**CS 381V Visual Recognition**

**Coding Assignment 2**

**Ambika Verma**

**(av28944)**

**Challenge A:**

For this challenge the provided training set of 100 images for each of the 25 scene categories from the SUN dataset were used. The following HOG + SVM method was applied –

Algorithm steps:

* Filenames.mat is loaded into Matlab which contains filepaths to the train and test image sets.
* Each training image is read in, converted to grayscale and resized to 200x200 pixels
* HOG features are computed for each training image using a cell size of 32x32
* Class labels are mapped to numeric values ranging from 1 to 25
* One vs One SVM classifiers are trained based on the HOG features and ground truth class labels for the provided training set
* While testing, each test image is converted to grayscale, resized to 200x200 followed by computation of HOG features which are classified by the trained SVM classifier
* Average accuracy and confusion matrix are obtained

The best results obtained are shown below –

**Average accuracy** = 40.80% (similar to that reported in SUN dataset paper).

Confusion matrix –

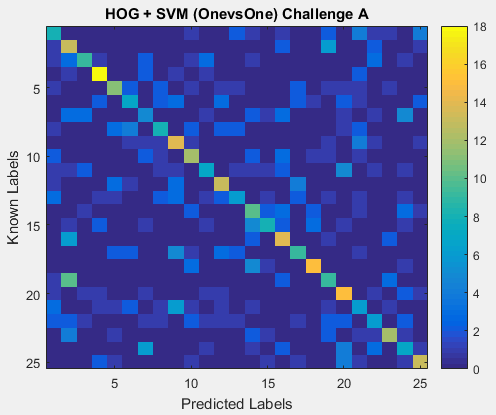


Fig.1 Confusion matrix for HOG + SVM (OnevsOne) (32x32 cell) (Avg accuracy 40.8%)

Some of the best performing as well as worst performing classes are mentioned below –

For the case of best performing HOG+SVM (OnevsOne) (32x32 cells) -

|  |  |  |  |
| --- | --- | --- | --- |
| Best performing classes (decreasing order of accuracy) | Accuracy | Worst performing classes ( decreasing order of accuracy) | Accuracy |
| street (#4) | 72% | dorm room (#6) | 28% |
| sand (#18) | 60% | train railway (#11) | 28% |
| mansion (#20) | 60% | observatory (#24) | 28% |
| orchard (#16) | 56% | chicken coop (#22) | 24% |
| bookstore (#9) | 56% | shoe shop (#21) | 24% |
|  |  | operating room (#13) | 24% |
|  |  | chalet (#7) | 20% |

Table 1 HOG + SVM (OnevsOne) (32x32 cell) (Avg accuracy 40.8%)

The following variation of the HOG +SVM are studied below –

SVM classifier (OnevsAll and OnevsOne):

For multiclass classification (in our case a 25 way classification) the problem can be simplified by using a number of binary classifiers. Two such approaches exist OnevsAll and OnevsOne. Both the approaches have their advantages and disadvantages and the same are observed here.

The overall best performance is obtained for the HOG+SVM (OnevsOne) (32x32 cell) model and details are depicted in Fig 1 and Table 1.

For OnevsOne SVM classifier, N\*(N-1)/2 binary classifiers are trained for an N way classification. Each binary classifier is trained to classify for one of the 2 classes it is trained for. The final class is predicted based on the maximum number of votes obtained from all these classifiers.

For OnevsAll SVM classifer, N binary classifiers are trained for an N way classification. Each binary classifier is trained for one of the N classes to be predicted. The results for OnevsAll are discussed below –

Average Accuracy = 37.6%

Confusion Matrix –

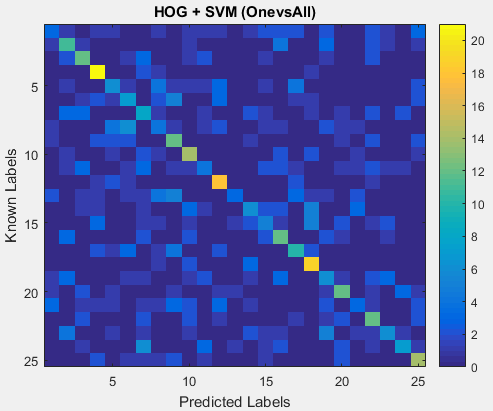


Fig2. Confusion matrix for HOG + SVM (OnevsAll) (32x32 cell) (Avg accuracy 37.6%)

|  |  |  |  |
| --- | --- | --- | --- |
| Best performing classes (decreasing order of accuracy) | Accuracy | Worst performing classes ( decreasing order of accuracy) | Accuracy |
| streets (#4) | 84% | hospital room (#8) | 16% |
| sand (#18) | 76% | train railway (#11) | 16% |
| bow window (#12) | 72% | shoe shop (#21) | 12% |
| bus interior (#25) | 56% | operating room (#13) | 12% |
| crevasse (#10) | 56% | butcher shop (#1) | 12% |

Table 2. HOG + SVM (OnevsAll) (32x32 cell) (Avg accuracy 37.6%)

Though OnevsAll has a clear advantage over OnevsOne in terms of computational requirements, since the former only needs to train N binary classifiers versus the later which requires N\*(N-1)/2 binary classifiers, we see a 3% performance dip.

This performance dip is clearly visible in the confusion matrices (more brighter is better). Some categories with strong orientations in the scene (which are mostly seen as perspective in the images, discussed later) perform well in both cases, such as ‘streets’, ’bow windows’ and ‘sand’.

Additionally some classes perform badly in both cases, such as ‘operating room’, ’shoe shop’ ,’train railway’, ‘hospital room’, highlighting the inability of HOG features to capture scene details faithfully.

Though, it is worthy to note that the OnevsOne method performs better overall since there are classifiers trained for differentiating between every two class instances, resulting in better discrimination especially when two classes seem very similar (such as hospital room (8) and operating room (13)).





Fig3. Examples of ‘street’ class (best performance). The perspective in all these images is clearly captured by HOG features

Fig4. Examples of ‘bow window’ class (one of the best performance). The characteristic in all these images is clearly captured by HOG features





F

Fig5. (Top row) hospital room (#8) class. (Bottom row) operating room (#13) class. These two classes are very similar and easily confused. (More confusion occurs in case of OnevsAll method as compared to OnevsOne method, as can be seen by comparing the two confusion matrices).

Fig6. ‘butcher shop’ class (one of the worst performers). It is evident from the above examples that HOG is unable to capture the high variability in these images. Additionally color does not help in better performance as gradient orientations remain effectively the same in color and grayscale images

Fig7. ‘train railway’ class (one of the worst performers). Again HOG is ineffective in capturing the high variability in orientation of the train in all these scene images.

HOG Cell size variations:

HOG cells over which local gradient orientations are computed and quantized in terms of histograms can be varied in size. Smaller the size of these cells, larger is the HOG feature vector and more spatially detailed local features are obtained. To capture more large-scale spatial information, the cell size is increased.

The effect of cell size increase can be seen easily from below results on the performance. This clearly indicates that for scene recognition fine grain detail (small cell size) is not required and can be depreciating for our performance goals, since scene details and variations occur at a much larger scale.

|  |  |
| --- | --- |
| HOG Cell size | Average Accuracy |
| 2x2 | 32.16% |
| 4x4 | 34.4% |
| 8x8 | 36.48% |
| 16x16 | 37.76% |
| 32x32 | 40.8% |
| 64x64 | 35.84% |

Table 3. HOG cell size comparison for HOG +SVM (OnevsOne). Same effect is observed for OnevsAll method as well

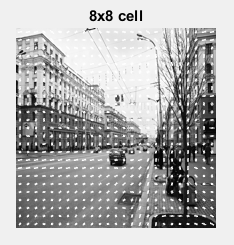
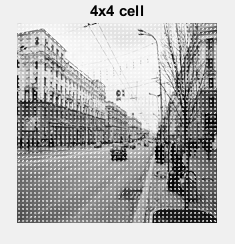




Fig8. Pictorial description of HOG cell sizes

32x32 cell sizes not only capture the large scale scene properties but also allow faster computation due to smaller feature sizes.

**Challenge B:**

As part of this challenge the given training images (100 images for each of the 25 classes) and test images (25 images for each of the 25 classes) were used to finetune the Caffe BVLC reference model. The reference model has 5 convolution layers (with corresponding ReLU, pooling and normalization layers) followed by two fully connected layers (fc6 and fc7). A final fully connected layer (fc8) was added to give 25 outputs (instead of the standard 1000, as designed originally).

Algorithm steps:

* Create train.txt and test.txt with paths to provided training and test images respectively. These files also contain the ground truth class labels for each image (labels ranging from 0 to 24).
* Finetune the BVLC reference model by changing the learning rate to zero of layers 1 to 7. This, ensures that the reference model (pretrained on ImageNet) maintains its learned weights. Only the learning rate of fc8 (the final output layer) is set to a high value (10).
* Training and test image sets are converted to lmdb format
* The pretrained network is finetuned by executing finetune\_sun.sh
* The finetuned network is then applied to all the test images to obtain accuracy, predicted labels and confusion matrix

The results obtained are shown below –

**Average accuracy** = 81.34%

Confusion matrix –

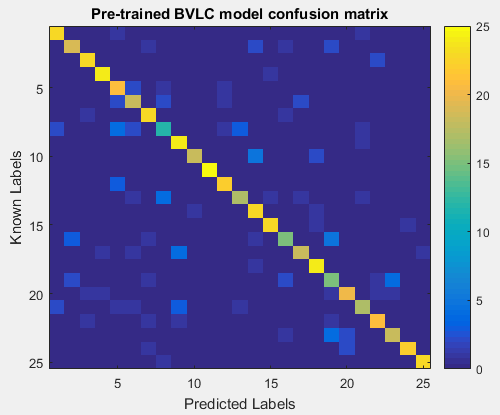


Fig9. Confusion matrix for finetuning pre-trained BVLC model

|  |  |  |  |
| --- | --- | --- | --- |
| Best performing classes (decreasing order of accuracy) | Accuracy | Worst performing classes ( decreasing order of accuracy) | Accuracy |
| Train railway (#11) | 100% | Operating room (#13) | 68% |
| Street (#4) | 96% | Shoe shop (#21) | 68% |
| Bookstore (#9) | 96% | Vegetable garden (#19) | 60% |
| Sand (#18) | 96% | Orchard (#16) | 60% |
| Butcher shop (#1) | 92% | Hospital room (#8) | 48% |
| Covered bridge (#3) | 92% |  |  |
| Chalet (#7) | 92% |  |  |
| Snowfield (#14) | 92% |  |  |
| Landing deck (#15) | 92% |  |  |
| Bus interior (#25) | 92% |  |  |

Table 4. Finetune pre-trained BVLC model (Avg. accuracy 81.34%)

The performance improvement by finetuning a pretrained network is evident from the figure shown above.

The best performing classes such as ‘street’ and ‘sand’ are more accurately identified, but do not reach 100% accuracy.

Though, it is interesting to note that class ‘train railway’ (#11) which was one of the worst performers in all variants of HOG+SVM now achieves 100% accuracy. This, implies that the network is recognizing the distinguishable object (train) and might not be paying attention to the overall scene at all.

Additionally class ‘hospital room’ and ‘operating room’ again do not perform as well as expected. This, points to the apparent confusion between the two classes since they look very similar as shown in Fig. 5.

**Challenge C:**

As part of this challenge the extra training images provided for each class are utilized for training the HOG+SVM model. Same algorithm steps as discussed in challenge A were performed. Same test images as the previous challenges were used to determine the overall accuracy and confusion matrix.

A comparison of two multiclass classification techniques was performed (OnevsAll and OnevsOne). The results obtained are as follows –

**Average Accuracy** = 37% (OnevsOne) and 36.16%(OnevsAll)

Confusion Matrix –

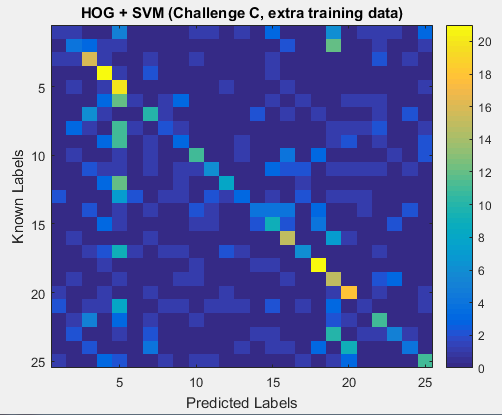


Fig10. Confusion matrix for case of extra training data (HOG+SVM)(OnevsAll) (32x32 cell) The scene classes with very high number of training images (1130 images) such as ‘dining room’ (#5) are highly benefitted, whereas classes with lesser training data suffer (such as ‘dorm room’ (#6) with 115 images)

The decrease in overall performance even in the presence of extra training data can be attributed to the imbalance in the dataset, such as ‘dining room’ class has 1130 training images and ‘dorm room’ class has only 115 training images.

This imbalance in dataset affects OnevsOne classification method more severely (3% drop) as compared to OnevsAll (1% drop). This is consistent with the design of these classification methods, since OnevsOne classifies between each set of 2 classes (which makes it more sensitive to data imbalance). Whereas, OnevsAll can still be more robust as it classifies one class at a time (for each classifier only one class is positive and all other classes are negative).

References:

* "SUN Database: Large-scale Scene Recognition from Abbey to Zoo". J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. IEEE Conference on Computer Vision and Pattern Recognition, 2010.
* https://en.wikipedia.org/wiki/Multiclass\_classification
* http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
* Caffe: Convolutional Architecture for Fast Feature Embedding, Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor, 2014

Appendix:

Class to number mapping –

1 '/b/butchers\_shop'

2 '/f/forest/broadleaf'

3 '/c/covered\_bridge/exterior'

4 '/s/street'

5 '/d/dining\_room'

6 '/d/dorm\_room'

7 '/c/chalet'

8 '/h/hospital\_room'

9 '/b/bookstore'

10 '/c/crevasse'

11 '/t/train\_railway'

12 '/b/bow\_window/indoor'

13 '/o/operating\_room'

14 '/s/snowfield'

15 '/l/landing\_deck'

16 '/o/orchard'

17 '/g/garage/indoor'

18 '/d/desert/sand'

19 '/v/vegetable\_garden'

20 '/m/mansion'

21 '/s/shoe\_shop'

22 '/c/chicken\_coop/outdoor'

23 '/f/formal\_garden'

24 '/o/observatory/outdoor'

25 '/b/bus\_interior'