Here there are two classes Louie and Not Louie.

Concept: Train with a single epoch vs multiple epoch.

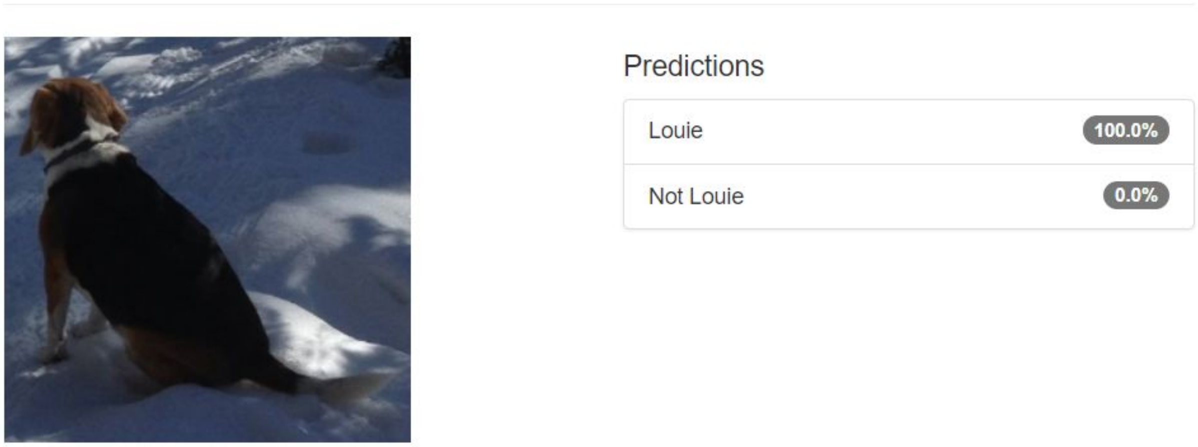
After 2 epochs: loss = 0.682632



Predictions

Not Louie	54.85%
Louie	45.15%

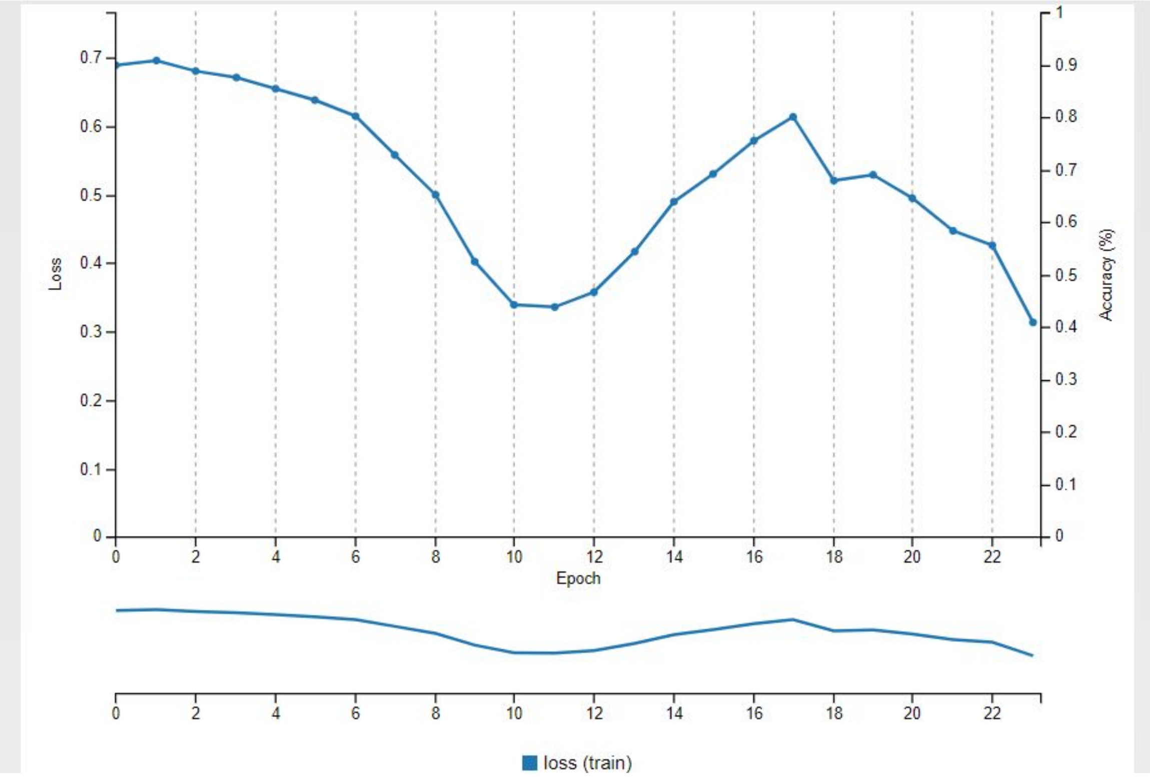
After 30 epochs: The Model recognised that its a pic of Louie 100%:



Predictions

Louie	100.0%
Not Louie	0.0%

The train loss decreased till some point and then there is a fluctuation in the loss. The loss after 100 epochs as follows:



GPU Task 2: Image classification on unseen Data.

In this section, we combine The GPU, Big data, Deep Neural Networks to train a neural network that is effective in the real world. We'll measure our success by the model's performance on *new data*.

In Task 1, the model has trained and performed on same data. In this GPU task, we will see how to measure the *performance of network on new data* while introducing *one* strategy that reduces overfitting, big data.

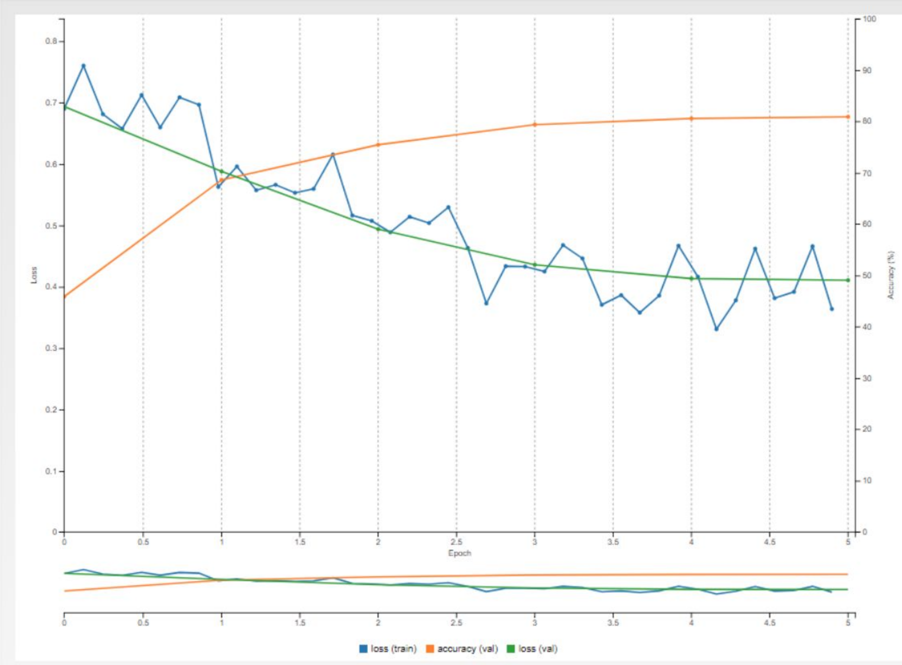
After 5 epochs:

Blue - Loss train

Orange - Accuracy Val

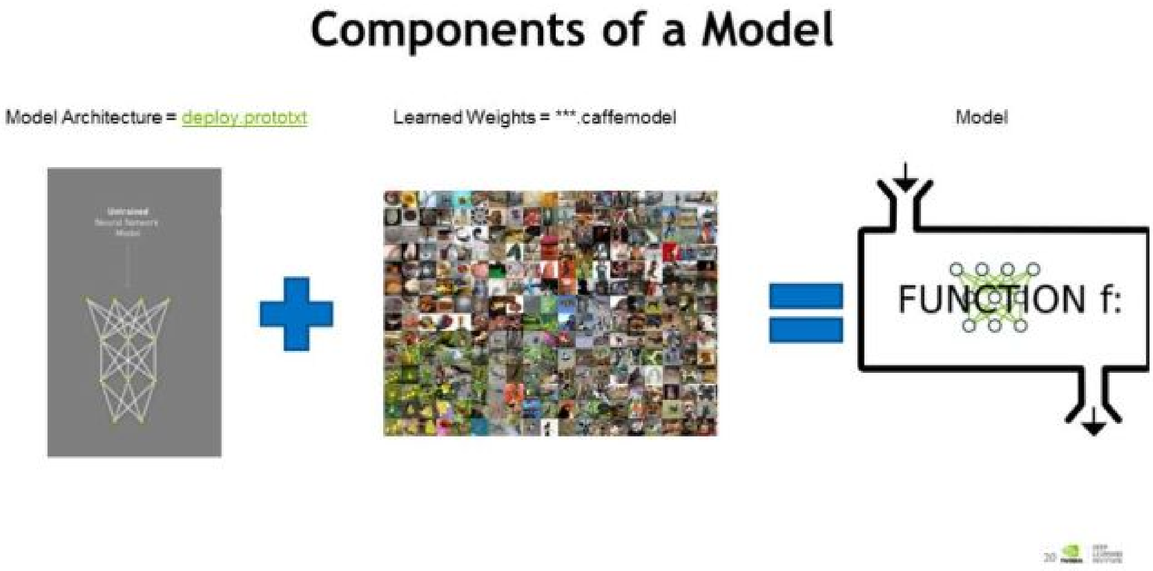
Green - Loss Val

Accuracy and loss plots



From this it is clear that the validation loss decreased with number of epochs and the accuracy went up gradually. (More experiment results in .csv file

GPU Task 3: Deploying trained networks



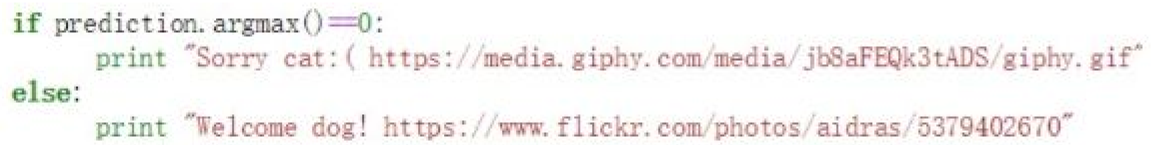
Components of a Model

Model Architecture = deploy.prototxt

Learned Weights = ***.caffemodel

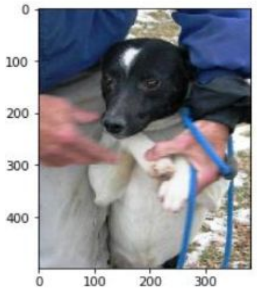
Model

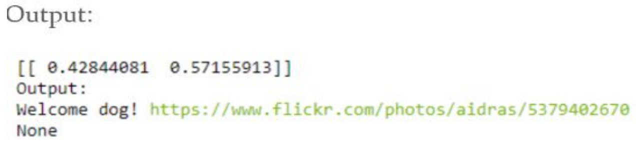
Deploy trained networks into applications to solve problems in the real world. We'll take on a sample simulated project that would have been impossible without the network we just trained.



```
if prediction.argmax()==0:
      print "Sorry cat:( https://media.giphy.com/media/jb8aFEQk3tADS/giphy.gif"
else:
      print "Welcome dog! https://www.flickr.com/photos/aidras/5379402670"
```

The model calculated the probabilities of each class for test images. Instead, of regular output the output is shown as:



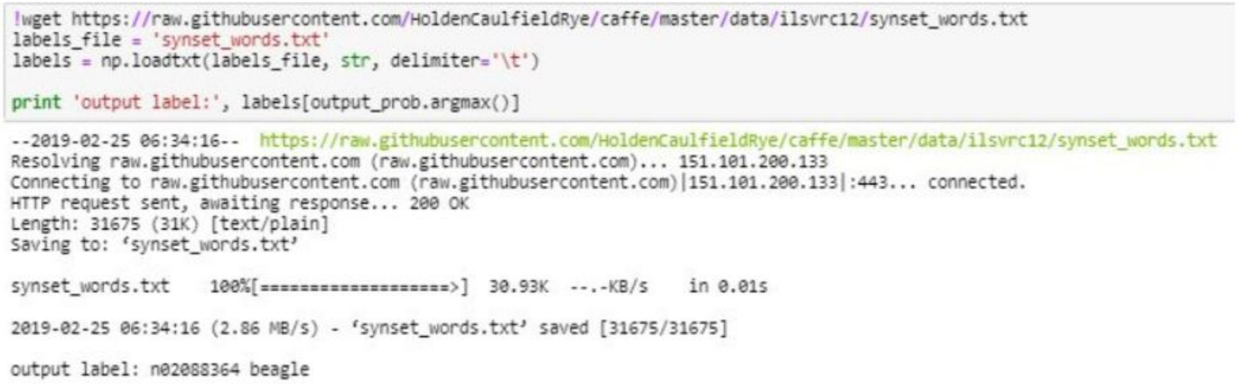


Output:

```
[[ 0.42844081  0.57155913]]
Output:
Welcome dog! https://www.flickr.com/photos/aidras/5379402670
None
```

GPU Task 4: Measuring and Improving Performance

Main goal of this task is to improve the performance of the model created in task 2. Trained the model by increasing the number of epochs to see if that makes a difference. But changing the number of epochs resulted in a very little effect on accuracy of the model(Shown below). Loading the labels and printing the output is as follows.



```
!wget https://raw.githubusercontent.com/HoldenCaulfieldRye/caffe/master/data/ilsvrc12/synset_words.txt
labels_file = 'synset_words.txt'
labels = np.loadtxt(labels_file, str, delimiter='\t')

print 'output label:', labels[output_prob.argmax()]
```

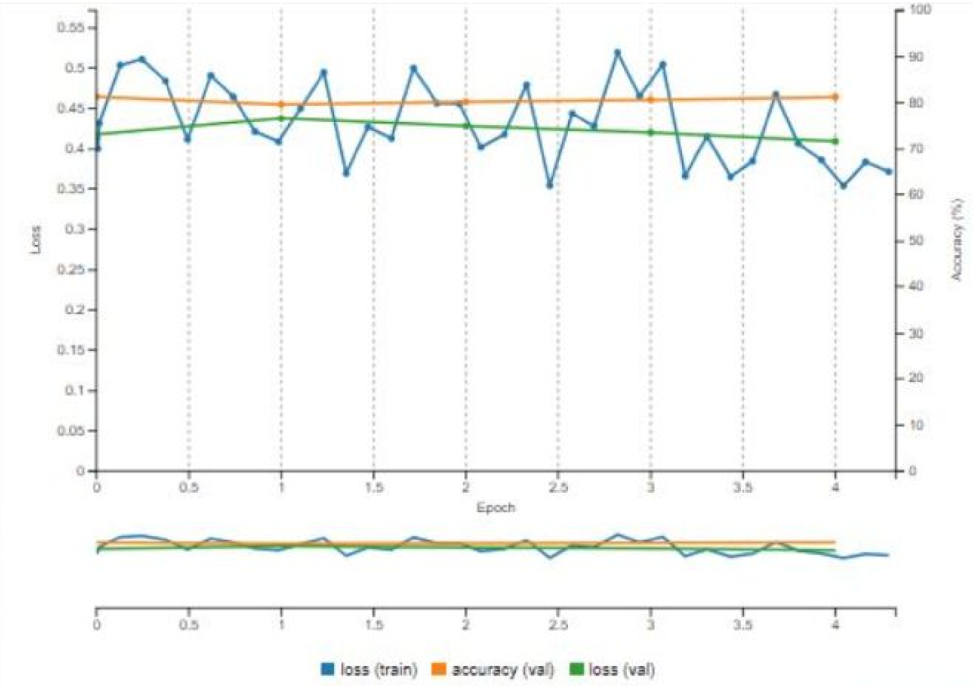
```
--2019-02-25 06:34:16--  https://raw.githubusercontent.com/HoldenCaulfieldRye/caffe/master/data/ilsvrc12/synset_words.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.200.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.200.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31675 (31K) [text/plain]
Saving to: ‘synset_words.txt’

synset_words.txt    100%[===================>]  30.93K  --.-KB/s    in 0.01s

2019-02-25 06:34:16 (2.86 MB/s) - ‘synset_words.txt’ saved [31675/31675]

output label: n02088364 beagle
```

Then used the pretrained Alexnet model on this dataset . Result as follows



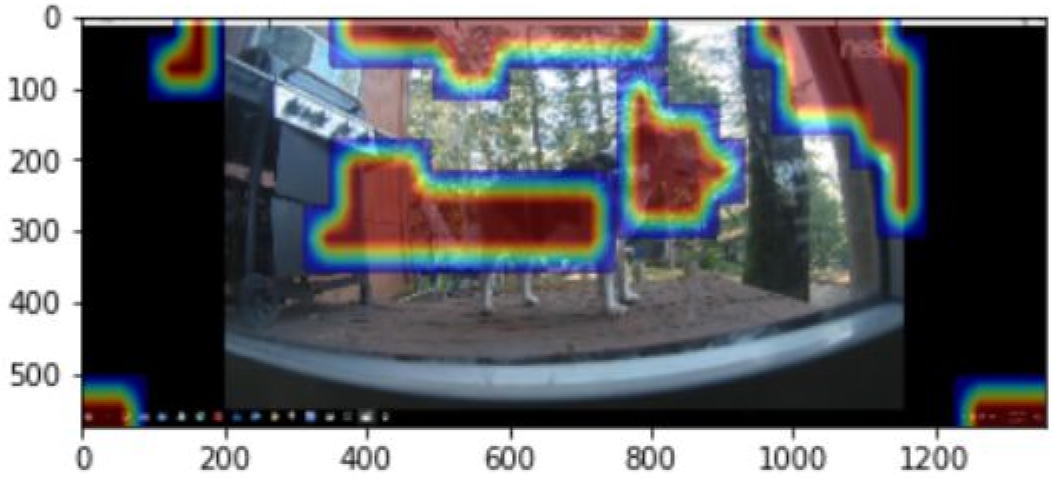
GPU Task 5: Object Detection

Goal is to detect objects and localize them within the image. Created a method using slider window method and then I used the Alexnet and converted it into a fully connected CNN. This is done in two steps.

Step 1: Changed the 6th layer to be convolutional and then connected it to previous and next layer to make sure of data flow.

Step 2: Then changed the 7th and 8th layers to CN layers.

The model using Alexnet performed better compared to the regular model.



Experiment results in Sheet.