Transfer Learning with VGG16

Yunsheng Li	Yi Luo	Xiyun Liu	Siwen Yan
A53086891	A53091187	A53099348	A53093655
yul554@ucsd.edu	yil485@ucsd.edu	xil429@ucsd.edu	siy043@ucsd.edu

Abstract

In this assignment, we use the pre-trained network on two different dataset. We tried to train the network with different number of input images per category and visualize the output of the intermediate layer. From the result, it shows that the VGG-16 network, which is trained on millions of images, can be generalized to other dataset simply by adding a linear classifier on the top of 'fc7' layer.

1 Introduction

VGG16 is the keras model of the 16-layer network used by the VGG team in the ILSVRC-2014 competition. In this assignment, we use this pre-trained model to investigate the generalization ability of the model to other dataset, in this assignment, Caltech 256 and Urban Tribes, by only replacing the last layer with a softmax classifier on top. In this assignment, we train this ConvNet with different number of sample per class and do visualization of the filters from layers first and last Convolution Layers of the trained model. Also investigated the effect of intermediate Convolutional Layers by feature extraction.

2 Caltech 256 Classification

2.1 Method

In this section, we first read in the data and pre-process the images by subtracting the mean and doing global contrast normalization. And then, we train the network of VGG-16 with a new FC8 layer by using different number of input images.

2.2 Results

2.2.1 Accuracy and Loss

In this section, we first report the performance(loss and accuracy) of the network by training with different number of input images.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

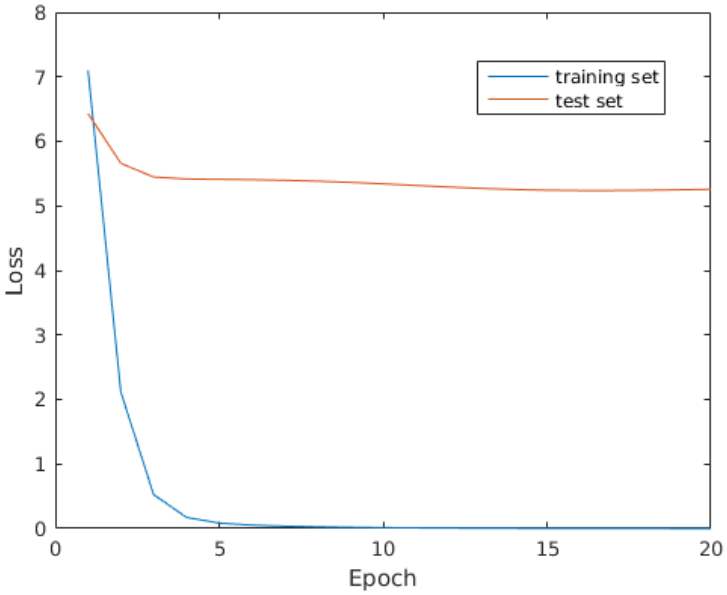


Figure 1: Training with 2 images per category

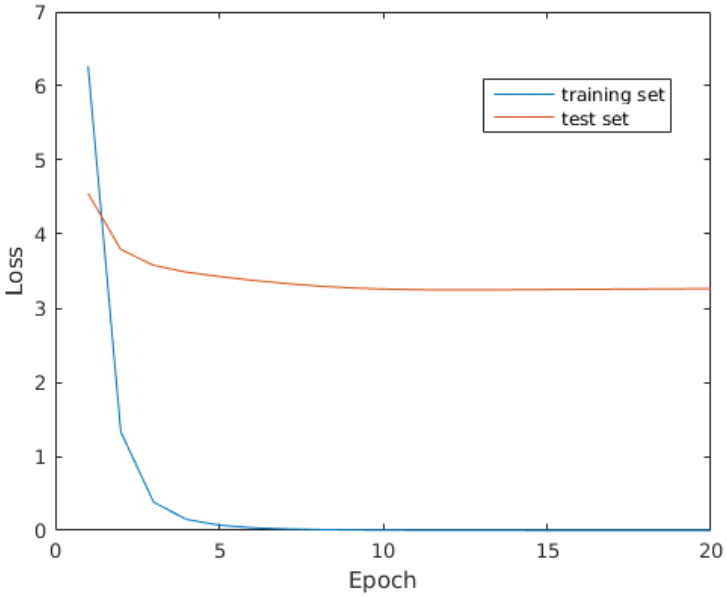


Figure 2: Training with 4 images per category

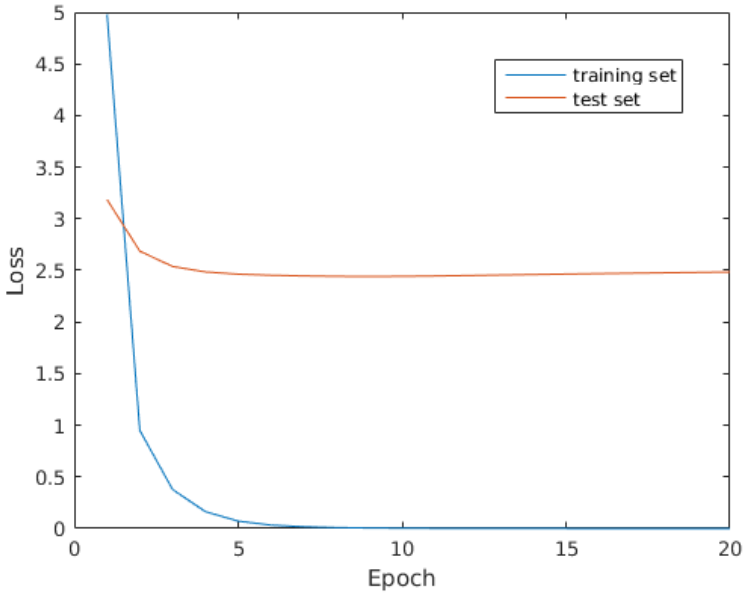


Figure 3: Training with 8 images per category

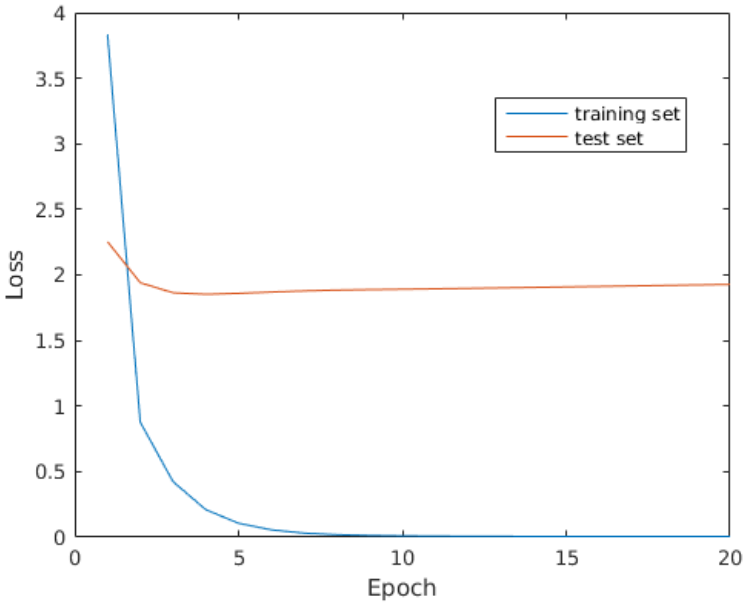


Figure 4: Training with 16 images per category

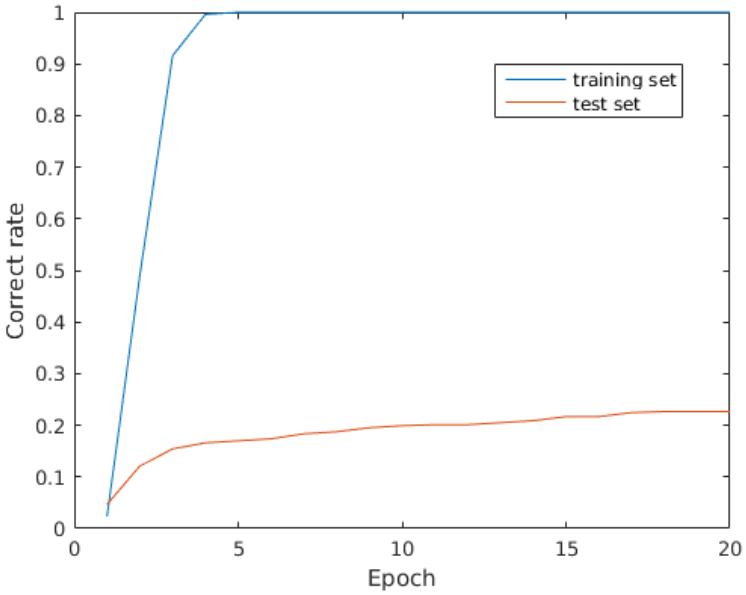


Figure 5: Training with 2 images per category

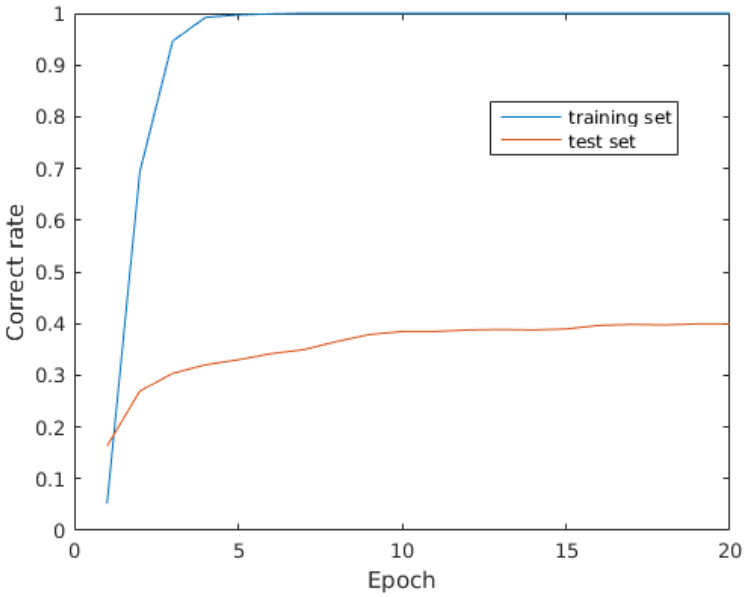


Figure 6: Training with 4 images per category

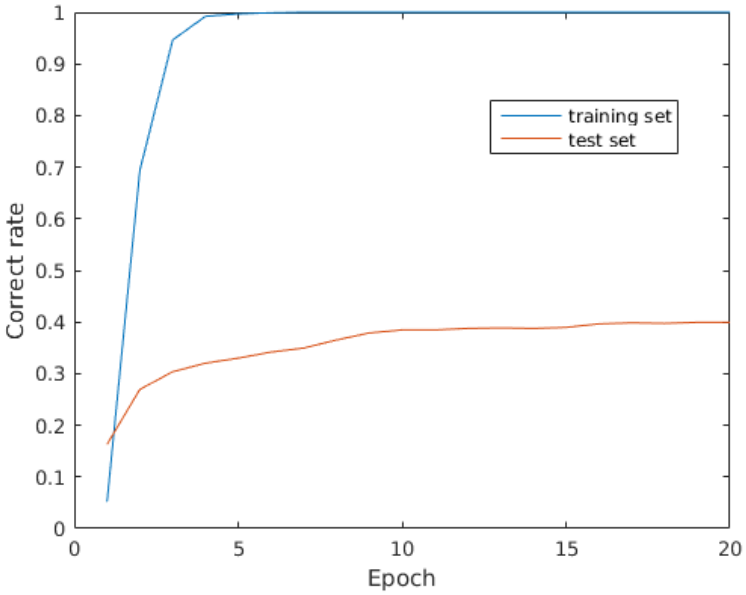


Figure 7: Training with 8 images per category

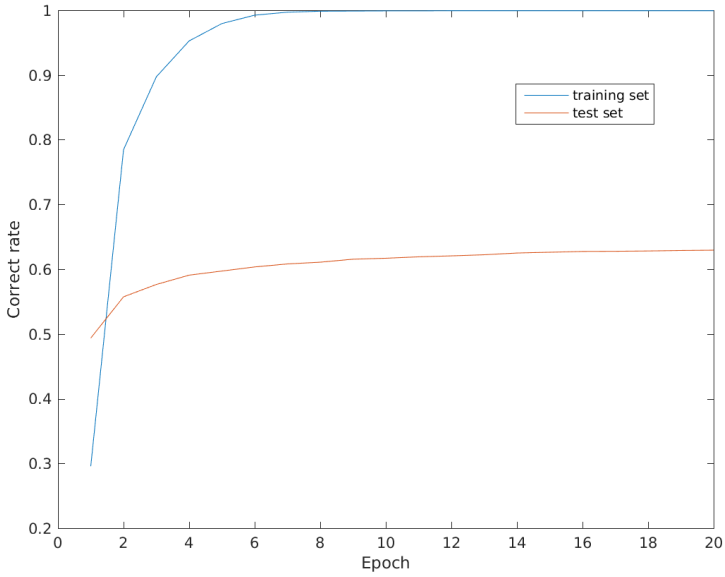


Figure 8: Training with 16 images per category

2.2.2 Classification Accuracy vs. Number of Samples

In this section, we report the classification accuracy of the test set vs the number of sample we use per category, we can find that as the number of sample increases the accuracy also increases.

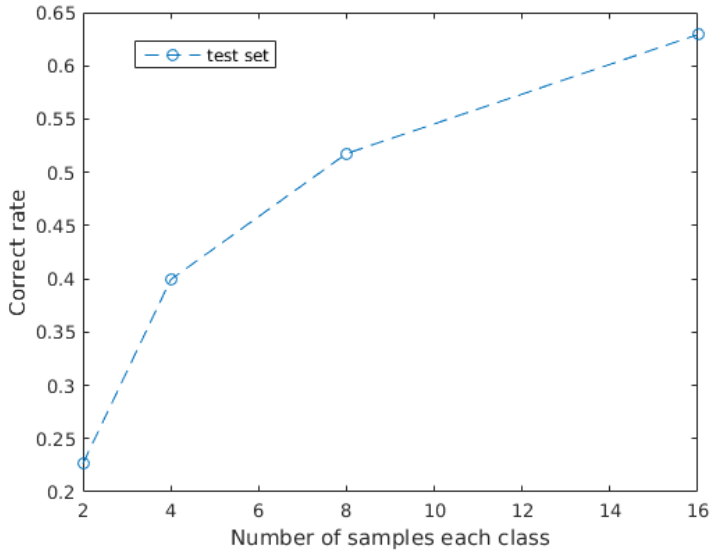


Figure 9: Classification Accuracy vs. Number of Samples

2.2.3 Discussion

We can find that the VGG-16 trained on ImageNet, which has millions of images can be generalized to the Caltech 256 dataset well by fine tuning the last fully connected layer. Even with 2 sample per category the network can converge within 5 epoch and the accuracy can be about 20%, which is much higher than 0.39%(randomly guess). Furthermore, from 9 we know that as we use more and more sample per category, the correct rate also increases. It is because, when we use more samples, the diversity of the each category also increases and it can give a better generalization to each class, thus leading to a better result. We can predict that as the number of sample further increases, the performance of the network can also be better until the redundancy happens.

2.2.4 Visualization

Figure 10: input image

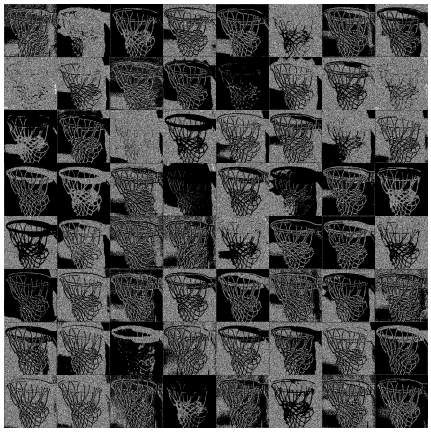


Figure 11: output of conv1-1 layer

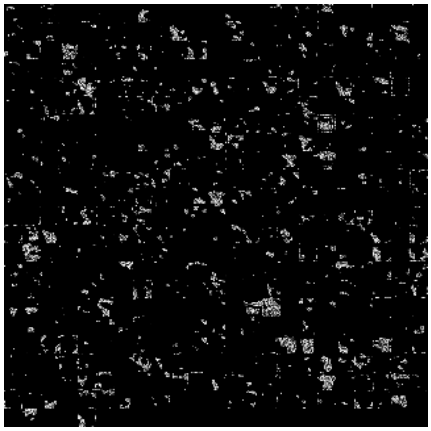


Figure 12: output of conv5-3 layer

In the section, we visualize the input image 'basket'. There are 64 outputs of conv1-1 layer and 512 outputs of conv5-3 layer. In 11, the outline of the basket is very obvious, it is because, the conv1 layer main focuses on the basic information of the input image, such as th edge and corner information or the illumination information. From the output of conv5 layer, we cannot distinguish the basket any more, we infer that the conv5 layer extract more

2.3 Feature Extraction

In this section, we use the features extracted from the conv5 layer and add a linear classifier(a fully connected layer) directly on it. The accuracy and loss are shown as following:

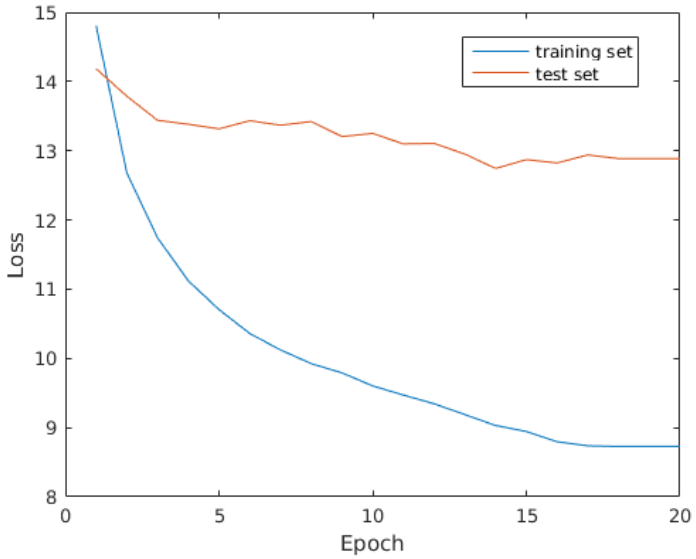


Figure 13: Training with output of intermediate layer

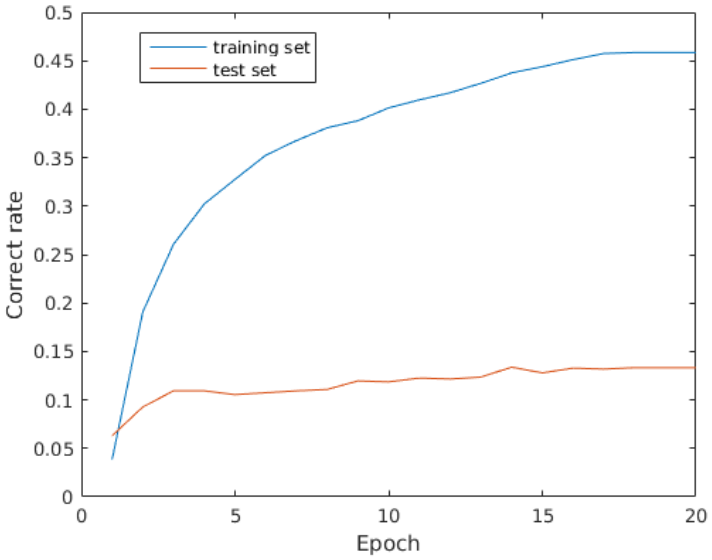


Figure 14: Training with output of intermediate layer

From the result, we can find that, by using the output of conv5 layer, the network can also converge. However, the accuracy is very low compared to the results in the first section. Actually, we have tried to use the output of conv3 and conv4 layer and add a fully connected layer on it, the result is even worse. So it seems that the two fully connected layers that we remove is very important here. I think it is because the two fully connected layers can convert the local feature map(the output of convolutional layer) to the global feature vector, which might include some semantic information.

3 Urban Tribes Classification

3.1 Method

Urban Tribes dataset describes subcultures of people who share common interests and tend to have similar styles of dress, to behave similarly, and to congregate together. It has 11 classes and approximate 100 samples per class.

Here, we do similar preprocessing steps as we have done for Caltech 256. To centering the input data, we use `keras.preprocessing.image.ImageDataGenerator` to subtract the mean. Noted that the global contrast normalization is not performed in this section for model performance. Since the number of samples per class is relatively small, it is very easy to get over-fitting during training. To help prevent overfitting and helps the model generalize better, we "augment" them via a number of random transformations by adjusting some preprocessing parameters, such as the `rotation_range`, `width_shift_range`, `height_shift_range` and `shear_range` in `ImageDataGenerator`. We set all the values to 0.05 when doing experiment with different size of subset from training set.

First, split the training set into 60% for training, 20% for validation and 20% for test. Train the whole training set and plot the loss and accuracy over iterations.

Then, select only a small number of subset(e.g. 2,4,8,16...) from training set to train the same model. Plot the classification accuracy vs. number of samples used per class. Get insight about what does the ConvNet do by visulizing filters from layers first and last Convolution Layers of the trained model.

Further explore the feature extraction by using the intermediate Convolutional Layers as input to the Softmax Layer.

3.2 Results

3.2.1 Training

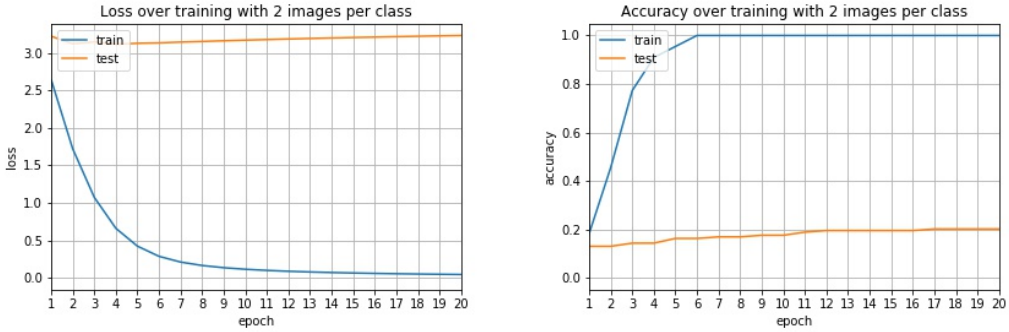
The loss and accuracy overing training on whole training set and test set are shown in figure 15 and figure 16.

Figure 15: Loss over training on whole training dataset

Figure 16: Percent correctness over training on whole training dataset

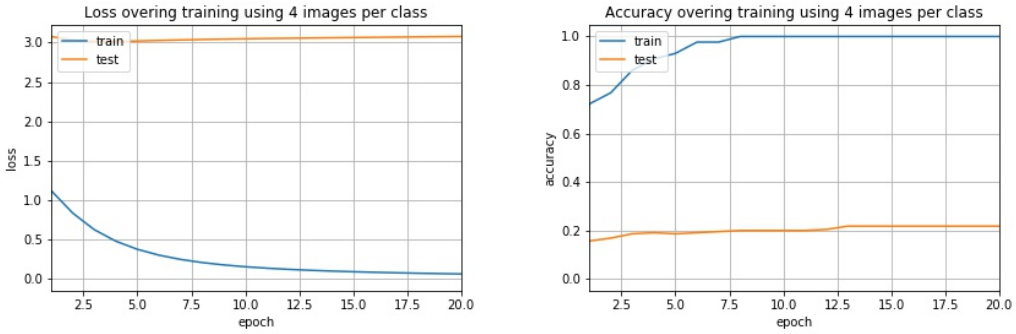
3.2.2 Inference

The loss and accuracy overing training on different size of training set and test set are shown in figure 17 to 20.



(a) The loss of 2 samples per class (b) The accuracy of 2 samples per class

Figure 17: Loss and accuracy using 2 samples per class



(a) The loss of 4 samples per class (b) The accuracy of 4 samples per class

Figure 18: Loss and accuracy using 4 samples per class

(a) The loss of 8 samples per class

(b) The accuracy of 8 samples per class

Figure 19: Loss and accuracy using 8 samples per class

(a) The loss of 16 samples per class

(b) The accuracy of 16 samples per class

Figure 20: Loss and accuracy using 16 samples per class

The accuracy of the models using different number of sample per class is shown in figure 21.

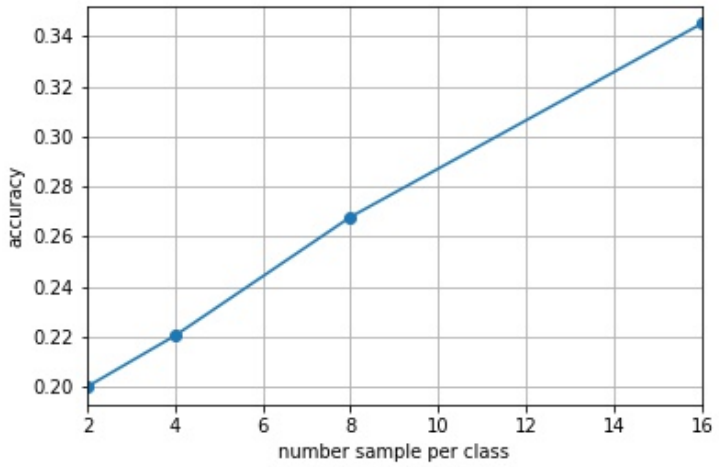


Figure 21: Loss over training on whole training dataset

3.2.3 Visualization

Figure 22: country group

Figure 23: output of conv1-1 layer

Figure 24: output of conv5-3 layer

3.2.4 Feature Extraction

Using 16 samples per class to, we connect the output softmax layer with `block1_conv2`, `block2_conv2`, `block3_conv3`, `block4_conv3`,`block5_conv3`, and then plot the final test accuracy for each case.

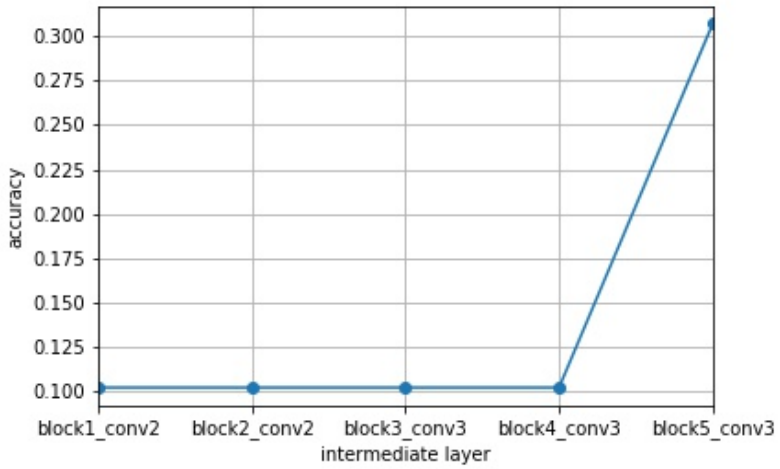


Figure 25: Loss over training on whole training dataset

3.3 Discussion

3.3.1 Inference

The loss and accuracy overing training with difference number of samples per class are shown from figure 17 to 20. For each case, we could see that the loss keeps decreasing and the accuracy keeps increasing over iterations. The network can always converge. Since the subset of training set is so small, it is easy to get over-fitting, and that's why the accuracy of the training set is much higher than the accuracy of the test set and the loss of the training set is much lower than the loss of the test set, although we have tried to avoid over-fitting by performing some random transformations.

As the number of samples per class used in training increasing, the final accuracy is increasing and the final loss is decreasing by looking at figure 21. Actually, it will keep increasing until the subset of training set includes some redundant data.

3.3.2 Visualization

One of the picture in the country group is chosen as the input image for visualization. In figure 23, we can find outlines of people and their clothes. So we can conclude that the conv1 layers can get the basic information of the input image. However, from the output of conv5, there is no obvious outlines for any objects. So the conv5 may extract higher level combination of features.

3.3.3 Feature Extraction

Compared figure 22 with figure 16, we can see that cutting off conv layers, the accuracy is decreasing. For the case that connecting a softmax output layer after block5_conv3, we could see that the accuracy is close to the experiment we did in the last section. However, if we cut off some convolution layers, the result is very bad. Also consider the Caltech 256 feature extraction we have done, it seems that the any fully connected layers is very important to the model. If we remove some of them, the result would be very bad.

4 Temperature-based Softmax Regression

In this section, we train the network with a new softmax layer adding on the top of VGG-16, and try different temperature values. The results are as following:

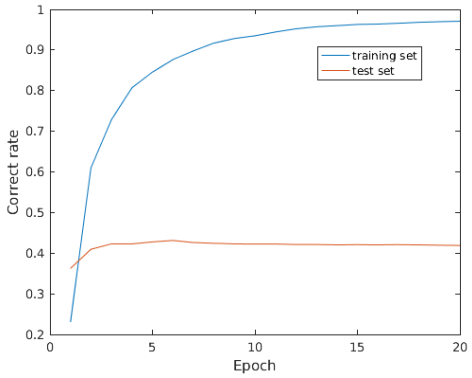


Figure 26: T=1

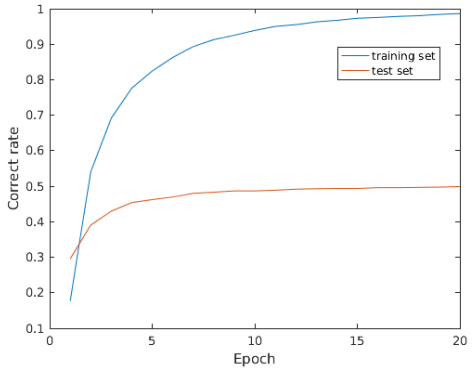


Figure 27: T=2

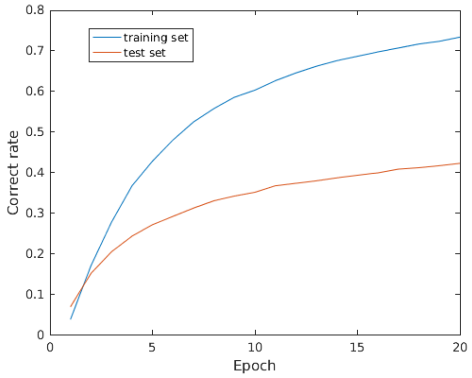


Figure 28: T=4

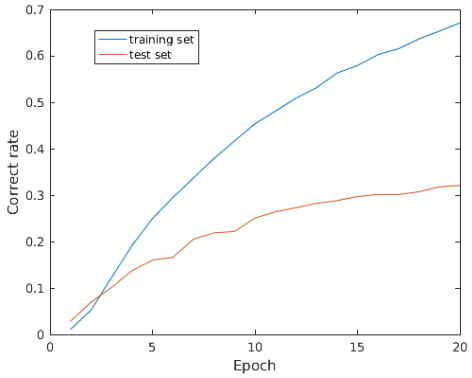


Figure 29: T=8

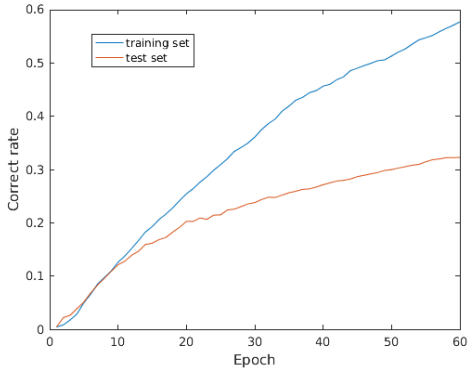


Figure 30: T=16

We can find that when $T = 2$, we can get about 50% accuracy with 8 input images per category. And when T increases the classification performance starts to drop. It is reasonable, because when T increases the norm of the input value of the activation function will shrink, which causes a less diversity of the gradient thereby a lower classification accuracy. Furthermore, when the value of T increases, the speed of convergence will become very slow, even though we train the network for 20 epoch, it is obvious not fully converged yet. It shows that we the input of the activation layer becomes smaller, it will cost more time to train the network.

5 Conclusion

The VGG-16 trained on ImageNet has millions of images can be generalized to the Caltech 256 dataset well. Even with 2 sample per category the network can converge within 5 epoch and the accuracy can be about 20%, which is much higher than 0.39%(randomly guess). For the urban tribes, it is easy to get over-fitting since the subset of training set is so small. Furthermore, we can predict that as the number of sample further increases, the performance of the network can also be better until the redundancy happens. The conv1 layer main focuses on the basic information of the input image, such as th edge and corner information or the illumination information. From the output of conv5 layer, we cannot distinguish the basket any more, we infer that the conv5 layer extract more. The two fully connected layers that we remove is very important here. It may be because the two fully connected layers can convert the local feature map(the output of convolutional layer) to the global feature vector, which might include some semantic information.

6 Contribution of Each Member

We contribute equally to this assignment, we program and discuss together.