In this project convolutional neural network (CNN) has been used in different ways for classifying dogs and cats. Each part is explained separately and has a separate matlab code. The file **Main_milestone3.m** can be used to run all the parts.

I. Warm up

In the first part we run the matlab code **"warmup_milestone3.m"** for a small dataset of images.The dataset has two classes of 'dog' or 'cat' and contains 20 images of each category. We use the original alexnet and store it in a variable convnet. The layers of this pretrained CNN is shown in the figure below. This network is designed for classifying 1000 class of obejects and has been trained over one million images.We use the network to extract features of our dataset images.In order to do that,we import our dataset images which contains 20 cat images and 20 dog images into a datastore and then split them into a training set and test test using command *splitEachLabel*. The code assigns 12 images of 40 images randomely to the training set and 28 remaining images to the set set. The code *readAndPreprocessImage* is used to open images from the datasetore when they are needed by doing a preprcessing operation on them as they are being read. Since alexnet requires RGB images of size 227 by 227,the *readAndPreprocessImage* code changes the images to be similar to RGB by stacking 3 replication of the image on each other if it is a grayscale image. It also resizes the image size to 227 by 227.

```
  1   'data'     Image Input                  227x227x3 images with 'zerocenter' normalization
  2   'conv1'    Convolution                  96 11x11x3 convolutions with stride [4  4] and padding [0  0  0  0]
  3   'relu1'    ReLU                         ReLU
  4   'norm1'    Cross Channel Normalization  cross channel normalization with 5 channels per element
  5   'pool1'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0  0  0]
  6   'conv2'    Convolution                  256 5x5x48 convolutions with stride [1  1] and padding [2  2  2  2]
  7   'relu2'    ReLU                         ReLU
  8   'norm2'    Cross Channel Normalization  cross channel normalization with 5 channels per element
  9   'pool2'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0  0  0]
 10   'conv3'    Convolution                  384 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
 11   'relu3'    ReLU                         ReLU
 12   'conv4'    Convolution                  384 3x3x192 convolutions with stride [1  1] and padding [1  1  1  1]
 13   'relu4'    ReLU                         ReLU
 14   'conv5'    Convolution                  256 3x3x192 convolutions with stride [1  1] and padding [1  1  1  1]
 15   'relu5'    ReLU                         ReLU
 16   'pool5'    Max Pooling                  3x3 max pooling with stride [2  2] and padding [0  0  0  0]
 17   'fc6'      Fully Connected              4096 fully connected layer
 18   'relu6'    ReLU                         ReLU
 19   'drop6'    Dropout                      50% dropout
 20   'fc7'      Fully Connected              4096 fully connected layer
 21   'relu7'    ReLU                         ReLU
 22   'drop7'    Dropout                      50% dropout
 23   'fc8'      Fully Connected              1000 fully connected layer
 24   'prob'     Softmax                      softmax
 25   'output'   Classification Output        crossentropyex with 'tench', 'goldfish', and 998 other classes
```

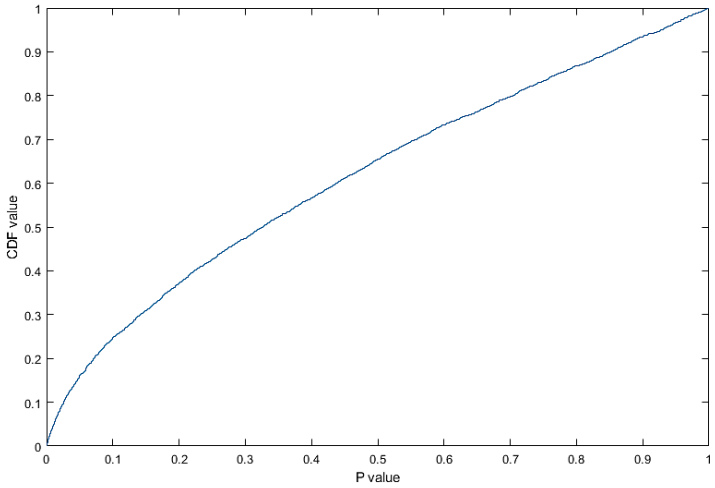
```
ClassificationOutputLayer with properties:

         Name: 'output'
   ClassNames: {1000×1 cell}
   OutputSize: 1000

 Hyperparameters
  LossFunction: 'crossentropyex'
```

After defining the training set,we use fully contect layer 'fc7' to extract feature of the trainig images. The features are extracted using the command activations and are stored in the matrix trainingFeatures. The

numeber of features are 4096. We use t-test which is a simple filter-based feature selection method in order to select discrimanting features for the classificatin task. After using t-test with the command ttest2 we plot the empirical cumulative distribution function (CDF) of the p-values which is shown in figure below. This plot shows that only a very small fraction of the features have near zero p-values.Therefore, we select different number of features each time for the classification using a multiclass SVM classifier.

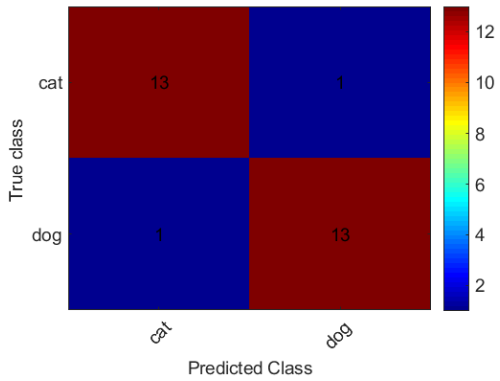


After evaluation of the classifier on the test set,we find the following performances.

1.Classification using top 100 features:

accuracy = 92.86%

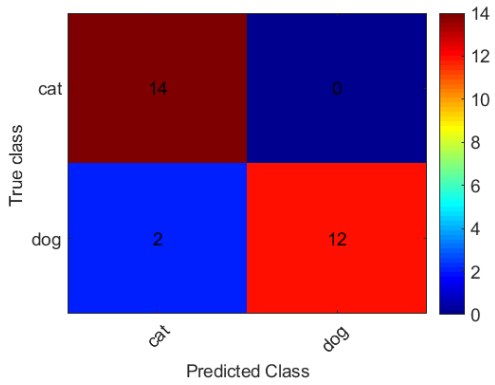
Doge was classified as cat.



2.Classification using top 29 features:

accuracy = 92.86%

Doge was classified as dog.



3.Classification using top 5 features:

accuracy = 89.29%

Doge was classified as cat.

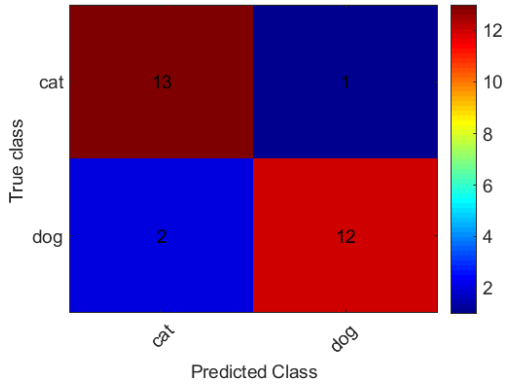


Figure below shows doge which can be classified as dog or cat depending on the model.

By selecting different number of features it was shown that the top 29 feature can produce the highest accuracy on the test set (92.86%). Choosing higher number of features such as 2500 features still produce accuracy of 92.86%.

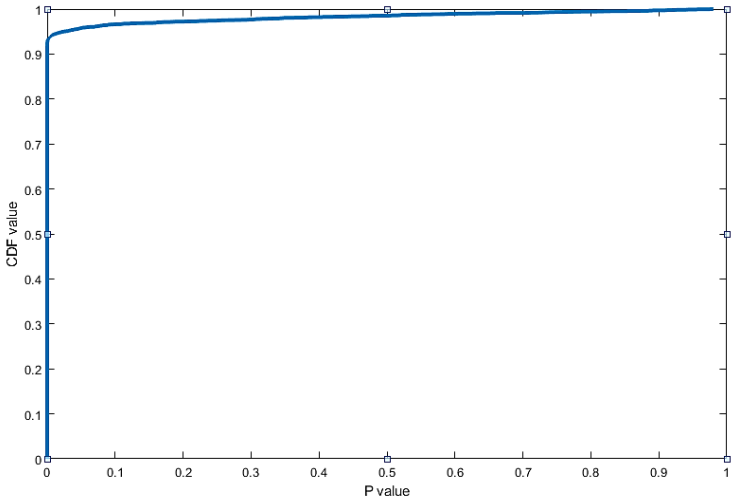
Number of selected features	Test accuracy
2500	92.86%.
100	92.86%.
29	**92.86%.**
5	89.29%

II. A pretrained CNN model (e.g., AlexNet) in which an appropriate (fullyconnected) layer's activations are used as features, which are then used by a conventional machine learning classifier(Baseline) (Part C)

The matlab code for this part is **"Milestone3_feat.m"**. The dataset has two classes of 'dog' or 'cat' and contains 25000 images of each category. We split the dataset into a training set,a validation set and a test set. 60 % or 15000 images for training, 20% for validation and 20% for the test set.

We run the code and extract the features of the training images using layer 'fc7' of the alexnet,then we do feature selection and after that we perform a multiclass SVM classifier.

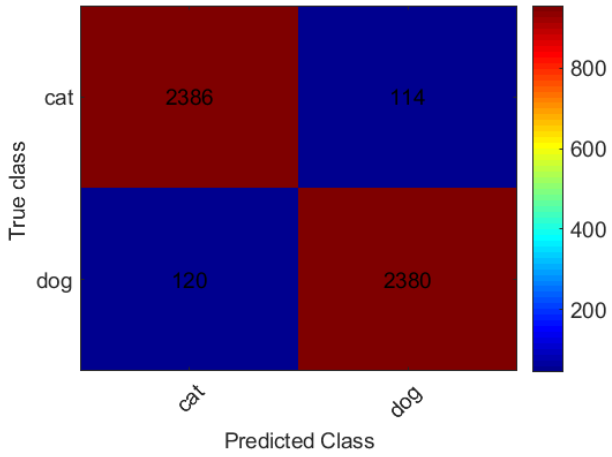
To select a smaller set of important features from 4096 total available features from layer 'fc7',we perform a simple t-test. The plot below shows the empirical cumulative distribution function (CDF) of the p-values. We can see that the p-value of more than 90% of features are very close to zero.



We build an SVM classifier models using different number of features and we compute the accuracy of each on the validation set which is shown in the table below.

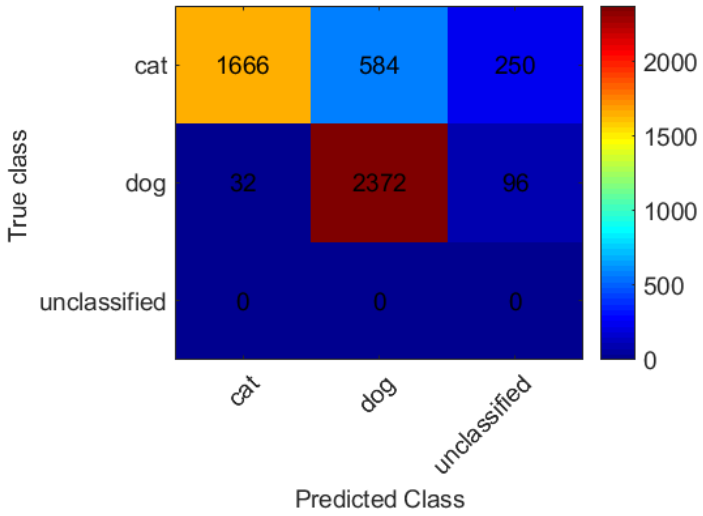
Number of features	Validation Accurcy
4000	96.9%
200	96.6%
100	95.9%
10	89.0%

Since we are looking for a fast and accurate model we choose 200 features to build the final model. We train multiclass SVM classifier using a fast linear solver using 200 features. The accuracy of the model on the test set is **95.3%** and the confusion matrix for the test set is shown in the following figure.



III. A pretrained CNN model (e.g., AlexNet) used *as is* (Part A)

The matlab code for this part is **"Milestone3_asis.m"**. Alexnet was used without any modification to classify the dog/cat test set containing 5000 images. The model classified dog and cat specificly by the type of cats or dogs. The cats.txt and dogs.txt files contain all the types of cats and dogs that were used in alexnet. After the classifcation the general category(dog or cat) for each output was obtained from the text files. The classification accuracy was **80.76%.**The following figure shows the confusion matrix for the test set. 250 of the cat images were classified wrongly as other objects by AlexNet, and 96 of the dog images were classified as other objects.



The following images are ***cat*** images from the test set, which were incorrectly classified by AlexNet.

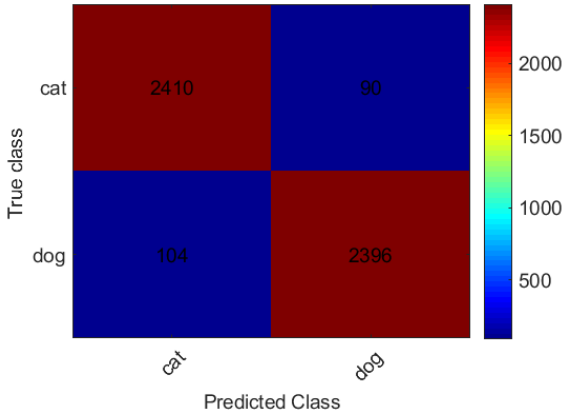
The image on the left was classified wrongly as dog, and the two images on the right were classified as other objects, which were assigned an unclassified label.

IV. A pretrained CNN model (e.g., AlexNet) used under the transfer learning paradigm, i.e., with the proper modifications to one or more layers.(Part B)

The matlab code for this part is **"Milestone3_transfer.m"**. AlexNet was used as the pretrained network and its last three layers were replaced with new layers. The output layer was set to have 2 outputs instead of 1000 classification outputs of the original network and the learning rate factor of the last fully connected layer was set to 20. The modified network was trained on a dataset of dog/cat images. The dataset contain 25000 images and was split into training (15000 images), validation (5000 images) and test (5000 images) sets. The training options were set as InitialLearnRate= 0.000001, MaxEpochs=4, MiniBatchSize=50. In addition, the training was validated using the validation set once per epoch. The network was almost trained after one epoch with the validation accuracy of 95.10%. The accuracy of the network increased slightly for epoch 2, 3 and 4 with the accuracies of 95.86%, 96.10% and 96.18% on the validation set.

After the training completed with 4 epochs, we classify the test set and find the accuracy of the network on the test set. The accuracy was **96.12%.**

In addition the confusion matrix of the test set is:



Summary and conclusion

Different type of classification methods were used to classify a subset of dog and cat images. The SVM model that was created using 200 features from 'fc7' layer of AlexNet was selected as the baseline model. The baseline accuracy was 96.6%. The second model was the original AlexNet with generalization of all types of cats and dogs. This model had the accuracy of 80.76%, which is the lowest accuracy, because the model had 1000 output options. The last model is a retrained model with the layers of AlexNet, and only two outputs. The accuracy of this model was almost as good as the baseline accuracy because the model had two output options and was retrained with cats and dogs images.

While the computer models had accuracy close to 100%, still they are not as accurate as the human vision. Normal human can accurately classify any image from the dataset as cat or dog with accuracy of 100% although they might not be able to determine the type of cat or dog.

Matlab file	Classification method	Test Accuracy
Milestone3_feat.m (**Baseline**)	Extracted Features	96.6%
Milestone3_asis.m	AlexNet as is	80.76%
Milestone3_transfer.m	Transfer Learning	96.12%