CS381V : Visual Recognition - HW2

Ashish Bora

March 2016

1 Introduction

In this assignement, we develop a system for image classification. 25 classes, and a subset of the correespond-ing images were randomly chosen from ImageNet challenge.

Since Convolutional Neural Networks are known to beat other methods by a very large margin for this task, we use them in this assignment. One drawback of CNNs is that they require a lot of training data. Since we are using a subset of the examples in ImageNet, we do not have that much data. Thus, we will use the standard domain adaptation techniques.

2 Approch

Pretrained model weights for the following models are avaialbe on Model Zoo.

1. bvlc-reference-caffenet: This is an implementation of the original AlexNet with some minor modifications. We will henceforth call this model CaffeNet.
2. bvlc-googlenet: This is implementation of GoogleNet model.

GoogleNet architecture is very deep (22 layers) than CaffeNet (8 layers). Even so, the model has considerably lesser number of parameters (50MB vs 200MB for CaffeNet). We shall compare the results on finetuning of these models.

3 Models

We investigate the following variants of finetuning:

1. **Model 1 : CaffeNet1** with layers upto fc7 frozen. Training of last fully connected layer which goes into a 25-way softmax. We keep all hyperparamters to be the same as those used orginally for training, except for making lr-mult of layers upto fc8 = 0 for freezing them.
2. **Model 2 : GoogleNet** with all except the last layer fixed. Training on last fully connnected layer.
3. **Model 3 : CaffeNet2** CaffeNet with fc7 initialized using pretrained weights, but is finetuned for teh new task. We also use a new fully connected layer on top of it which again goes into 25-way sfotmax.
4. **Model 4 : CaffeNet3** Same as Model 3, but with lower learning rate for fc7.

4 Results

4.1 Model 1

Since the model makes random guesses in the beginning, the acuracy is 2% whereas random guess would get 4%. This improves very very rapidly and within 100 minibatches of size 256, we are at about 93% test accuracy. The model does not improve much from there and until 1000 iterations it keeps boucing around this value. At this point the training was terminated and a model snapshot was taken which was then used further. In Fig. 1, shows the confusion matrix for this model.

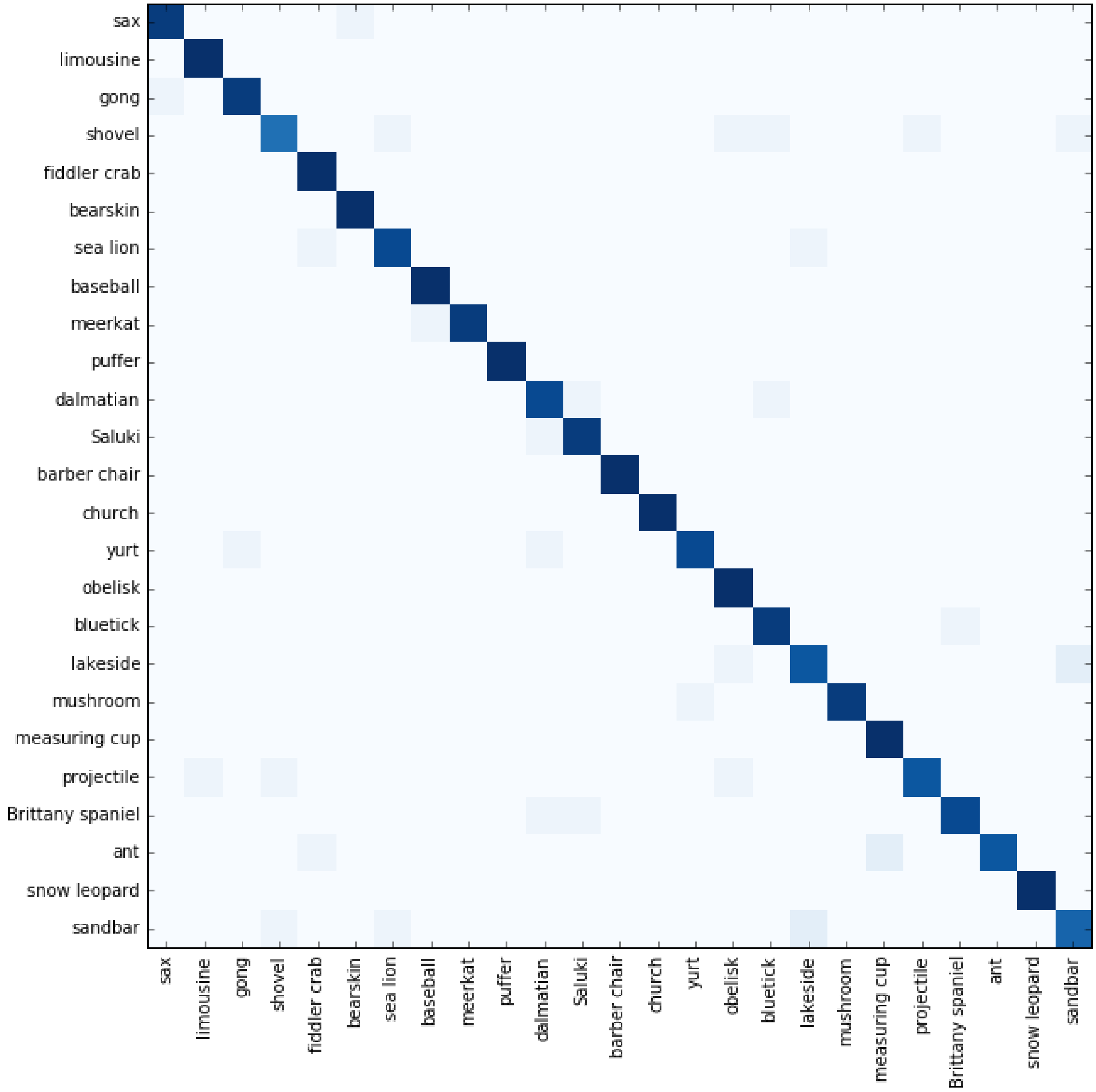

Figure 1: Confusion matrix for Model 1

The final test set accuracy which is the average of the diagonals of the confusion matrix was 93.6%.

4.2 Model 2

This model is better than CaffeNet. While training with lmdb backends, the model gave a very good accuracy of 96.6%. But I could not figure out a way to test it on new images. Unlike the CaffeNet model, the preprocessing is not clearly mentioned, and various combinations that were tried did not work.

Since a confusion matrix could not be obtained, only test set accuracies set accuracy is reported. A trace file of output produced during training is also included for reference (hw2-GoogleNet.err).

4.3 Model 3

The final test set accuracy was 91.8%. The confusion matrix can be seen in Fig 2

4.4 Model 4

The final test set accuracy was 93.0%. The confusion matrix can be seen in Fig 3

5 Observations

We only examine the best performing model, i.e. Model 1. Model 2 gices better test set accuracy but since we do not have the confusion matrix, we use Moel 1 instead which is the second best.

5.1 Mistakes

We try to see which classes are not classified very well. We report their respective accuracy and propose some reasons why these classes might be hard to classify:

1. shovel (0.75) : Lot of variability in viewpoint, background. Many images are of people using a shovel where the actual shovel is digged inside ground or snow. This makes it harder to recognize. Also, many times it is not clear whether the shovel is the central concept in the image since it involces other distinctive objects such as cars, dogs, humans, etc.
2. sandbar (0.8) : Lot of scene variability, lack of saliency, lack of parts. There are not specific parts to a beach. We usually infer it from how shallow the water looks or sand texture, which are not very discriminative.
3. ant (0.85) : Different types of ants, pose variation, usually takes up a very small of the total image. Very thin features, which might get distorted when resizing image before feeding to CNN. Backgorund variation: colorful flowers, leaves, ground. Ants are also confused with fiddler crabs which appear in very similar environments.
4. projectile (0.85) : The projectiles have very nice features such as pointed top, long body and so on. So it is not immediately clear what the problem might be. We see that three test images are misclassified. One has a long car, which is confused with limousine. Another has the projectile in the far which looks like an obelisk. The third image looks more like bullets rather than projectiles. So these mistakes are not very bad.
5. lakeside (0.85) : Same reason as sandbar. Low saliency. It is in fact confused with sandbar, which is reasonable.

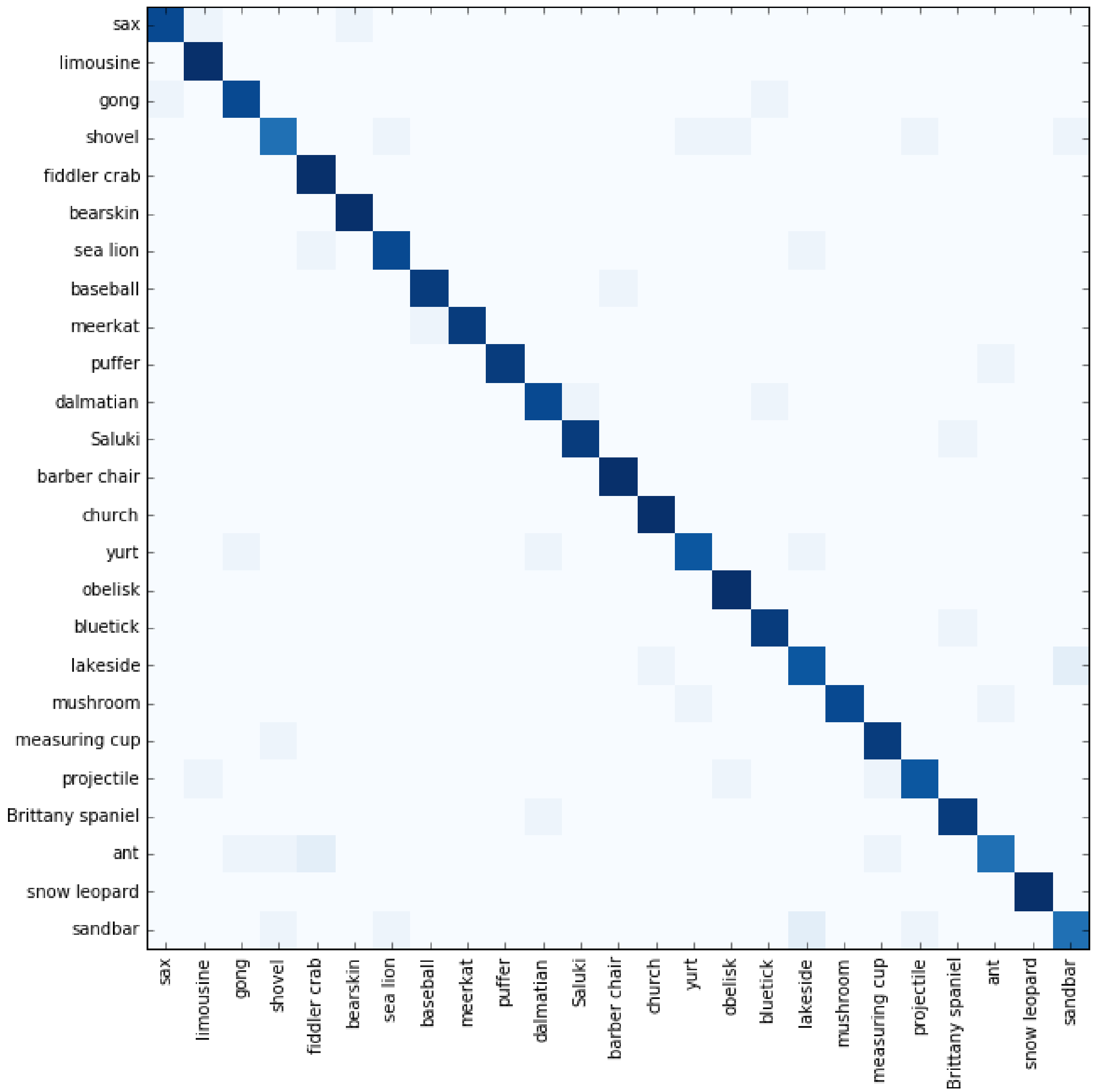

Figure 2: Confusion matrix for Model 3

6 Strengths

Best performing classes (with 100% accuracy) are reported along with some observations.

1. limousine : The dataset actualy contains two different kinds of limousine poses. Outside views are very canonical and distinctive. Inside the limousince, there usally are many people together, which might be a discriminative feature. Thus, we note the network has learned amutimodal classification
2. fiddler crab : It is usually large and has crooked legs, with lot of corners and other visual features. Although there is a lot of scene variability, these features seem to work well to give good accuracy.
3. bearskin : Uniforms give a good template which makes classification easier

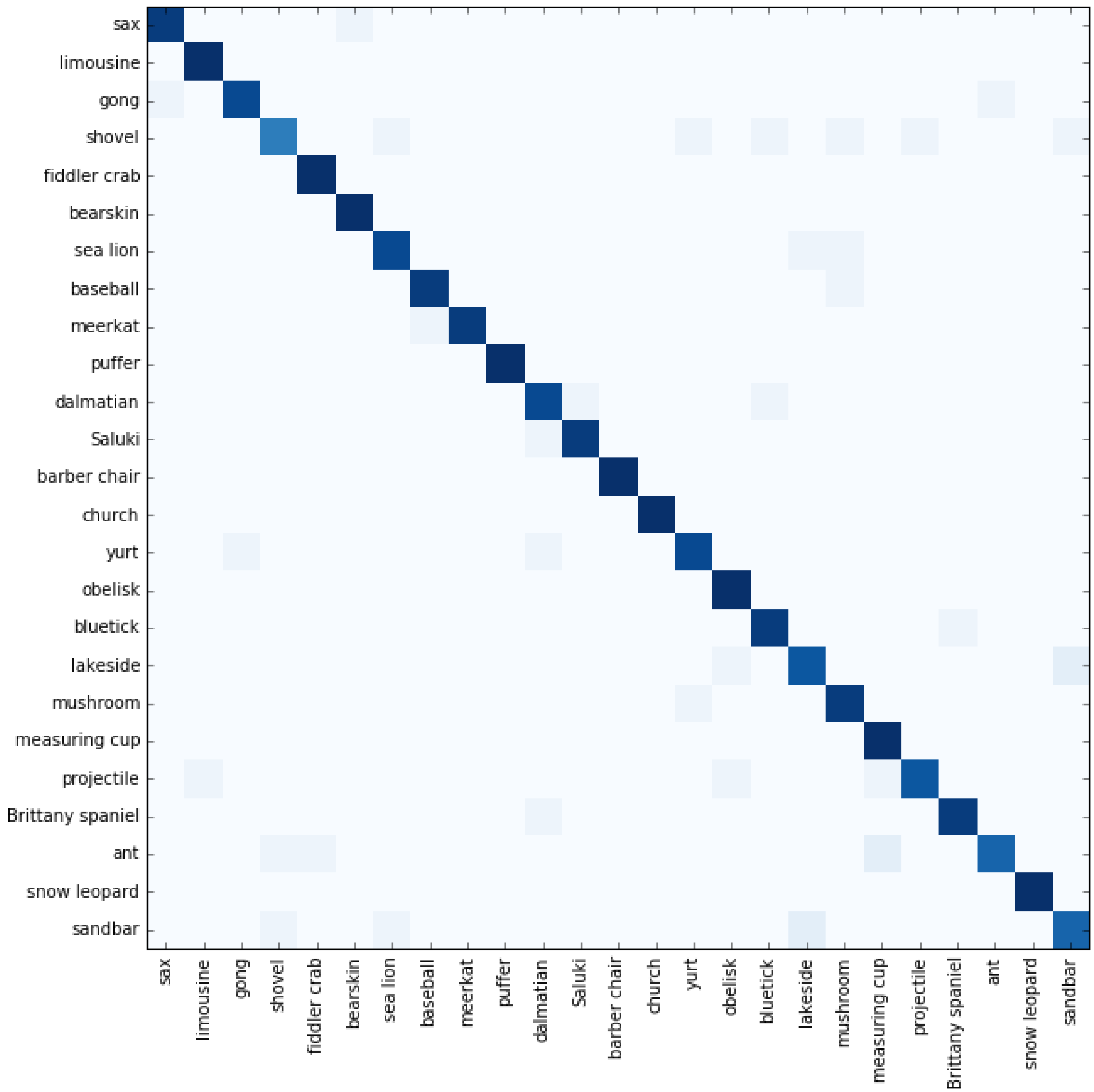

Figure 3: Confusion matrix for Model 4

4. Other full accuracy categories are : baseball, puffer, barber chair, church, obelisk, measuring cup, snow leopard.

References

[1] BVLC caffe package was used for the pupose of this assignement. http://caffe.berkeleyvision.org/