A Comparative Study about Object Classification Based On Global and Local Features

Hammad Naeem[1], Maria Minhas[2] and Jameel Ahmed[3]

[1]Department of Electrical Engineering ,HITEC University, Taxila Cantt,47040,Pakistan.

[2]Department of Electrical Engineering, University of Minnesota, Duluth, 55812,United states.

[3]Department of Electrical Engineering , HITEC University, Taxila Cantt,47040,Pakistan.

Abstract

Scene classification and object recognition is a hot area of research in the field of computer vision and has always fascinated researchers to explore strategies for optimization of results. Global and local features are manipulated to find a match in the images or scene categories. This paper mainly comprises of finding the scene labels based on the objects present in it. The image is transformed into a feature space and the classifier is trained to differentiate each class in the feature space. Various feature extraction techniques like RGB histogram , SIFT and covariance are explored in this paper to find an optimized result. Different classifiers were tested individually as well as their combinations to achieve better results. Combination of Sparse SIFT and Dense SIFT techniques was found to perform better compared to others.

Keywords: *Scene Classification, object recognition, Bag of words, Sparse SIFT, Dense SIFT.*

1. Introduction

Object recognition in an image is an old yet inconclusive area of research in computer vision. The major problems encountered in object recognition are objects present in the scene having high clutter, illumination variations, high occlusion, high degree of geometric transformations and intra class variation.

Early techniques were mainly appearance based that is using the global features of the image. Origin is the empirical appearance based technique which involves sub-space methods and histograms. Their major drawback was requirement of large number of training images yielding a high computational cost [1]. Also as global features are usually sensitive to variations; these methods were unable to tolerate the geometrical transformations. So, these methods were replaced by techniques which used local invariant features. Since local features are more robust to clutter, occlusion, light changes and geometric transformations hence, strategies using them proved to be more successful [1]. Local features were successfully utilized for a long time but the problem with local features was that they normally lose all the information about the spatial layout of the image and hence lack the descriptive ability. However it should be mentioned that local features are playing an important role in classification and recognition tasks.

In this paper, we have worked on the PASCAL dataset from 2006 in which there are ten different classes of object and we have to detect a particular object in a given image. The ten different object classes are: bicycle, bus, car, cat, cow, dog, horse, motorcycle, people and sheep. Classification is not easy as objects have high degree of geometrical transformation, illumination changes, clutter, occlusion and intra-class variations [1]. There are also cases in which an image contains objects from different classes. This only contributes in making classifier confused in deciding about the particular object in the image.

2. General Strategy

Fairly, large set of training images is provided for each of the ten classes. Classifier is trained with the help of these training images. First features are extracted from images of each class using any feature extraction technique and are then mapped according to the bag of words vocabulary. The feature spaces generated by extraction of features from the images of different classes are grouped into k clusters with the help of k-mean algorithm. The numbers of clusters generated define the size of vocabulary. This technique of grouping or clustering of features is called ‘Bag of words’ technique.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

Bag of words technique has demonstrated a fair deal of success and has surpassed many past techniques. Next histogram is generated for each of the bag of words feature and classifier is trained for each of the class. The classifier is trained to indicate the presence or absence of an object in the image with some level of confidence. Classifier is trained with positive and negative examples of the class to generate decision boundaries for a class in the feature space.

Ground truths are provided for each image. They are compared with the labels assigned by the classifier and ROC curves are generated showing the performance of technique for that particular class.

It is necessary to mention that histograms are normalized in order to have same number of features from each image. For maximum computational efficiency, we normalize all histograms by the total weight of all features in the image, in effect forcing the total number of features in all images to be the same.

2.1 RGB Histogram

First strategy used was computing the histogram of the image. Training images are separated into R, G and B components and histograms are generated. This technique is tested to check the performance of semantic information of an image in object detection. It is expected to give good results with object classes of almost similar background all the time for example car which is expected to have a road (black or grey) or greenery in the background majority of time. Similarly cow might have grass field (green) or sky (blue) background for most of times.

2.2 SIFT (Scale Invariant Feature Transform)

Next strategy is based on the local invariant features. SIFT (Scale Invariant Feature Transform) has brought revolution in the field of computer vision. As SIFT features are fairly robust to geometric variations and perform better in the presence of occlusion and clutter [3], they provide very good features for object or scene labeling. The SIFT feature can be extracted using two techniques, Sparse SIFT and Dense SIFT.

Sparse SIFT: In this, only strong SIFT features are extracted from the image. Instead of taking features from every part of the image uniformly, sparse sift only returns feature which are strong and distinct. There are two or three factors which affect the performance of the sparse SIFT. The most important one is the threshold value used in the SIFT function. The threshold value decides which peaks have to be filtered or rejected in extracting SIFT features. High value of this threshold means that the number of features extracted will be less and some of the features lie below the threshold specified. We tried different values of the threshold and the findings are discussed in the results section.

Dense SIFT: In this technique, SIFT features are uniformly extracted from every part of an image. In this method, window frames with specific scales are defined and then whole image is scanned to extract features for every window frame. This results in a large number of features as now we not only extract strong features but also some weak features. So we used different values of the scale for the frames and received different number of features as discussed in the results section.

2.3 Covariance

It is always a good idea to find the covariance within the image. This is useful in the sense that a particular feature or object has some covariance with another object or a particular background in the image for example cow might have some covariance with the green or blue regions depicting grass or sky in the background. A feature image is constructed where each pixel is represented by the vector of "d" pixel level features [4].Hence any region can be represented by ad-by-d covariance matrix. For comparing two covariance matrices, Tuzelet. al. devised a distance metric based on their generalized eigenvectors[4].The covariance matrix is full-rank, so there will be 9 eigenvectors for a feature vector of length 81.The relation is given in following equation.

$$F(x, y)=[x, y, R(x, y), G(x, y), B(x, y), \left|\frac{\partial I(x, y)}{\partial x}\right|, \left|\frac{\partial I(x, y)}{\partial y}\right|, \left|\frac{\partial I^{2(x, y)}}{\partial x^2}\right|, \left|\frac{\partial I^{2(x, y)}}{\partial y^2}\right|] \quad (1)$$

3. Results

Results for the various strategies used areas follows.

3.1 RGB Histograms

As a start, the trial was done with the RGB histogram feature of the images. The results are shown in table 1.

Table 1: RGB histogram feature of the images

Object	Accuracy
Bicycle	0.401
Bus	0.467
Car	0.613
Cat	0.592
Cow	0.706
Dog	0.524
Horse	0.628
Motorbike	0.563
Person	0.515
Sheep	0.647
Average	0.576

Low Accuracies were expected because object categories with a lot of variation in the background, will not be classified well using RGB features. Best results were obtained for the Cow as expected because for most of the cow scenes, there is usually a greenery and sky in the background, so RGB histograms are able to model these images in a good way. Hence, it can be concluded that RGB histogram can perform good if there is a high c between the object type and colour of the backgrou

3.2 SIFT (Scale Invariant Feature Transformation)

SIFT features were tested with a number of parameters which are discussed in the following sections:

3.2.1 Vocabulary Parameters

As for SIFT, ‘Bag of Features’ technique was used, so three different parameters were tested for the generation of the vocabulary. The parameters include number of images used for generating the vocabulary, size of the vocabulary (number of clusters or words in the vocabulary) and threshold value for extracting the SIFT features from the images. These are explained in the following sections a, b and c.

(a) Number of Images: For this, the number of images used to generate the vocabulary is varied. The other two parameters, vocabulary size and threshold were kept constant at ‘500’ and ‘0’. The numbers of images used are 5, 10 and 15. The results are shown in table 2.

Table 2: Results for varying number of images

	5	10	15
Bicycle	0.611	0.625	0.632
Bus	0.392	0.365	0.365
Car	0.665	0.685	0.673
Cat	0.628	0.649	0.655
Cow	0.638	0.673	0.679
Dog	0.486	0.493	0.471
Horse	0.401	0.394	0.426
Motorbike	0.568	0.515	0.508
Person	0.621	0.628	0.628
Sheep	0.717	0.725	0.711

It can be observed from table 2 that by increasing the number of images per class for vocabulary generation improved the results. However, increasing the number of images does not guarantee an improvement in the results. Hence, no definitive conclusion can be drawn between the performance improvement and number of images used to generate vocabulary. It might be because of the quality of images used for that particular class.

b) Varying Vocabulary Size: In this, the vocabulary size was varied while keeping the threshold and number of images constant at ‘0’ and ‘10’ respectively. The value of ‘k’ was varied in K-mean algorithm, which is used to cluster features to generate different sizes for vocabulary. The values of ‘k’ taken were 200, 500 and 1000. Thus we get 200, 500 and 1000 clusters or words in the vocabulary. The results are shown in table 3.

Table 3: Results for varying the vocabulary size

	200	500	1000
Bicycle	0.649	0.625	0.775
Bus	0.498	0.365	0.667
Car	0.728	0.685	0.745
Cat	0.495	0.649	0.614
Cow	0.701	0.673	0.815
Dog	0.496	0.493	0.549
Horse	0.452	0.394	0.400
Motorbike	0.542	0.515	0.573
Person	0.494	0.628	0.539
Sheep	0.607	0.725	0.814

A definitive trend can be seen from table 3 that by increasing the size of the bag of words, results were greatly improved hence a better result classification can be observed.

c) Varying The Threshold: Next the variation was done for the threshold to extract the SIFT features from an image, while keeping the Vocabulary size and number of images constant as 500 and 10 respectively. Three values (0,5,10) of threshold were used for this computation. Table 4 shows the AUC obtained.

Table 4: Results for varying the threshold value

	0	5	10
Bicycle	0.649	0.720	0.667
Bus	0.498	0.540	0.518
Car	0.728	0.646	0.502
Cat	0.495	0.609	0.568
Cow	0.701	0.644	0.663
Dog	0.496	0.647	0.600
Horse	0.452	0.568	0.598
Motorbike	0.542	0.568	0.597
Person	0.494	0.683	0.518
Sheep	0.607	0.600	0.495

When we increase the value of the threshold, we are actually rejecting peeks with low absolute values. The number of features extracted will be less for high values of threshold but features will be of good quality. Thus choosing high threshold implies that the number of features extracted will be less and more confined and we may not be able to classify the object classes efficiently. From the results, shown in table 4 it is obvious that different classes have good results for threshold value in the range 0 to 5. This is because for threshold value of 0, we are accepting all the features and results are not good for difficult classes like dog, person etc. so in order to have better results for difficult classes features with fairly good quality must be accepted. This is the reason why difficult classes have shown good results with the threshold value of 5, in which we have accepted features of considerably good quality.

3.3 Dense SIFT

Next, the experiment was done with the dense SIFT by varying the size or scale of frames used to extract features from the image. As in Dense SIFT, we scan the whole image and features are extracted uniformly from each part (Frame or window) of the image, so by changing the scale, we effectively change the number of descriptors for an image. In this experiment, the vocabulary size was kept fixed at 500 and number of images used to build vocabulary had a constant value of 10. The scale of the frame had values of 1.5, 5 and 10. The results are shown in table 5.

Table 5: Results for varying scale in dense SIFT

	1.5	5	10
Bicycle	0.785	0.812	0.726
Bus	0.718	0.599	0.645
Car	0.756	0.736	0.708
Cat	0.689	0.667	0.661
Cow	0.841	0.801	0.783
Dog	0.665	0.524	0.501
Horse	0.638	0.498	0.4
Motorbike	0.583	0.595	0.5

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

Person	0.617	0.559	0.583
Sheep	0.695	0.662	0.629

Results came out quite good for a scale value of 1.5. Generally, as we increase the value of the scale, most of the classes don`t have the good results as compared to low scale values. It is observed that Dense SIFT gives better results than Sparse SIFT but for some classes trend is opposite in which Sparse SIFT is better. DENSE SIFT has particularly given better result for the difficult class ‘person’ in which we get high value for AUC as compared to optimized sparse SIFT parameters. But for another difficult class, ‘horse’ results are not good for dense SIFTand sparse SIFT gives better results. So, it is difficult to conclude that dense SIFT is better than sparse SIFT, but still dense SIFT has an edge over sparse sift as results are quite good for some of the classes and average classification value is better for dense SIFT.

3.4 Covariance

Feature space is built with the 9 values from the eigenvectors of the covariance matrix. The results of this technique were not good. It was expected that objects may have covariance with their respective background types. But the results were against this as most of the classes have very low AUC. The results as shown in table 6 are highly unexpected especially for cow which usually has high AUC with other methods. The main reason could be the varying and diverse nature of backgrounds for each object class.

Table 6: Results for covariance

Bicycle	0.467
Bus	0.653
Car	0.559
Cat	0.628
Cow	0.496
Dog	0.552
Horse	0.598
Motorbike	0.605
Person	0.438
Sheep	0.583
Average	0.5579

4. Classifiers

Performance of different classifiers was also tested in order to achieve better results.

4.1 Varying Classifier Type

Here, the main comparison is done between KNN classifier and Support Vector Machine SVM classifier.10 images were used to build a standard vocabulary of 500 words. The value of the threshold is kept constant at 0. Next, different classifiers are selected from the ‘PRtool box’ and their performance is tested with the same dataset.

Table 7: Results for varying classifier types

	KNN-1	SVM-P1	SVM-R1
Bicycle	0.412	0.625	0.627
Bus	0.489	0.365	0.381
Car	0.312	0.685	0.697
Cat	0.624	0.649	0.632
Cow	0.739	0.673	0.675
Dog	0.592	0.493	0.484
Horse	0.457	0.394	0.491
Motorbike	0.513	0.515	0.521
Person	0.516	0.628	0.623
Sheep	0.565	0.725	0.729

The results of the experiment can be seen from table 7.It was observed that SVM classifier clearly performs better as compared to KNN classifier.

4.2 Combining Different Classifiers

In this we combined different classifiers together. The classifiers combined were SVM, KNN and LDC.

Table 8: Results for combining different classifiers

Bicycle	0.322
Bus	0.522
Car	0.306
Cat	0.664
Cow	0.710
Dog	0.601
Horse	0.394
Motorbike	0.467
Person	0.532
Sheep	0.618

From table 8,it is observed that combining different classifier`s didn`t produced good results. The performance of the individual classifiers was better compared to combined ones.

5. Combining Features

In sparse SIFT, we only take strong features and in Dense SIFT, we take features from a window scanned over the whole image. This thing brought an idea that the performance of both these features should be checked together. So for this purpose the sparse and dense SIFT features were combined.

Table 9: Results for combined Sparse and Dense SIFT

Bicycle	0.792
Bus	0.568
Car	0.765
Cat	0.753
Cow	0.849
Dog	0.652
Horse	0.514
Motorbike	0.634
Person	0.552
Sheep	0.714

From table 9 it can be seen that the performance is quite good as compared to the individual case. The main reason for this can be that by combining sparse and dense SIFT; features from all over the image are taken along with the strong features. This might have helped the classifier to draw better decision boundaries, considering the image is now represented in a better way.

6. Optimization

Keeping track of all the experiments done in the previous sections and their optimized parameters, a final classifier is built. As both sparse and dense SIFT show competitive results and it was difficult to decide about the selection of one, final classification is done with both sparse and dense SIFT. For sparse SIFT, the value of the threshold is taken to be ‘0’ and for dense SIFT, the value of the scale is taken to be ‘1.5’ with the spacing step of ‘10’. This time the size of the vocabulary is taken as 500. Vocabulary size of 1000 and 1500 can also be used for further improvement of results. The results of the final classifier are shown in table 10.

Table 10: Results for final classification

	Sparse SIFT	Dense SIFT
Bicycle	0.792	0.817
Bus	0.667	0.725
Car	0.745	0.756
Cat	0.598	0.649
Cow	0.815	0.848
Dog	0.647	0.549
Horse	0.614	0.503
Motorbike	0.539	0.642
Person	0.683	0.540
Sheep	0.814	0.725

7. Conclusion

Object classification is a difficult and complex problem. There is no one technique which outclasses other techniques in all cases. Further, the performances of the classes are not uniform for any given technique. Different classes behave differently with the changed set of parameters. So, it is difficult to devise a single strategy which is equally good for all the classes.

In general, it is found that SIFT performs better than other techniques where dense SIFT has a slight edge over sparse SIFT. The number of images used to build vocabulary also affects the results. The greater the number of images better will be the results. But after a certain value, increase in number of images have a very small impact on result improvement but adds a huge computational cost. The size of the vocabulary is also a very important factor and results improve a lot with the increase in the vocabulary size. Bag of words is a smart technique and has proven quite good in object or scene classification techniques. Also it is observed that quality and quantity of the SIFT features also effects the performance of the classifier. The SVM classifier was found to be better than KNN classifier for this case. The quality (in terms of diversity and description for a particular class) of images used for the training and bag of words is also an important factor for classification performance.

References

[1] M. Everingham, A. Zisserman, C. K. I.Williams, and L. Van Gool. The pascalvisual object classes challenge 2006.PASCAL Network of Excellence on PatternAnalysis, statisticalModelling andComputational Learning.

[2] R. Duin, P. Juszczak, P. Paclik, E.Pekalska, D. de Ridder, and D. Tax,*PRTools4*, *AMatlab Toolbox for patternRecognition*, Delft University ofTechnology, February 2004.

[3] A. Vedaldi and B. Fulkerson, “VLFeat:An open and portable library of computervision algorithms,”

[4] O. Tuzel, F. Porikli, and P. Meer,“Region covariance: A fast descriptor fordetection and classification,” in *In Proc.9th European Conf. on Computer Vision*,2006, pp. 589–600.

Hammad Naeem received his degree of Master in Computer Vision and Robotics from Heriot Watt University-Edinburgh, University of Girona-Spain and University of Burgundy-France. He was on an Erasmus Mundus program which was funded by European Union. He holds Bachelors in Electrical Engineering from Air University, Islamabad, Pakistan. Currently he is serving as a Lecturer in Department of Electrical Engineering at HITEC University, Taxila, Pakistan. He has particular interests in Vision based robotics and image processing based intelligent machines.

Maria Minhas is currently doing her Master’s in Electrical Engineering from University of Minnesota, Duluth, USA. She did her Bachelors in Electrical Engineering from HITEC University, Taxila , Pakistan. She has research interest in image processing and Bio-medical vision based solutions.

Jameel Ahmed received BSc. degree in Electronic Engineering from NED University of Engineering & Technology, Karachi, Pakistan in 1991. He did MSc. Electrical Engineering (Telecommunication) from NUST, Pakistan and PhD (Telecommunication Engineering) from Hamdard University Pakistan and Nanyang Technological University (NTU) Singapore. Subsequently, carried out Post Doctoral Fellowship from NTU Singapore. Jameel Ahmed is actively involved in teaching and research for the last twenty years. Presently, he is serving as professor and chairman, department of Electrical Engineering at HITEC University, Taxila Cantt-Islamabad, Pakistan. Dr Jameel is also heading the Signal Processing & Communication Systems (SPCS) research group of the department.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.