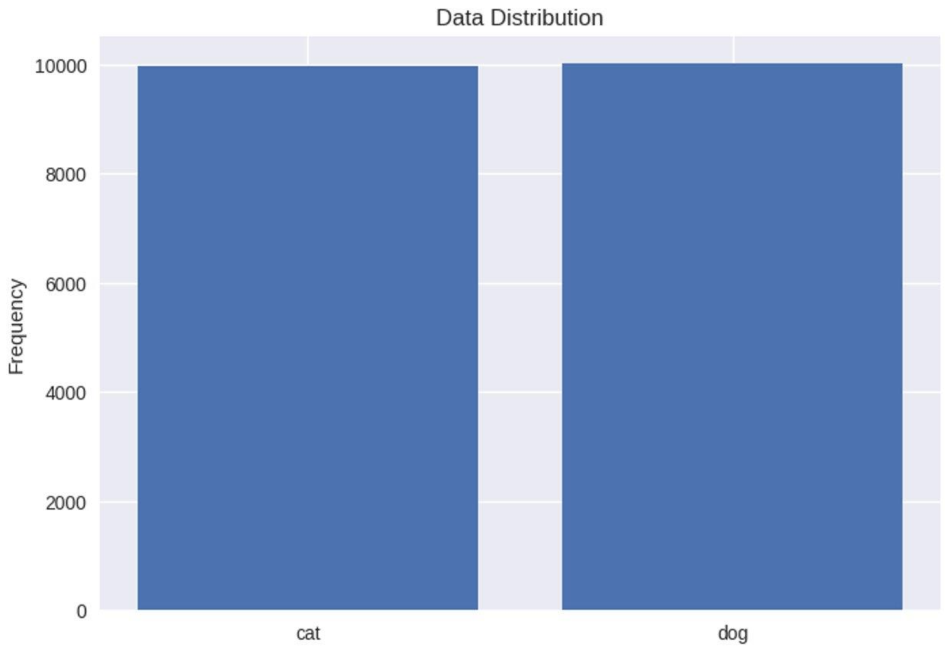
Project Report

INTRODUCTION:

This project implements the theory behind the information bottleneck principle, which states that neural networks experience a bottleneck of information while training. They learn how to be efficient with the information they want to store to forget about becoming efficient at learning the data. I trained a neural network on cat and dog data and then, I used the layers of the neural network to show how different layers extract features from which another model, like an SVM, can learn.

DATA ANALYSIS:

As you can see below, the data is very well distributed into two classes. So, the chances of bias in the deep learning model are very small.



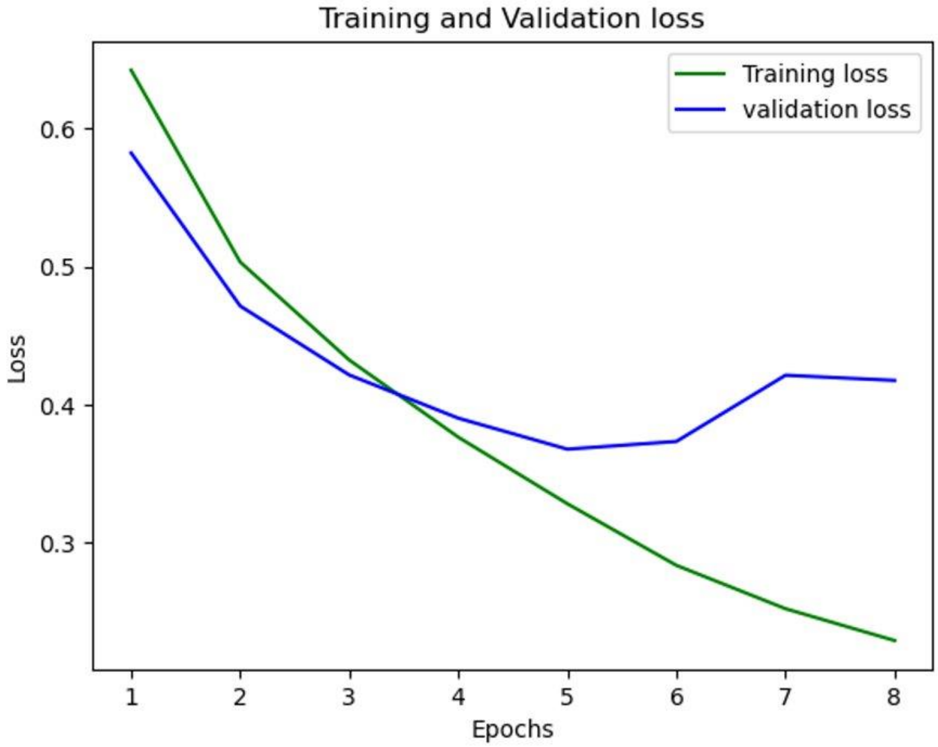
PROCEDURE:

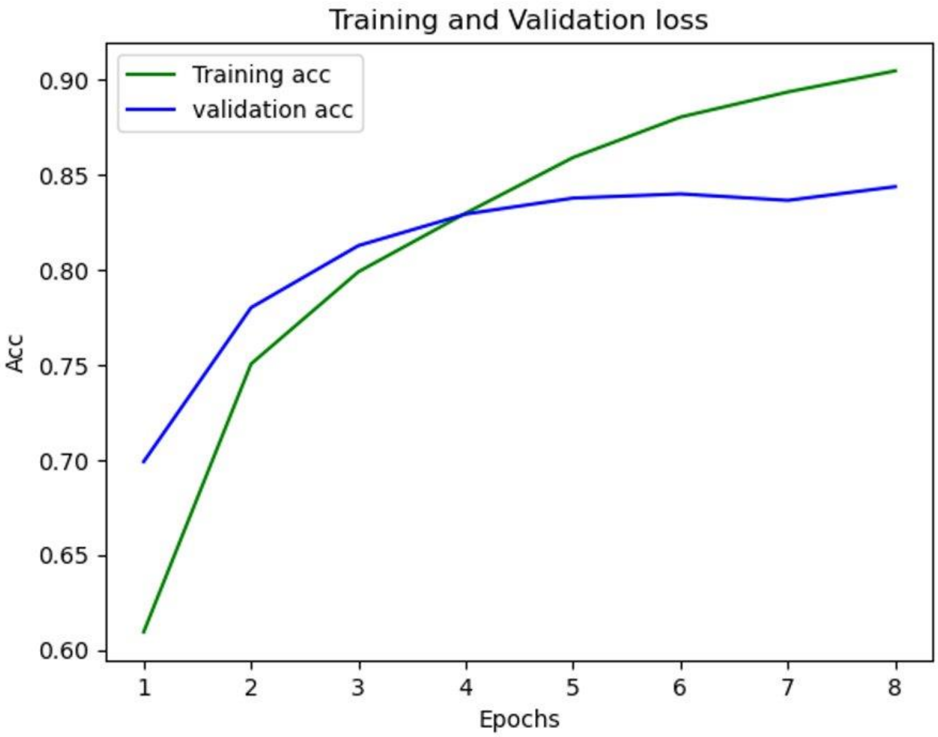
I have trained two models here. One is a neural network with 3 CNN layers and 3 fully connected layers. The last FC layer is the classification layer with 2 outputs, so I will not extract features from there. I will use the three CNN and 2 FC layer features to train the next ML model, for which I used SVM.

The hyperparameters for the Neural Networks are:

Optimizer is Adam with learning rate of 0.001 and cross entropy loss

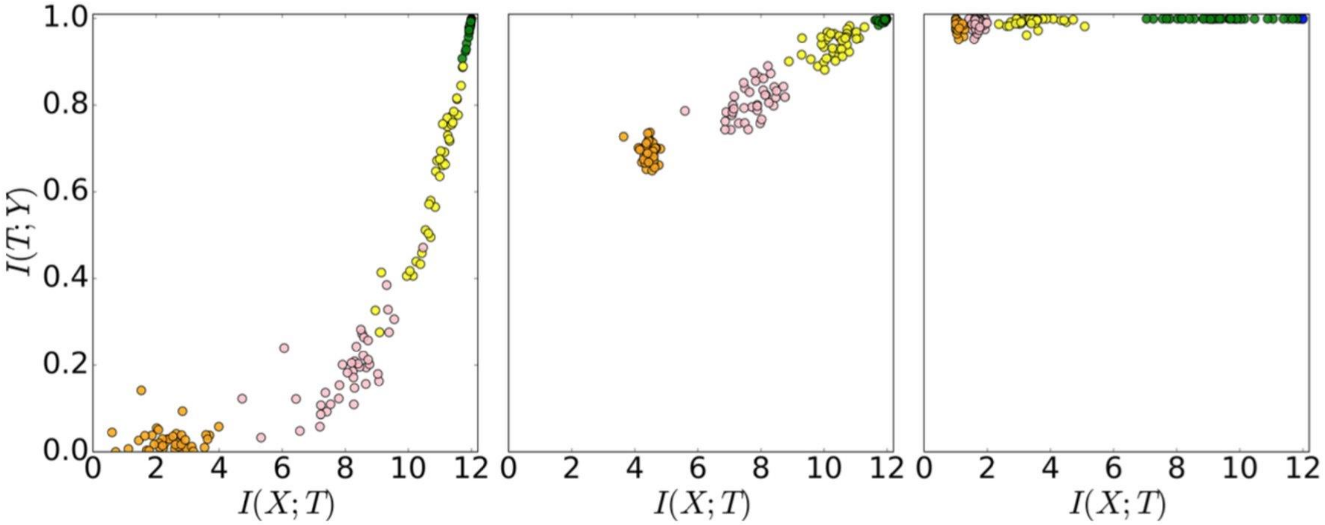
As you can see from the plots, the model is trained well with some chances of overfitting, but it stopped before it could overfit





INFORMATION BOTTLENECK THEORY:

The theory, in some easy terms without going into the maths, states that the layers of neural networks while training experience a bottleneck of information while training. They must see if they can store or forget information so that the model can learn well from the data. Because of this, some layers experience "forgetting" about the input data and learning about the output. Since the layers can store some limited quantity of data, initial layers tend to store information about the entire process while deeper layers tend to store information exclusively about the output. The plot below shows what happens as we proceed with the training



(Image source: Shwartz-Ziv and Tishby, 2017)

In the plot above, green dots represent the initial layers and hidden layers are represented as yellow and pink. Orange dots are the final layers. The x axis is the mutual information between the input and encoded data. The y axis is the mutual information between the encoded data and the output. Mutual Information, in simple terms, means how similar the distribution is. So, 1 means exactly similar and 0 means not similar at all. So, as you can see, the initial layers tend to store information about both the input and output. But the deeper layers exclusively store more information about the output. This will be proven using SVM.

DEEP LEARNING MODEL FEATURES:

The deep learning model extracts features from the input data and they look something like this:

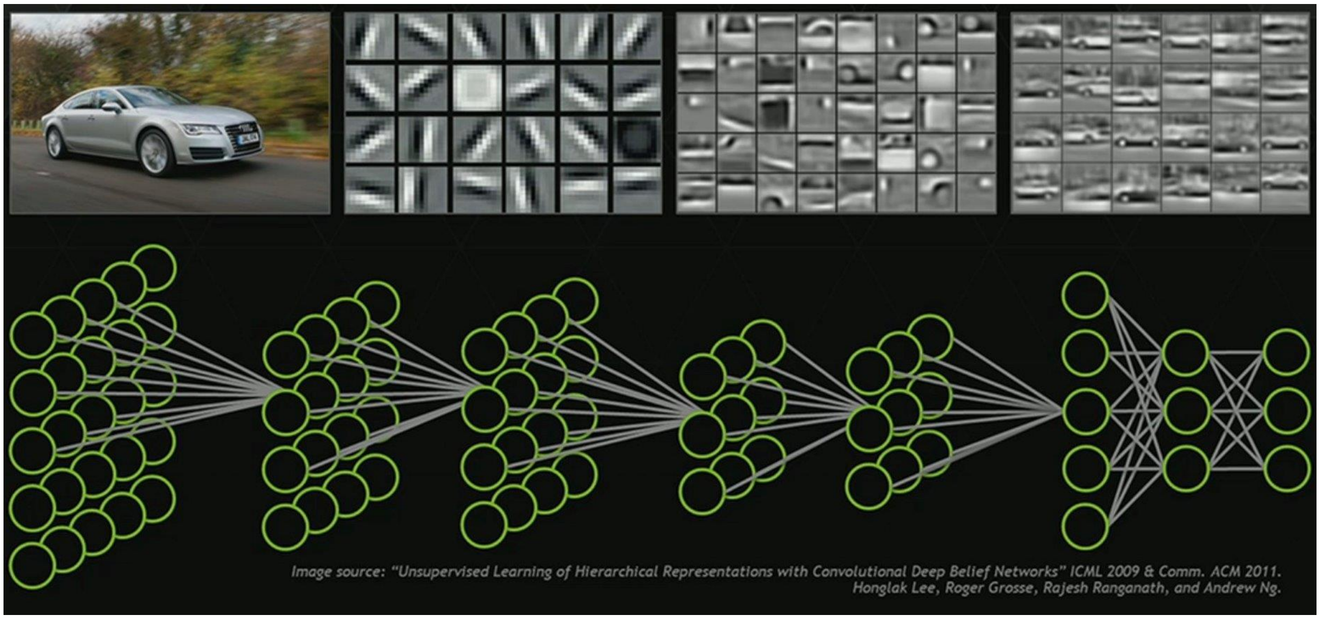


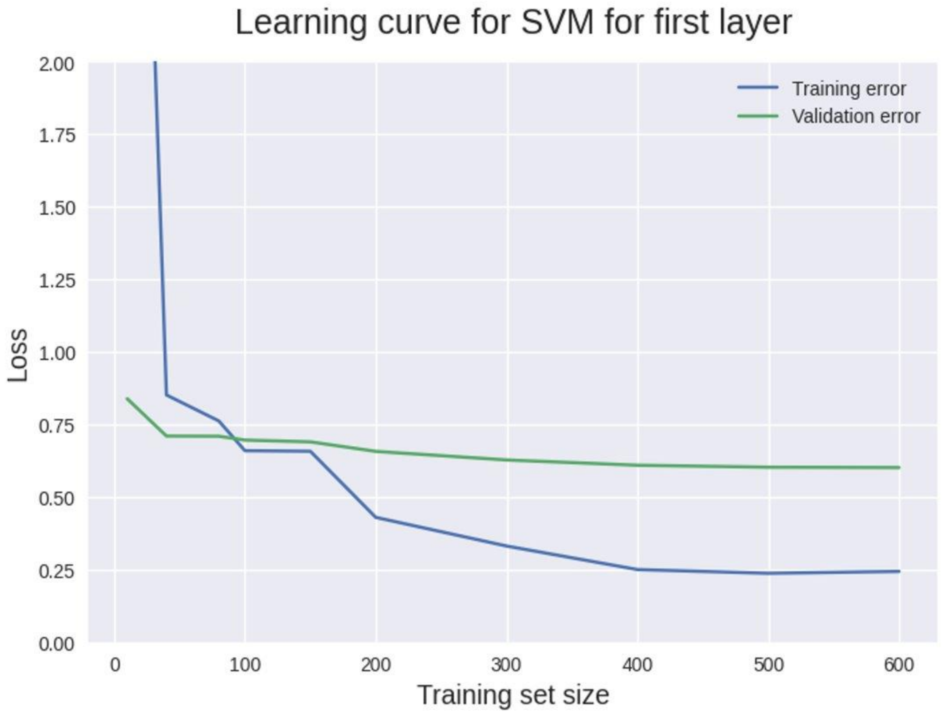
Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011.
Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

Source: Representation Learning

SVM TRAINING:

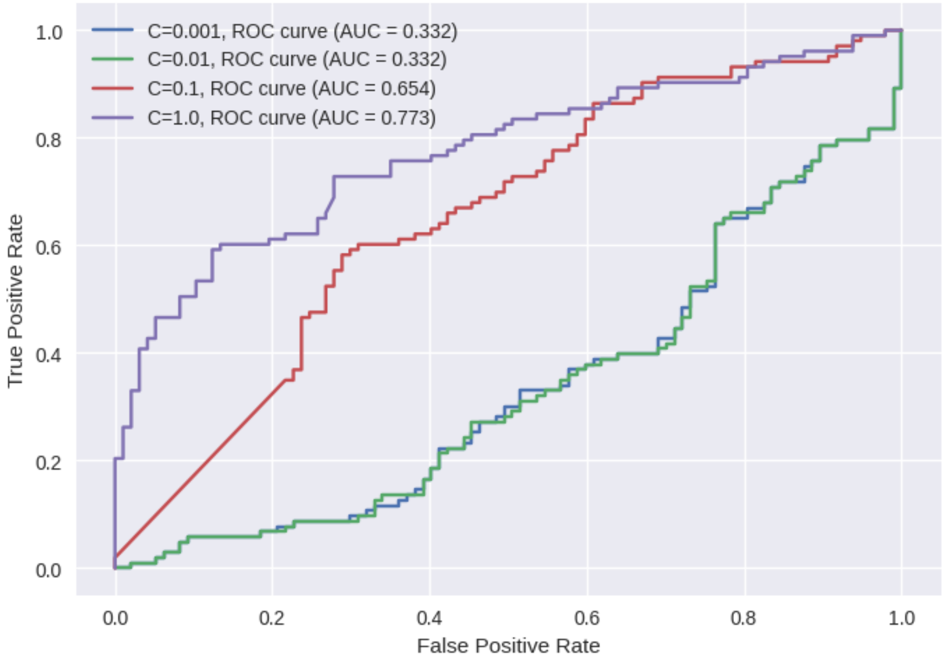
After extracting features from the CNN and non-out fully connected layers, I train an SVM for different regularization parameters and this is what I got:

For 1st CNN feature:

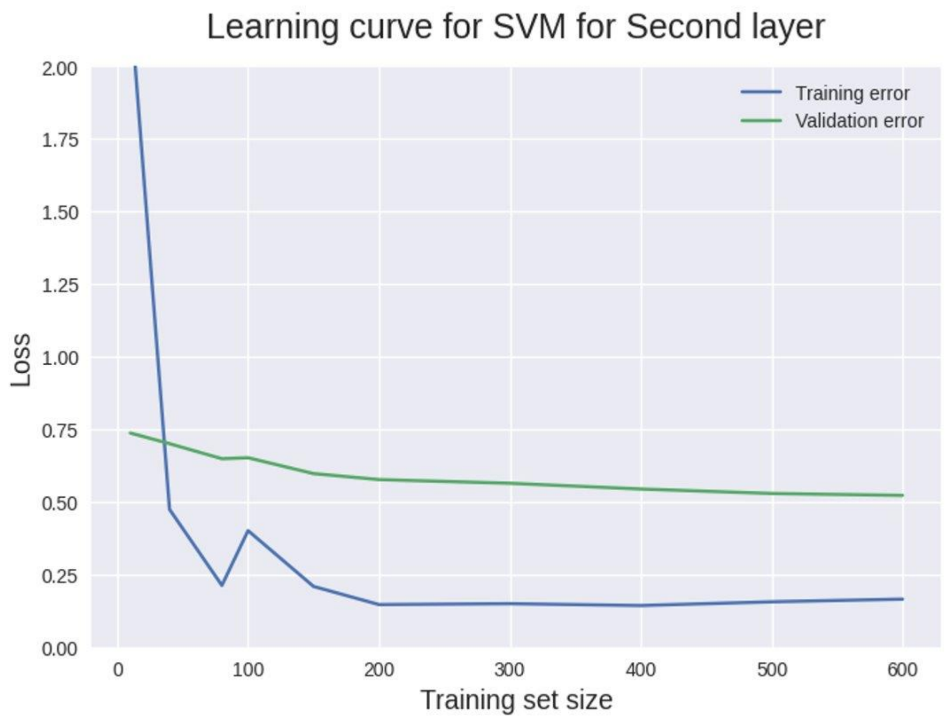


The training is neither the best nor the worst but as we can see the AUC score and ROC curve, we see that for C = 0.001 and 0.01, it learns worse than just guessing. For C = 1 is the best result.

The size of features is 16x55x55 or 48400 features

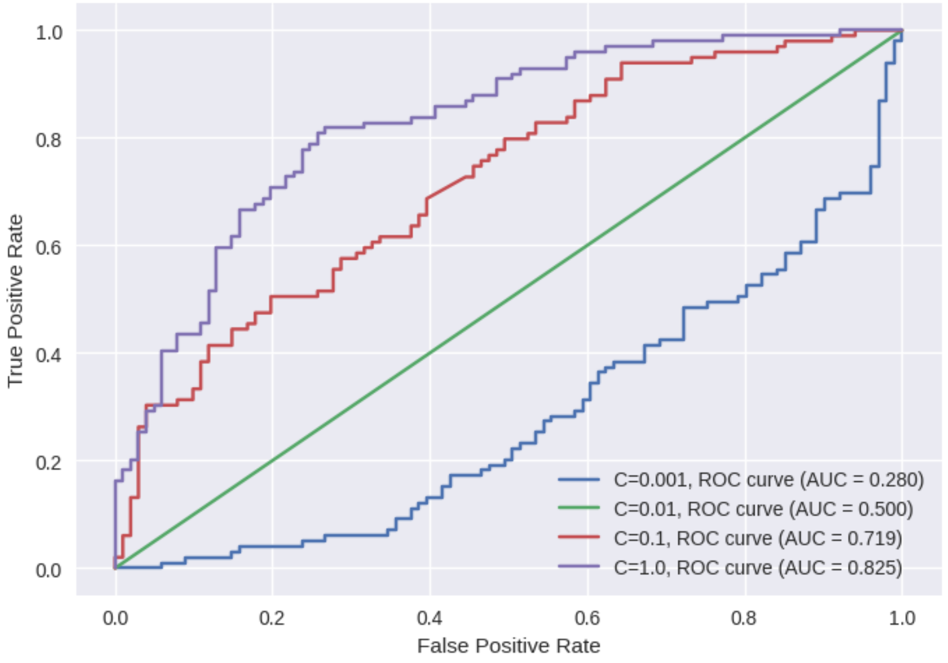


For 2nd CNN feature:

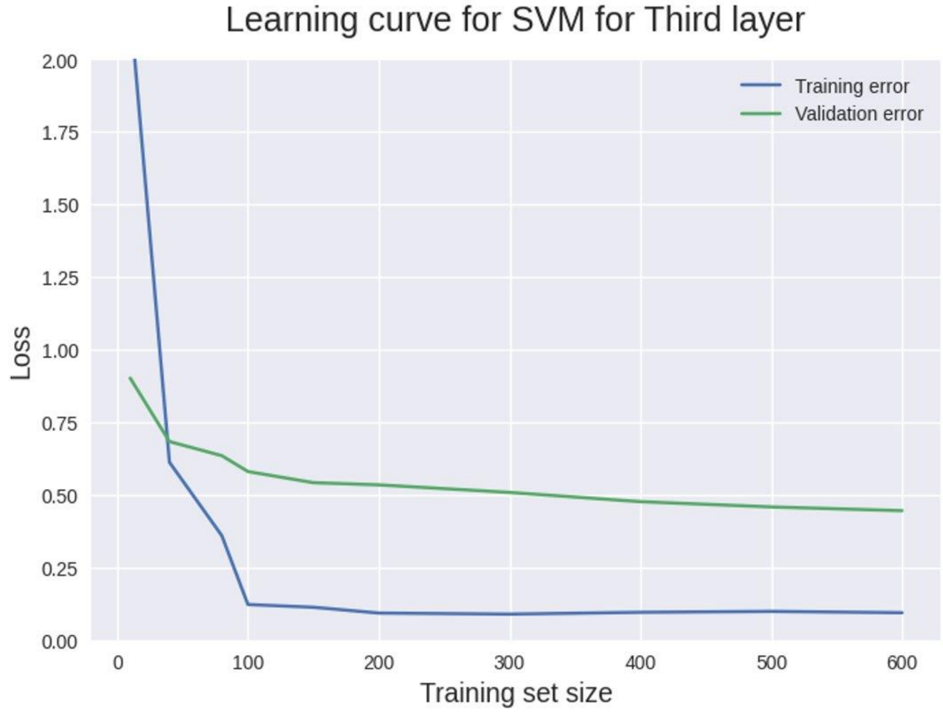


The training is again neither the best nor the worst but even with just 5408 features, the SVM isn't performing well enough. As we can see from the AUC score and ROC curve, for C = 0.001 and 0.01, it is worse than just guessing. So, this means that these features do not contain much information about the input.

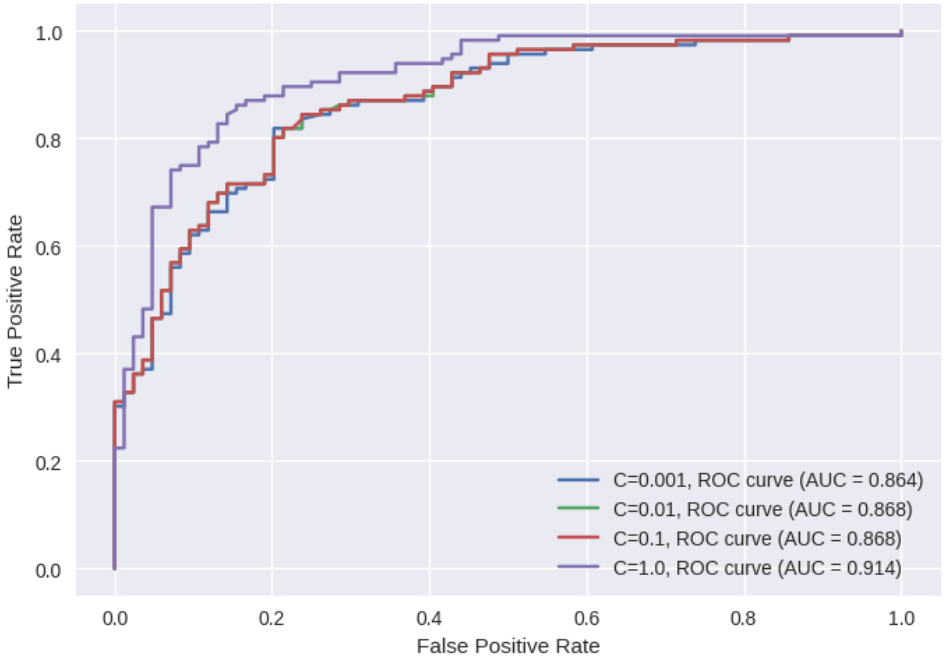
In the first layer, it could be possible that due to high features, the SVM is not training well. But even with 10 times less features, these features don't contribute much to the learning of SVM



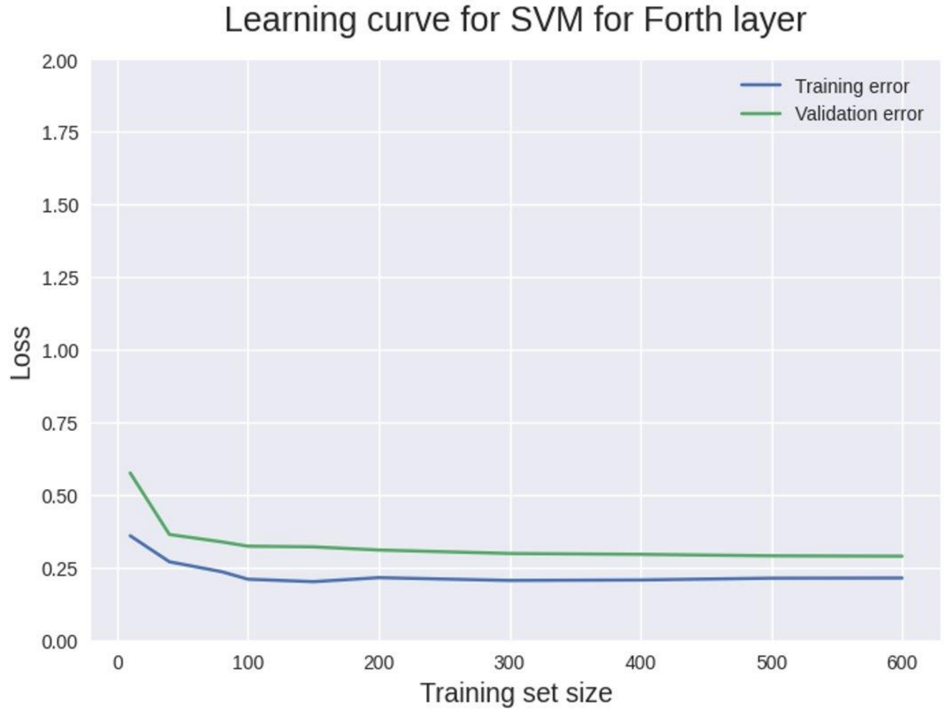
For 3rd CNN feature:



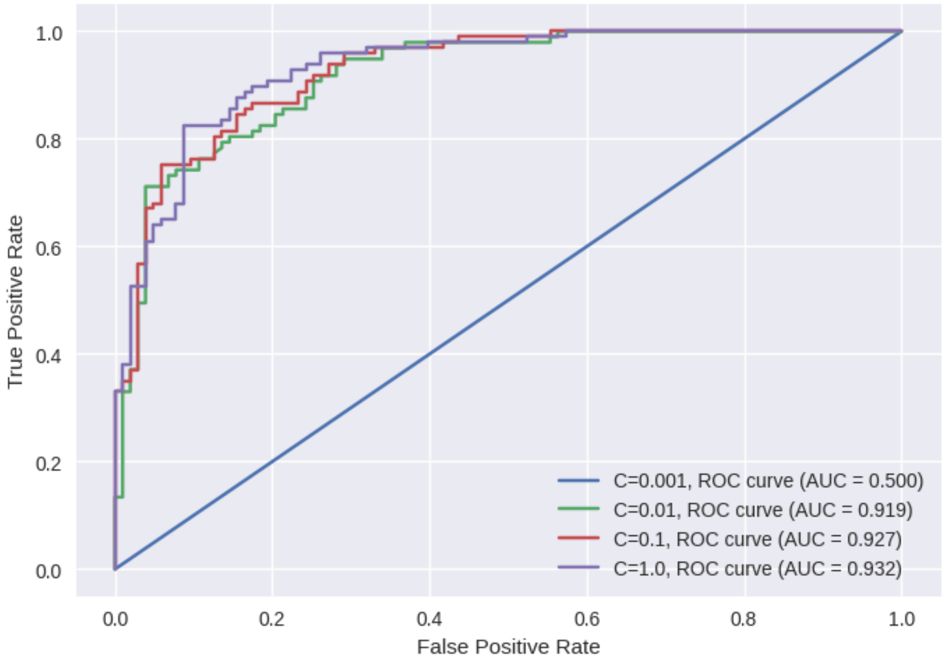
Now, we start to see some improvement in the training of the SVM. The total number of features is 2304 and these are good features from which the SVM can learn. The AUC score is > 80% for all test cases



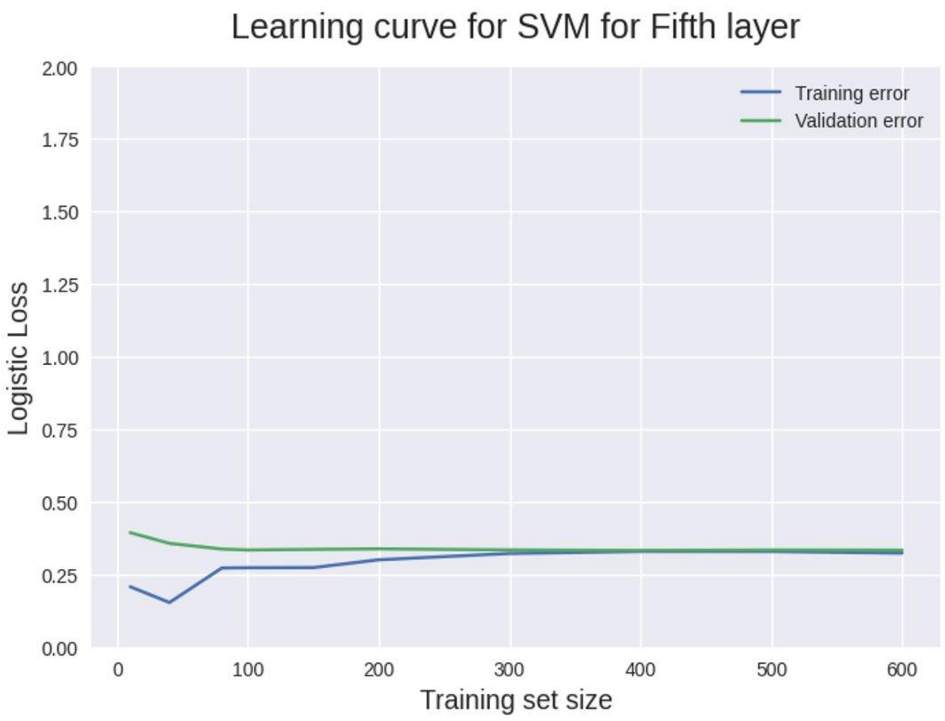
For 4th Layer or 1st FC layer feature:



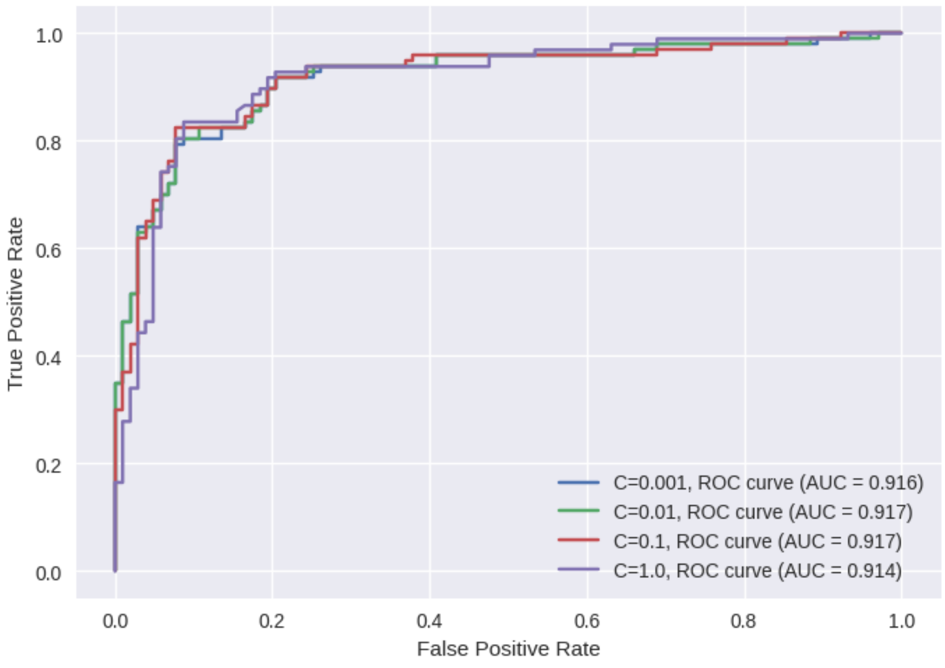
We see a further improvement in the learning curve and the AUC score. The number of features is 500



For 5th layer or 2nd FC layer feature:



These 50 features are best SVM got, and the best learning was achieved for this layer



CONCLUSION:

The information bottleneck theory is proven through this experiment. Since this was a small neural network and I was using SVM, the data is not as concrete as the original authors’, but it is proof enough to show the quality of features extracted by a neural network at each layer.