Scene Segmentation and Interpretation

University of Girona

Master in Computer Vision and Robotics [VIBOT]

Final Project - PASCAL Object Recognition Challenge

Name:
Hassan ZAAL
AbdelRahman ABUBAKR

Supervisor:
Prof. Xavier Lladó
Prof. Arnau Oliver

Contents

1 Introduction

Object recognition is the topic of computer vision for finding and identifying objects in an image or video sequence. In this project, we are asked to develop a system that can recognize objects from a number of visual object classes in realistic scenes. This project follow the same philosophy of the well known PASCAL Challenge 2006, in which ten object classes have to be recognized: bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep, person. For each of the ten object classes, the goal is to predict the presence/absence of at least one object of that class in a test image.

To solve this problem, we used the approach of Bag of features, also known as Bag of visual words, and we decided to try a combination of different features and classifiers. Therefore, we used SIFT as descriptors, then combined it with HOG in one descriptor, and reported the difference in classification scores. In addition, we tried 5 different classifiers, Nearest Neighbor, SVM, Neural Networks, AdaBoost, and Randomized Decision Forests.

Regarding the dataset, the PASCAL Visual Object Classes (VOC2006) dataset was used. the 255 training images were used as training set, and the 268 validation set were used in testing. Although we could combine the training and validation images in training, and the provided testing set to test the algorithm, we preferred to use the small sets and try different approaches and classifiers rather than using more images which will take more computation time on our computers.

This report is organized as follows, first we will discuss our approach, and the features used with a comparison between them with different parameters, then the classifiers we tried, and finally reporting the results for all classifiers with the feature descriptors.

2 Bag of Words approach

Originally, the bag-of-words model is a methods of document classification where the frequency of occurrence of each word is used as a feature for training a classifier. By knowing the frequency of certain words, we can have an insight about the content of the text and then classify it by its topic. The same idea can be applied in computer vision for image classification, by treating image features as words. Therefore, in computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.

Our approach depends on dense features extraction, so we have a dense bag of words approach. We also tried different number of words per image, by changing the step size between the window of features. In addition, we tried different features as will be discussed in next section. In this section we will briefly explain what we did to form the dictionary to be used in next steps of classification, and the implementation of this part can be found in the matlab function "BagOfWords.m".

2.1 Visual Vocabulary

First step to form the bag of words is to know your visual vocabulary that will be used to form the dictionary. We used the dense approach, so we used dense SIFT (DSIFT), we also used HOG with DSIFT, and to make sure the features are from the same visual word, we extracted HOG feature in the same center of the window of DSIFT, then added them together as one descriptor. The details of these feature descriptors will be discussed in next section.

In addition to using different features, we changed the parameter of the step size, to allow an overlapping between the windows. In all cases, we use a window size of 16x16 pixels, and at the beginning, we used step size = 16 pixels, so that no overlapping between the windows. However, later we made an overlap between the windows, so, chosed the step size = 8 pixels. Figure 1 shows an example from DSIFT, in which the centers of the windows are shown, which are non-overlapped windows of size 16x16.

2.2 Forming Dictionary

After getting the vocabulary, next step is to form the dictionary. We used k-means clustering on all words we have. We used vl_ikmeans() function in VLFeats library, which deals with the features as integers, so it is faster in clustering, and later in training classifiers. We first used 500 word in the dictionary, but later we used, 800 and 1000 words with the overlapped data.

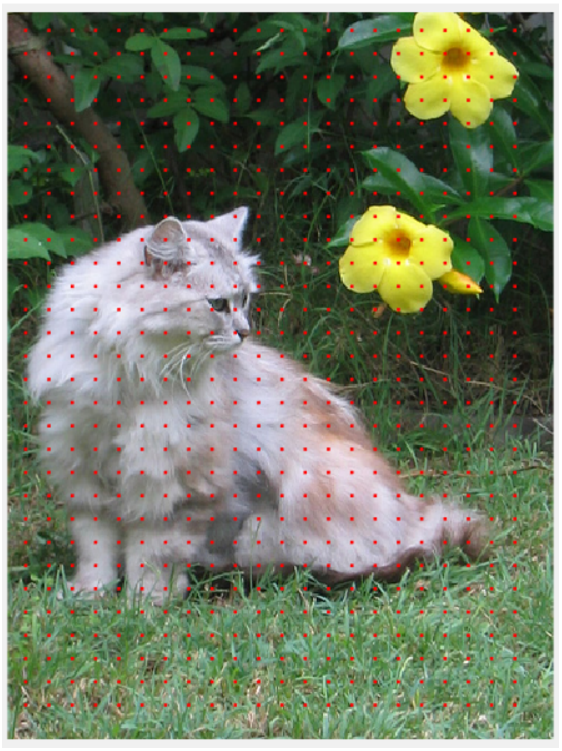

Figure 1: The centers of DSIFT windows along a sample image (no overlap)

2.3 Building Histograms

After having the dictionary, we made a new function "GetHistogramOfWords.m" which takes the input image features, and count the repetition of each visual word relative to dictionary. Therefore, the output from this function is the histogram of the visual words in this image. This histogram is the final feature that will be fed to classifiers, which are the features of both training and testing images.

3 Feature Extraction

Because the computer cannot visually understand the images as human do, we need to define some other features of the images, which should help to discriminate different images depending on some criteria, such as color, texture, etc.

The step of feature extraction is important in our problem, as it forms our vocabulary that builds the dictionary, and hence affects the classifiers. Therefore, we tried different features, in addition to adding two of them together and forming a new feature vector. In this section, we will briefly explain the features used.

3.1 Dense Scale-Invariant Feature Transform (DSIFT)

Dense SIFT is a SIFT descriptor at every location in the image, while the normal SIFT descriptors work at the locations determined by interest points locations. The interest points of SIFT may not be very reliable for some scene/object types. It is biased towards high texture areas, so flat regions tend to be under-represented, so, dense SIFT may be more robust.

We used in our implementation "vl_dsift" function, which is usually used for object categorization. This function works as SIFT on a dense gird of locations at a fixed scale and orientation. We considered in our implementation two cases: overlapping between DSIFT windows, and without overlapping. We specified the descriptor size by the parameter "size" which controls the size of a SIFT spatial bin in pixels. We take bin size equal to 4, so that the total window size = 16x16 pixels.

For the case of non-overlapped windows, we take a window step equal to the same size of the window, which is 16 pixels. In the overlapping window, we take step equal to 8 as the half size of the window.

3.2 Histogram of Oriented Gradients (HOG)

Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest. Implementation of the HOG descriptor algorithm is as follows:

1. Divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell.
2. In our case, we divide the image to cells each cell surrounding the same center of the frame used in dense sift.
3. The cell size we use is 13x13 on sub-image in the same size.
4. The length of the descriptor is 31.

5. Each cell's pixel contributes weighted gradient to its corresponding angular bin.
6. The descriptor values returned from the function "vl_hog" are between zero and one, so we scaled them between 0 and 255 to be suitable for combining them with DSIFT descriptors.

3.3 Comparison of different features

We compared the results using DSIFT with and without overlap, and DSIFT stacked with HOG. In addition, we implemented with different number of clusters: 500, 800 and 1000. We notices that by using overlapped windows with higher number of clusters, the results became better in most of the classes, however, by adding HOG with DSIFT, the results doesn't improve so much, and in some classes the score slightly decreased relative to DSIFT with 1000 histogram bins.

In next sections, all classifiers will be compared relative to these 3 features to compare their performance. For all comparisons in next sections, the parameters are as follows:

DSIFT (no overlap): DSIFT of window size = 16x16 with no overlapping, and number of histogram bins are 500.

DSIFT (overlapped): DSIFT of window size = 16x16 with overlapping, and number of histogram bins are 1000.

DSIFT + HOG: Concatenating DSIFT with HOG features, and use overlapped windows, and number of histogram bins are 800.

4 Classification

classification is the problem of identifying to which of a set of classes, a new observation belongs. To do so, we need a training process, in which the algorithm try to "learn" some pattern in different categories, so that it can recognize new samples not in the training set. In our problem, it is a binary classification problem, which means for each class, the algorithm should determine existence of an object of that class, so that the problem is to classify between "True", or "False". We decided to try 5 different classifiers, 4 of them are provided by PRTools matlab toolbox. In this section we will briefly explain these classifiers, and report the results we got using the features discussed in previous section.

4.1 Nearest Neighbor

First classifier is Nearest Neighbor algorithm. Its idea is simply to check the distance between the test sample, and all training data, then find the training image with minimum distance to the testing image, and assign its label to testing sample. Although its simplicity, it still give us AUC score more than 0.8 in some classes, and even sometimes it is better than SVM.

Table 1 shows the results of this classifier with all features discussed before, which shows that this algorithm gives very good results in car, bus, cow, and bicycle classes. Regarding time, this is the fastest classifier of the 5 classifiers we tried.

Table 1: Nearest Neighbor with different features

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Bicycle	0.6663	0.7416	0.7430
Bus	0.7329	0.7915	0.7707
Car	0.8638	0.8812	0.8823
Cat	0.6403	0.6704	0.6815
Cow	0.7141	0.7316	0.7040
Dog	0.6503	0.6557	0.7098
Horse	0.4855	0.5145	0.5484
Motorbike	0.6232	0.6524	0.6579
Person	0.5881	0.5962	0.5782
Sheep	0.6455	0.6429	0.6478

4.2 Support Vector Machines (SVM)

A Support Vector Machine (SVM) is a classifier formally defined by a separating hyperplane, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. The algorithm then try to find the best hyperplane that increase the separation between different classes in training samples.

In our code, we tried a linear kernel of SVM. We used the svc function in PRTools, which is basically a two-class classifier, so it should be better for our problem. Table 2 shows the results of this classifier with different features, which shows good results in car, cow, and bicycle classes. Regarding time of training, SVM is much more faster than all classifiers that will be discussed in next subsections.

Table 2: SVM with different features

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Bicycle	0.7075	0.6580	0.6253
Bus	0.7987	0.8380	0.8561
Car	0.8954	0.8834	0.8377
Cat	0.6012	0.6380	0.7175
Cow	0.7288	0.8735	0.7024
Dog	0.5541	0.6043	0.5888
Horse	0.5280	0.6821	0.6029
Motorbike	0.5960	0.7000	0.5619
Person	0.4515	0.5469	0.5000
Sheep	0.6701	0.5386	0.6110

4.3 Neural Networks

Our next classifier is Neural Network, we used the PRTools Back-propagation trained feed-forward neural network classifier (bpxnc) function. We kept the default configuration which uses a single hidden layer, its size is the half of the number of objects in training set divided by feature size plus class size with a maximum of 100. We didn't change these parameters because it gave us good results (relative to SVM), in addition to the very long computation time it takes on our computers. However, we used it with all combination of features we have. Table 3 shows the results of using this classifier with different features.

4.4 Adaptive Boosting (AdaBoost)

AdaBoost is a type of "Ensemble Learning" where multiple learners are employed to build a stronger learning algorithm. AdaBoost works by choosing a base algorithm (for our case, decision trees) and iteratively improving it by accounting for the incorrectly classified examples in the training set. We assign equal weights to all the training examples and choose a base algorithm. At each step of iteration,

Table 3: Neural Networks with different features

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Bicycle	0.7324	0.8118	0.8075
Bus	0.8447	0.8966	0.8323
Car	0.8395	0.8780	0.8584
Cat	0.7358	0.7743	0.7841
Cow	0.8627	0.8366	0.8371
Dog	0.7260	0.6875	0.6934
Horse	0.7274	0.6442	0.7105
Motorbike	0.6773	0.7629	0.7680
Person	0.5886	0.5928	0.5609
Sheep	0.7896	0.7308	0.6964

we apply the base algorithm to the training set and increase the weights of some learners. We iterate n times, each time applying base learner on the training set with updated weights. The final model is the weighted sum of the n learners.

Table 4: AdaBoost with Dense sift and histogram of 1000 bins

Classes / Features	DSIFT (overlapped)
Bicycle	0.8383
Bus	0.8188
Car	0.9168
Cat	0.7427
Cow	0.8562
Dog	0.7035
Horse	0.7192
Motorbike	0.7623
Person	0.6009
Sheep	0.7682

We used the PRTools "adaboostc()" function, with 100 weak classifiers. The use of this function, gave us the best results, we even got results with AUC score = 0.91 in "car" class for example. However, the training of this algorithm takes too much time relative to previously discussed algorithms, the whole process of training and testing took around 6 hours on our computers. Table 4 shows the results with respect to DSIFT features with overlapping windows and 1000 histogram bins, which shows that mainly AdaBoost gives a better results than all previous classifiers, specially in classes such as car, Bicycle, Bus, and cow.

4.5 Random Forests

Random decision forests are a learning method for classification, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. We used a "randomforestc()" classifier from PRTools, with 200 trees. This classifier takes much time in training, in our case training and testing process for all classes took around 9 hours.

Table 5 shows the results of random forests with Dense SIFT. Although it gives good results in some classes like car, bus, cat and cow, we can see that mainly Neural-Networks and AdaBoost still better on average.

Table 5: Random Forests with Dense sift and histogram of 1000 bins

Classes / Features	DSIFT (overlapped)
Bicycle	0.7153
Bus	0.8119
Car	0.9146
Cat	0.7293
Cow	0.7915
Dog	0.6763
Horse	0.6657
Motorbike	0.6638
Person	0.5305
Sheep	0.6237

5 Results and discussion

In classification section, we see the performance of each classifier relative to all classes. In this section, we will show the results of each class individually, and highlight the best result of each of them to know which classifier and which features give the best results for each class, in addition to a brief discussion regarding the results of each class.

First class is the Bicycle, table 6 shows the results of this class with all classifiers and features. We can see that AdaBoost gives the best result with area under curve score of 0.8383.

Table 6: The results of Area Under Curve of Class: Bicycle

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.6663	0.7416	0.7430
SVM	0.7075	0.6580	0.6253
Neural-Nets	0.7324	0.8118	0.8075
AdaBoost	-	**0.8383**	-
Random Forest	-	0.7153	-

Next class is the bus, table 7 shows the results of this class with all classifiers and features. We can see that Neural-Networks gives the best result with area under curve score of 0.8966 with DSIFT descriptors with overlapped windows.

Table 7: The results of Area Under Curve of Class: Bus

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.7329	0.7915	0.7707
SVM	0.7987	0.8380	0.8561
Neural-Nets	0.8447	**0.8966**	0.8323
AdaBoost	-	0.8188	-
Random Forest	-	0.8119	-

Table 8: The results of Area Under Curve of Class: Car

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.8638	0.8812	0.8823
SVM	0.8954	0.8834	0.8377
Neural-Nets	0.8395	0.8780	0.8584
AdaBoost	-	**0.9168**	-
Random Forest	-	0.9146	-

Car class gives the best results from all classes, table 8 shows the results of this class with all classifiers and features. We can see that again AdaBoost gives the best result with area under curve score of 0.9168 with DSIFT descriptors with overlapped windows, we can also notice that the car class mainly gives good results with all classifiers. We think this is because cars appears big in the images, and maybe because in most images there are more than one car in the image.

Next class is cat, table 9 shows the results of this class with all classifiers and features. We can see that Neural-network gives the best result with area under curve score of 0.7841 with DSIFT + HOG descriptors with overlapped windows. Although the best result comes from DSIFT + HOG, we can see that only DSIFT still gives relatively good results.

Table 9: The results of Area Under Curve of Class: Cat

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.6403	0.6704	0.6815
SVM	0.6012	0.6380	0.7175
Neural-Nets	0.7358	0.7743	**0.7841**
AdaBoost	-	0.7427	-
Random Forest	-	0.7293	-

For cow class, table 10 shows that SVM gives the best result with area under curve score of 0.8735 with DSIFT descriptor with overlapped windows. We can notice also that Neural-Nets and AdaBoost still give good results.

Table 10: The results of Area Under Curve of Class: Cow

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.7141	0.7316	0.7040
SVM	0.7288	**0.8735**	0.7024
Neural-Nets	0.8627	0.8366	0.8371
AdaBoost	-	0.8562	-
Random Forest	-	0.7915	-

Next class is dog, table 11 shows the results of this class with all classifiers and features. We can see that Neural-network gives the best result with area under curve score of 0.7260 with DSIFT with non-overlapped windows., and 500 histogram bins.

Table 11: The results of Area Under Curve of Class: Dog

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.6503	0.6557	0.7098
SVM	0.5541	0.6043	0.5888
Neural-Nets	**0.7260**	0.6875	0.6934
AdaBoost	-	0.7035	-
Random Forest	-	0.6763	-

For horse class, table 12 shows that Neural-Nets gives the best result with area under curve score of 0.7274 with DSIFT descriptor with non-overlapped windows. We can notice that horse class gives better results with non-overlapped windows, we think this maybe because these animals appears big in most of the images, so that even with non-overlapped windows the histogram can represent these objects well.

Table 12: The results of Area Under Curve of Class: Horse

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.4855	0.5145	0.5484
SVM	0.5280	0.6821	0.6029
Neural-Nets	**0.7274**	0.6442	0.7105
AdaBoost	-	0.7192	-
Random Forest	-	0.6657	-

Table 13 shows the results of motorbike, we can see the best results come from Neural-Networks with DSIFT+HOG features, and overlapping windows. We can also notice that AdaBoost gives very close score to the best score.

Table 13: The results of Area Under Curve of Class: Motorbike

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.6232	0.6524	0.6579
SVM	0.5960	0.7000	0.5619
Neural-Nets	0.6773	0.7629	**0.7680**
AdaBoost	-	0.7623	-
Random Forest	-	0.6638	-

Person class is the most interesting class, the results of this class doesn't increase much with different classifiers and features. We also read that HOG features works good to detect humans, but this doesn't happen. Table 14 shows the results of this class, in which we can see that again AdaBoost gives the best results, but still low score relative to all other classes.

We think that this bad results maybe caused because always people in images appears with other objects, and crowded background, so that the histogram may contain many different visual words from other objects which confuse the classifiers.

Table 14: The results of Area Under Curve of Class: Person

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.5881	0.5962	0.5782
SVM	0.4515	0.5469	0.5000
Neural-Nets	0.5886	0.5928	0.5609
AdaBoost	-	**0.6009**	-
Random Forest	-	0.5305	-

Final class is the sheep, table 15 shows the results of this class, which shows that Neural-Networks gives the best results with non-overlapped windows. We can still see that non-overlapped windows gives better results, the same case we see in horse.

Table 15: The results of Area Under Curve of Class: Sheep

Classes / Features	DSIFT (no overlap)	DSIFT (overlapped)	DSIFT + HOG
Nearest Neighbor	0.6455	0.6429	0.6478
SVM	0.6701	0.5386	0.6110
Neural-Nets	**0.7896**	0.7308	0.6964
AdaBoost	-	0.7682	-
Random Forest	-	0.6237	-

6 Code Organization

In order to organize the code of this project, we implemented many matlab functions, each one has certain task, with passing a parameter to choose the configuration required. First function is for Bag-of-Words generation, which do the whole process of generating vocabulary, and building dictionary using k-means clustering. Second function is for Feature-Extraction, which have a flag to choose the feature you want with the preferred parameters (overlapped windows, or non-overlapped windows). In addition, there is another function for PRTools classifiers in which you can choose between 4 different classifiers (SVM, Neural-Network, AdaBoost, Random Forests) by passing a string parameter containing the name of the classifier to be used.

The results of each run is stored automatically in a text file, in which the name of the class and the auc score are stored. This system helped us to quickly choose the preferred classifier and feature descriptor, in addition to automating the process of results generation.

7 Conclusion and future work

In this project, we worked in the problem of image classification using the PASCAL 2006 challenge framework and dataset. We used Dense SIFT and HOG as feature descriptors, in addition to combining both of them as new descriptor. In this report, we showed the results of using 5 different classifiers with combination of different descriptors, which showed that mainly Neural-Networks gave the best average results in most of the classes, followed by AdaBoost classifier.

As a future work, we can try using different window sizes of DSIFT, and different dictionary lengths, in addition to adding the color and spacial information to the feature descriptors. We can try using more features such as SURF, and combine it with the existing features. As Neural-Networks gave the best results, we can try to use convolution-Neural-Network, which hopefully can give better results.

8 References

1. Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

2. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005.

IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

3. A. Bosch, A. Zisserman, and X. Munoz. Image classifcation using random forests and ferns. In Proc. ICCV, 2007.

4. Raul Rojas. ”AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting”

5. http://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/