Coding Assignment 2: Recognizing Object Cateogories

Manu Agarwal
UTEID: ma53767

1 Introduction

Image classification is the task of classifying images from a dataset into one of several base categories. Several approaches have been proposed for classification. These approaches fall into one of the two categories - unsupervised classification and supervised classification. In this assignment, we explore supervised image classification on natural images containing 25 object categories from ImageNet. We used deep Convolutional Neural Networks for the task. We used the Caffe implementation of the bvlc reference model provided with the standard Caffe distribution followed by fine-tuning the last layer. The original CNN had 5 convolutional layers, some of which are followed by max pooling layers and 2 fully connected layers. We added a fully connected layer at the end for fine-tuning with a final 25-way softmax.
This approach produces a 93.4% accuracy on the test set given out for this assignment consisting of 20 images per each of 25 classes.
We further implemented Histograms of Oriented Gradients (HOG) descriptors followed by classification using SVM and compared it to the performance of the bvlc reference model. The second approach achieved a paltry accuracy of 19.40%. We resized images to 200x200 and then computed HOG features on cellsizes of 4x4.

2 The Basic pipeline-CNN

The basic pipeline for the implementation of the deep learning approach is as follows:

Algorithm 1 Image classification using Fine-tuned CNN descriptors

Input: 25 ground classes with 100 images per class for training and 20 images per class for testing

Output: Classification of the testing images into one of 25 ground classes and a confusion matrix

1: Create files train.txt and test.txt containing training and test image paths along with their labels. We need to make sure that these labels are numerical and start with zero.
2: We shall use the bvlc reference caffe model for this experiment followed by finetuning. Change learning rates of all layers but the last finetuning layer to zero. Set the learning rate of fc8 to 10.
3: Convert the training and test image sets to lmdb format
4: Finetune the already trained CNN using the generated lmdb for the training set
5: Test images from the lmdb for the test set and write to a file the top 5 class predictions for each image along with its true class
6: Generate confusion matrix using the text file generated above

3 Experiments on CNN

We used the training set of 100 images per category given out for the purposes of this experiment for finetuning. We tested our approach on the test set of 20 images per category that was given out. The confusion matrix for the same is shown in Figure 1. The class 'shovel' performed the worst with an accuracy of 0.75 on the 20 test images provided in the test set followed by 'ant, emmet, pismire' and 'sandbar, sand bar' each with an accuracy of 0.8. An accuracy of 1 is achieved mostly on very distinctive classes like 'limousine, limo', 'meerkat, mierkat', 'puffer, pufferfish, blowfish, globefish', 'barber chair' and 'church, church building'.

We observe that classes such as 'limousine, limo' or 'church, church building' did not have another similar class these classes might be confused with. Hence, they must have performed better. Some classes that performed better were very distinctive animals such as 'blowfish'. They have distinct characteristics and are easily distinguishable from other classes. The class 'shovel' got confused with 'mushroom', 'projectile', 'sealion', 'sand bar' and 'measuring cup'. A projectile is similar in shape to a shovel, hence the confusion there. Also, a shovel is many times used for digging sand around sea shores. There might be some images suggesting that which could have got the classifier to interpret them otherwise. The class 'ant' got confused with 'ant', 'measuring cup', 'mushroom' and 'shovel'. This result was strange as these classes do not bear too much of a similarity. The class 'sand bar' got confused with 'lakeside' and 'shovel' which is understandable as sand bars are mostly found around beaches which have water and are similar to lakesides. Moreover, one of the images of sand bar actually contains the image of a girl covering the major part of the image which might have had an adverse effect on the classification of that image. Some of the worst performing images along with their top predictions by the model are shown in Figure 2, Figure 3 and Figure 4.

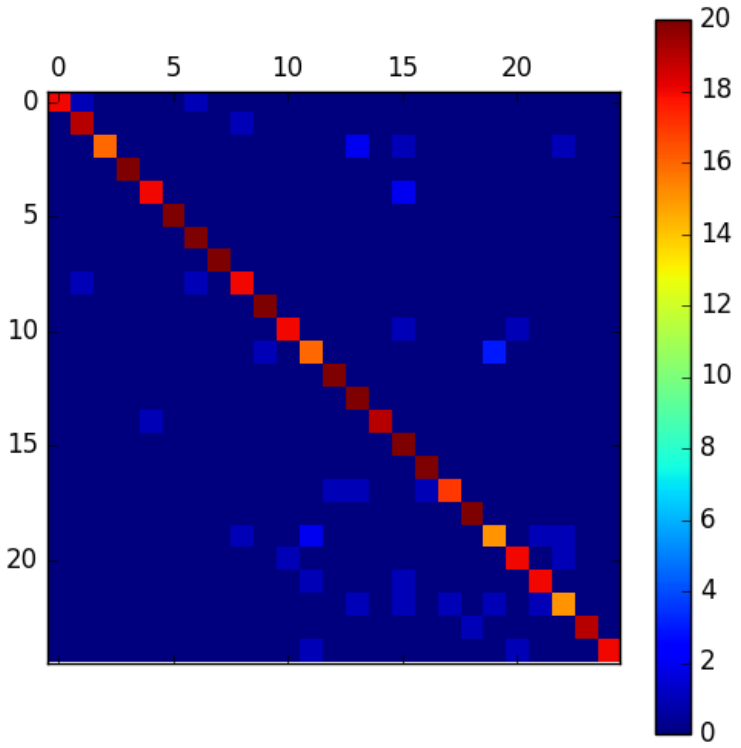

Fig. 1: Confusion matrix for the case when finetuning using just the last layer fc8

(a) (b)

Fig. 2: Some of the images of category 'shovel' that performed bad. The left one was identified as 'projectile' and the right one was identified as 'sea lion'

(a) (b)

Fig. 3: Some of the images of category 'ant' that performed bad. The left one was identified as 'measuring cup' and the right one was identified as 'mushroom'

(a) (b)

Fig. 4: Some of the images of category 'sand bar' that performed bad. The left one was identified as 'shovel' and the right one was identified as 'lakeside'

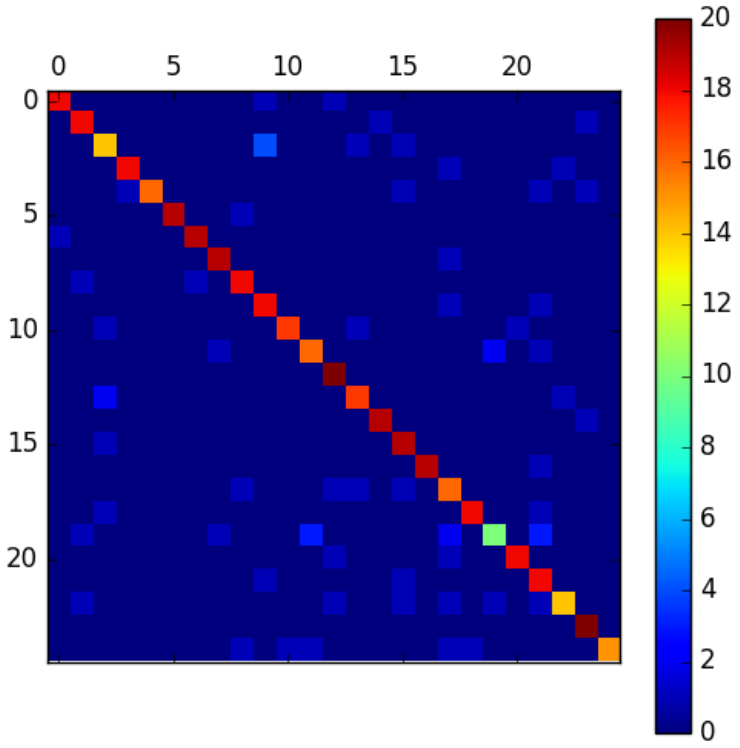

Fig. 5: Confusion matrix for the case when finetuning all the layers

Next, we changed the weights of the layers 1 through 7 to some positive values and tried classifying the test images provided to us. This time we got an accuracy of 86.6% which is less than what we got when we just fine tuned the last layer. This is understandable as setting non-zero weights for the initial layers perturbs the already trained model which was trained on a huge number of images. Finetuning usually improves performs better if applied to the last 1 or 2 fully connected layers. The confusion matrix for this case is shown in Figure 5 We also tried removing the penultimate layer fc7 which gave an accuracy of 87.6%. This shows that the fc7 layer does not contribute significantly to finetuning.

4 HoG with SVM

We now compare our approach with methods which do not employ deep learning. We extract HoG features from images and train a SVM classifier to classify the test images. To make the Hog features of the same size for every image, we rescale images to 200x200. The basic pipeline for this approach is as follows: As expected, this method performed considerably worse than the deep learning

Algorithm 2 Image classification using Fine-tuned CNN descriptors

Input: 25 ground classes with 100 images per class for training and 20 images per class for testing

Output: Classification of the testing images into one of 25 ground classes and a confusion matrix

1: Rescale each image to a size of 200x200
2: Extract HoG descriptors for every training image using a cellsize of 4x4.
3: Train a SVM classifier using the obtained features and the ground truth labels for the training data
4: Use the trained SVM to classify the test images and generate a confusion matrix

method. The accuracy over all classes turned out to be 19.40% which is almost one-fourth of the accuracy obtained by using the bvlc reference model.
The best performing classes were 'church, church building' with an accuracy of 0.45 and 'barber chair' with an accuracy of 0.40. This is again attributable to the distinct characteristics of these classes and the absence of similarly confusing classes. The worst performing classes were again 'shovel' with an accuracy of 0.05 and 'ant' with an accuracy of 0.10. Figure 7 and Figure 8 show the worst performing images. This time there is no significant connection between the predicted labels and the ground truth labels, thereby showing that this method does not perform as well. The confusion matrix for this case also indicates the same. We could further improve the accuracy by resizing the images to a size larger than 200x200 and using a cellsize of 2x2 when extracting HoG features.

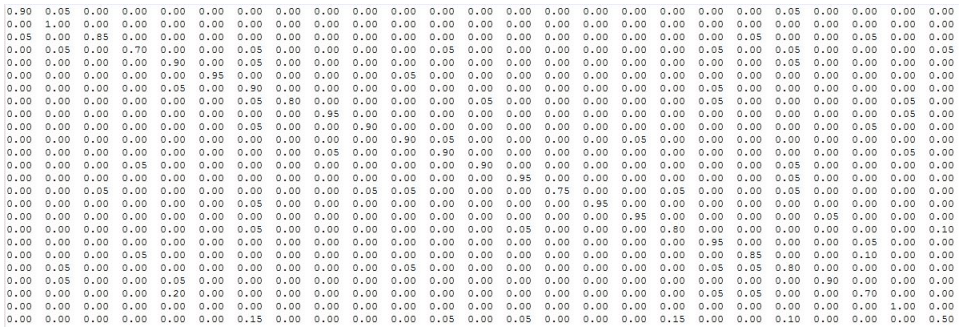

Fig. 6: Confusion matrix for approach of HoG followed by SVM

(a)

(b)

Fig. 7: Some of the images of category 'shove' that performed bad. The left one was identified as 'mushroom' and the right one was identified as 'sea lion'

(a)

(b)

Fig. 8: Some of the images of category 'ant' that performed bad. The left one was identified as 'projectile' and the right one was identified as 'sand bar'

5 Conclusion

The experiment suggests that deep learning methods perform far better than other methods when a large amount of training data is available. The performance of the CNNs is however sensitive to each layer and the associated weights.

References

1. Krizhevsky et. al "ImageNet Classification with Deep Convolutional Neural Networks" [Advances in Neural Information Processing Systems]
2. Resources on the Visual Recognition course homepage
   http://vision.cs.utexas.edu/381V-spring2016/