Scene Segmentation and Interpretation

PASCAL Project Visual Object Classes Challenge

Submitted by : Wajahat Akhtar & Omair Khalid

Submitted to : Arnau Oliver Malagelada and Xavier Llado

Dated : 28/05/2017

University of Girona - Master VIBOT-11

Contents

List of Figures

1 Introduction

The goal of Pascal Visual Object Classification challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). It is fundamentally a supervised learning learning problem in that a training set of labelled images is provided. The ten object classes that have been selected are: bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person.

The success of classification lies in choosing the features which offer us the highest discriminatory information between the classes, and choosing a classifier that is able to exploit the information given by the features. By visual inspection, it could be seen that for some images classes like sheep and cow, colour information would form a good basis of discrimination because they are usually found in blue and green backgrounds. For more complex classes like motorbike and bicycle, Histogram of Gradients and Dense Sift features could provide better discriminatory basis due to their structural form. Therefore, we chose to use these features for our strategy.

2 Feature Descriptors

The features for an image are extracted for all the image, which includes the object and the background. We experimented with following features in this project.

2.1 Dense Scale Invariant Feature Transform

In Dense Sift, we force extraction of SIFT features at a specific jump, but for all of the image. In contrast, SIFT features are conventionally only extracted at specific keypoint locations, i.e. sparsely. Here, the reasoning behind using Dense SIFT features is to extract a reasonable number of features from an image, and to encode spatial information in the features as well. Spatial information is preserved since feature descriptors extracted out of adjacent locations (row wise) are stored consecutively.

In our implementation, we use step size of 8 between two consecutive windows in order to have a partial overlap. We get a `128 x N` descriptor for an image, where N varies according to the size of an image. Furthermore, we use VLfeat Library to implement Dense SIFT descriptor.

2.2 Histogram of Gradients

The histogram of oriented gradients (HOG) is a feature descriptor which counts occurrences of gradient orientation in localized portions of an image. This method is similar to scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. Moreover, it is different from dense SIFT in that it gives 31 descriptors for one window, as compared to 128 descriptors for one window of SIFT.

In our implementation, we set the `cellSize` to be `8`, to make the window size `8 x 8`. The HOG feature descriptor for an image is obtainted in a 3 dimensional matrix, which is then reshaped into `31 x N`. This is then fed, either to make Bag of Words, or to train/test features. Furthermore, we use VLfeat Library to implement HOG descriptor.

2.3 Colour

Colour features, although very crude, can be useful for us while discriminating between certain classes. For example, most of the images of sheep and cows appear in the context of green grass and blue skies, so color information alone should help classify these image

In our implementation, we extract the mean value of the RGB channels to extract three features from each image.

2.4 Pre-Trained Convolutional Neural Network alexnet

Handcrafted features can be really tedious to compute, and are mostly unable to form the best basis for discrimination. To compute specific features and check results, tweak and combine features, can be quite cumbersome and often leads to unsatisfactory results. One alternative approach is to let the system detect the most useful features itself, and this can be achieved using Convolutional Neural Networks.

In our implementation, we utilized the pre-trained CNN (alexnet) available in MatConvNet library to extract features from the images. After resizing, normalizing and centering the image, it is fed into the CNN. The results from the 20th layer of the CNN are extracted as the features of that image. The size of the features is `4096`.

3 Bag of Words - Dense Sift, HOG, and combined approaches

In our experimentation, we used different features including Histogram Of Gradients, Dense SIFT, and Colour. In order to reduce the dimensionality of the features, we employed the technique of Bag of Words.

In Bag of Words approach, we extract a vocabulary of visual words which form a part of positive as well as negative class images. Visual words are feature descriptors of any sort e.g. HOG, SIFT, Dense SIFT, etc. that represent the image. After extracting features from each of the images in our pool, we concatenate them together to form a large space of features. On this feature space, we perform K-Means Clustering in order to get centers or "Words" for our dictionary. K-means Clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Each prototype is a visual word. We want to imply that an image can only be comprised of the words available in the dictionary, also known as Bag of Words.

3.1 Feature Extraction using Bag of Words

After forming Bag of Words, we can start our feature extraction process, which goes as follows. First, we extract the features (HOG, Dense SIFT etc.) from an image. Next, we take all the feature descriptors and map them onto the 'words' in our Bag-of- Words. At this stage, each feature descriptor takes the form of its nearest visual word. Next, we count the frequency of occurrence of each word in the image, and save it as a histogram. This histogram is our final feature of this image, which is telling us the frequency of occurrence of each word in the image. For each image, the same process will be repeated, before feeding features of all the images into the classifier.

4 Classification

In this project, three types of classifiers have been experimented with.

4.1 Trivial Classifier

The trivial classifier classifies by returning the ratio of L2 distance between nearest positive (class) feature vector and nearest negative (non-class) feature vector. The performance of this classifier was the poorest

4.2 Support Vector Machine, nu-algorithm

Classifying data is a common task in machine learning. In order to perform better classification we need to carefully select the classifier depending on the type of classification needed. After some research we found that Support Vector Machine (SVM)is a great choice in order to perform binary classification. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms.The task assigned was to classify whether the testing images contains a given class objects or not.Which was two-class 1 vs rest binary classfication problem.

To perform binary classification we used a classifier called SVC a Support Vector Machine classifier from Prtools.SVC is basically a two-class classifier. Multi-class problems are solved in a one-against-rest fashion by MCLASSC. The resulting base-classifiers are combined by the maximum confidence rule.The results were promising but the classifier was limited to linear classification but in order to perform non-linear classification we found that there was support vector machine classifier called NUSVC from PRTools which use kernels in order to solve non-linear classification problem.It use kernel of polynomial degree 3 , implicitly mapping their inputs into high-dimensional feature spaces Solving a non-linear classification problem , secondly using Kernel is faster and better appraoch to deal with non-linearity as the features of both classifier could be lying in same cluster making difficult for the classifier to dicriminate between them.We have used diffierent other binary classifier from Vl.feat and libsvm libraries but the one we choosed were among better considering our problem.

5 Experiments

In order to determine the best features and the best classification techniques, we conducted a series of experiments and verified the quality of the approach with the ROC curves obtained. In this section, all the approaches tried are explained. The results obtained are shown in the next section.

5.1 Dense SIFT + Trivial Classifier

In this approach, only the Dense SIFT features are extracted from all the training images. A Bag of Words of 300 and 1000 words, is created and tested, respectively. The classifier used is the Trivial Classifier.

5.2 Histogram Of Gradients + Trivial Classifier

In this approach, only the HOG features are extracted from all the training images. A Bag of Words of 500 words, is created and tested, respectively. The classifier used is the Trivial Classifier. We also form Bag of Words using 300 and 1000 words and observe that it increases performance.

5.3 Histogram Of Gradients + Support Vector Machine Classifier

In this approach, only the HOG features are extracted from all the training images. A Bag of Words of 500 words, is created and tested, respectively. The classifier used is the NUSVM classifier.

5.4 (Dense SIFT+ Histogram of Gradients) + Support Vector Machine Classifier

In this approach, the HOG and Dense SIFT features are extracted from all the training images. A Bag of Words of 500 words is created for the HOG features and a Bag of Words of 1000 words is created for the Dense Sift features. The classifier used is the Support Vector machine classifer of two types(with and without Kernel).

5.5 (Dense SIFT+ Histogram of Gradients + Colour) + Support Vector Machine Classifier

In this approach, the HOG, Dense SIFT, and Colour features are extracted from all the training images. A Bag of Words of 500 words is created for the HOG features and a Bag of Words of 1000 words is created for the Dense Sift features. The classifier used is the Support Vector Machine classifier of two types(with and without Kernel).

5.6 Features from CNN + Support Vector Machine Classifier

in this approach, features are extracted using alexnet pre-trained CNN and Support Vector Machine classifier is used for classification.

6 Results

The following table shows results of Area Under Curve of the ROC curve obtained using by each of the techniques.

Approach	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
5.1(300 BoW)	75.5	69.6	86.7	68.1	83.1	76.1	63.3	69.9	65.7	58.9
5.1(1000 BoW)	79.6	76.8	85.4	62.3	78.5	68.4	61.9	71.0	63.3	63.9
5.2	77.2	82.2	87.6	75.8	79.2	69.6	64.4	69.9	60.5	66.7
5.3	77.9	91.2	87.5	74.8	87.7	75.1	74.8	64.0	63.6	76.4
5.4	80.5	82.8	84.0	72.9	77.1	73.6	59.6	68.6	60.8	56.6
5.5	80.7	81.9	90.2	73.9	85.3	66.0	78.5	66.5	65.6	80.1
5.6	98.7	90.9	88.5	95.5	98.1	88.9	93.9	94.7	79.6	92.4

Table 1: AUC obtained using Different Techniques - Val Set

After determining the best approach to be 6.6, we tested the classifier on the full test dataset. The result of the best approach is as follows

Approach	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
alexnet + SVM	93.3	94.1	89.2	94.3	99.2	85.4	84.2	90.1	72.0	98.8

Table 2: AUC obtained using alexnet+SVM - Full Test Set

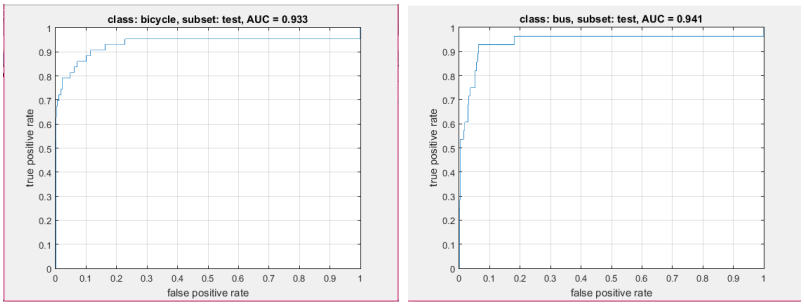

Figure 1: Left: ROC for Bicycle class: ROC for Bus class

.

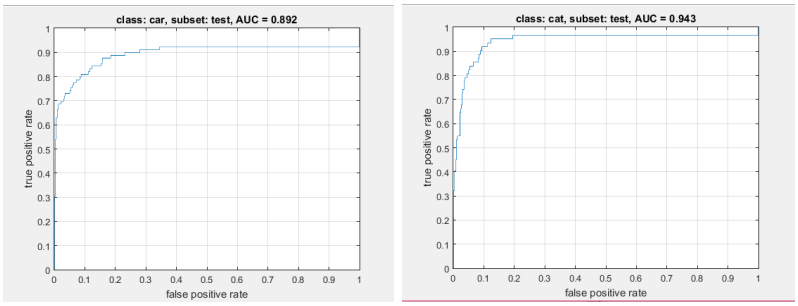

Figure 2: Left: ROC for Car class, Right : ROC for Cat class

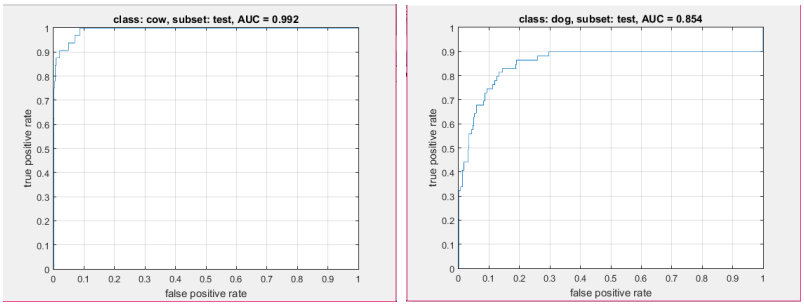

Figure 3: Left: ROC for Cow class , Right : ROC for dog class

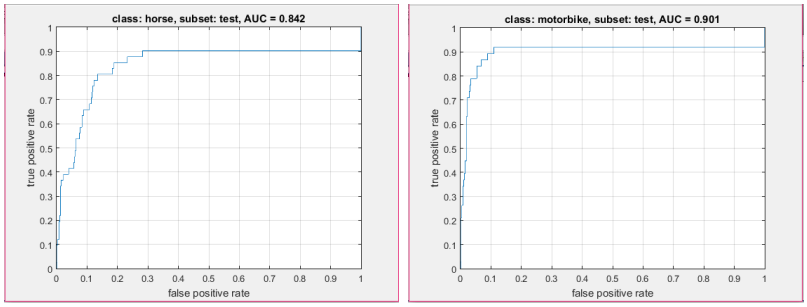

Figure 4: Left: ROC for Horse class , Right : ROC for motorbike class

.

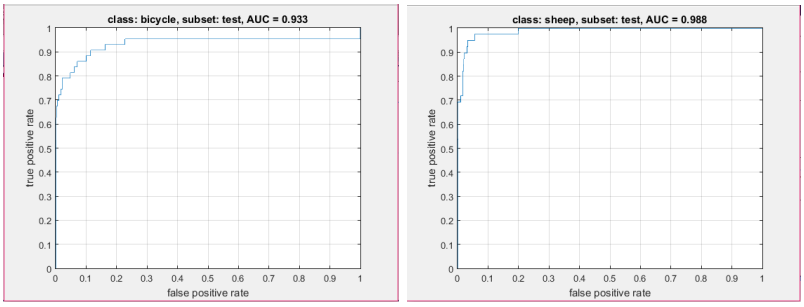

Figure 5: Left: ROC for Person class , Right : ROC for Sheep class

7 Conclusion

The approach that gave us the best result(92.1 AUC) was the one where we extracted features from CNN and used Support Vector Machine Classifier. As the alexnet CNN was already trained for 1000 classes, it had learnt the best features and was able to produce the best features for our case (of 10 classes) as well. Further improvement in the results is possible by using Transfer Learning, whereby only a part (the fully connected layers) of the CNN are trained with our own data.

Although the second best approach(77.3 AUV) was the one where we extracted only HOG features and used a NUSVM classifier. This showed that HoG features form a strong basis for classification. As compared to using the trivial classifier, the results improved 5.6%.

Further suggested improvements are as follows:

- In our approach for Bag of Words, we were forming separate bag of words for HOG and DSIFT features. An improvement upon this could be that we make a single bag of words, by concatenating the HOG and SIFT features, and then applying K-Means Clustering.
- We observed that in the training data, the number of negative examples are far more than the positive examples. Therefore, when making the Bag of Words, we extract most of the words from the negative examples. A better strategy could be to take all the positive examples from the training data, and take an equal number of negative examples to make the Bag of Words.
- While training classifiers, it may be better to feed it with equal number of positive and negative examples, so that the classifiers are able to discriminate properly. If we feed it with too much of the negative examples, there is a high probability that even the positive images will be classified as negative images.
- Instead of using pre-trained CNN for feature extraction, a new CNN may have been designed which may have been better suited for the problem.