Multi-Class Object Recognition Using Shared SIFT Features

Siddharth Batra
(In collaboration with Stephen Gould and Prof. Andrew Ng)

Abstract

The current state of the art object recognition systems work reasonably well for limited data sets. In the case of feature based methods, apart from being object specific the amount of features in the database also grows linearly with the number of objects. Similarly, in patch based methods if the number of patches are kept constant irrespective of the number of objects to be recognized, the performance drops. This paper presents a novel approach towards sharing features between objects of the same class to create a dictionary of shared features for that class. It uses a combination of the feature & patch based approaches to enable creation of Shared SIFT features & also recognition of other objects of the same class using these Shared SIFT features.

1. Introduction

The problem of object detection within images is an age old one and the solution to which has tremendous applications in a variety of fields such as robotics, human computer interaction and online advertising. One of the commonly used feature based techniques for object recognition is SIFT (Scale Invariant Feature Transform) **[1]**. SIFT gives extremely good results for very specific objects but does not generalize well across a class of objects. Also, the number of features in the database increases linearly with the number of objects to be recognized. Hence, SIFT is not a very good choice for recognizing objects from the same class, especially for objects it has not been trained to recognize.

A different approach is the use of shared patch based features for multi-class object detection **[2]**. This approach works quite well but if the numbers of patches are kept constant and the numbers of objects are increased the performance will take a hit. It is faced with a similar challenge of increasing patch features with the number of objects though not linearly. Similar to these Torralba patches, **[3]** also uses a patch based approach via the use of Gabor wavelets to enable feature sharing for object recognition. Another interesting approach is to create a 3D model of the object class from training images and share SIFT features by simply placing them onto correct locations of the 3D model **[4]**. This approach does not take into account the fact that SIFT features work very well for specific objects but do not generalize well across that class of objects.

This paper presents a novel approach that combines elements from the feature and patch based approaches to provide a method that creates Shared SIFT features which can be used for recognizing multiple objects of the same class.

In its first stage termed as descriptor training it looks at a set of images from the same object class and uses patch based correlation to find similar areas on the image which should be shared using the proposed Shared SIFT technique. This provides a set of descriptors which act as the dictionary for that object class, such as mugs, bottles etc. Using matching techniques in feature space, matches are located from the dictionary in positive and negative training images and are used to train a multinomial naïve Bayes classifier. This classifier is then used to iterate over a given test image to assign to each part a probability that an object from that object class is located in that part. Filtering based on these probabilities results in localizing regions within the test image that contain objects of the same class which the classifier has never seen before.

2. Shared SIFT Descriptors

The original SIFT approach uses difference of Gaussians, neighborhood maximization & minimization followed by magnitude & other filtering mechanisms to find unique locations on the image which are consistent and hence should be found repeatedly under a variety of scales ,orientations and viewpoints. Further, the algorithm uses gradient magnitude and orientation maps to assign each of these locations (or keypoints) with an orientation & scale which is used to create a descriptor of the patch around that keypoint.

The proposed Shared SIFT approach aims to use gradient and orientation maps from similar looking patches in a neighborhood of the original keypoint from all of the descriptor training images to create a shared descriptor for that point. Figure 1 illustrates this process.

To begin with keypoints (not descriptors) are individually extracted from all descriptor training images using the original SIFT approach. A patch correlation technique similar to the one described in **[2]** is then used to find the most similar match to the patch surrounding the keypoint in the source image in all of the other descriptor

training images. To avoid having to perfectly align the images and allowing sharing across images of slightly different sizes this search is conducted in a neighborhood of the location of the keypoint in all of the descriptor training images. In the next step, gradient magnitude and orientation maps at these similar patches are combined to give orientation and scale information to that keypoint.

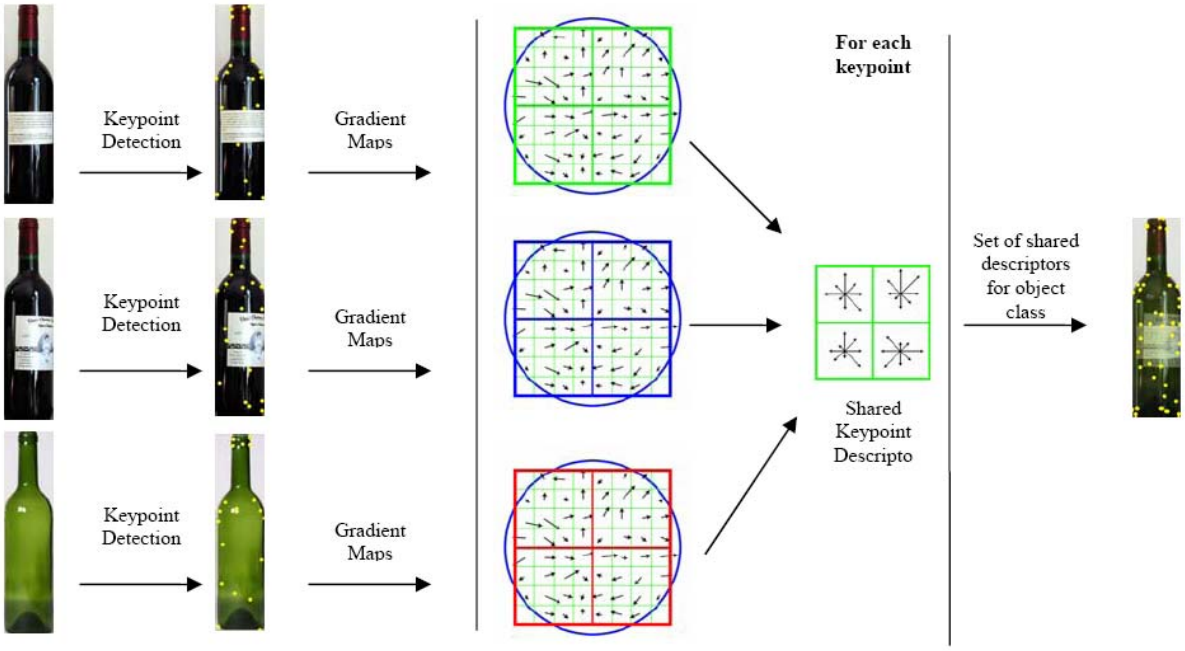


Figure 1: The process of creating Shared SIFT descriptors

Further, instead of creating separate histograms for each of the individual patches in the images, the keypoint information computed above and all the gradient maps from similar patches are used to create a shared or averaged histogram which provides a smoothened common representation of all the patches being shared. The shared sift descriptor for that location is then computed from this shared histogram.

One major advantage of sharing patches across the descriptor images is that variance metrics can be used to judge how well the shared patch represents the original patches in the descriptor training images.

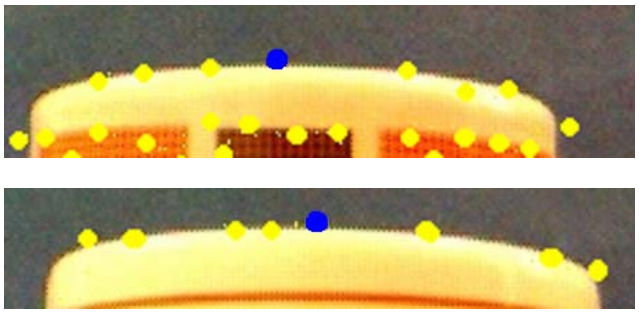


Figure 2: SIFT keypoints found on the rims of two mugs. A patch correlation algorithm found the keypoints in blue to be representing a very similar patch

Shown in Figure 2 is the zoomed in view of keypoints on the upper rims of two mugs. The patch based correlation found the patch around the keypoints in blue to be quite similar in both the images. On the other hand Figure 2 also shows 2 keypoints in blue which were found to be similar but the similarity was quite low.

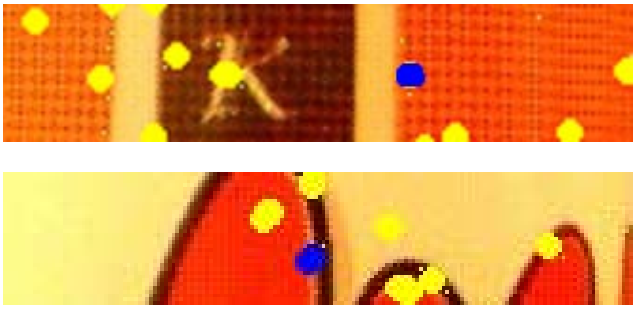
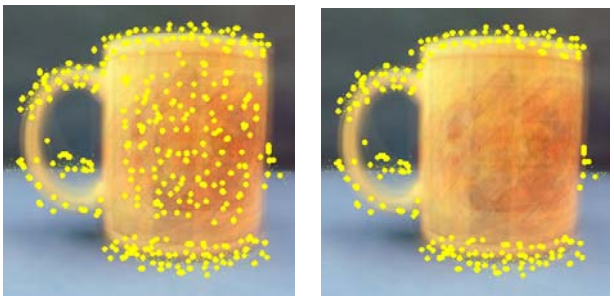


Figure 3: SIFT keypoints found on the centre of two mugs. A patch correlation algorithm found the keypoints in blue to be representing a not so similar patch

Thus, it is quite intuitive to see that a shared descriptor for the keypoints in Figure 2 will be much closer in the 128 dimension space (All SIFT descriptors are 128 length vectors) to the keypoints it represents and similarly the shared descriptor that represents the keypoints in blue in Figure 3 will be further away from the keypoints it represents in feature space. The variance metric shown in Figure 4 computes the average distance between a shared descriptor and the individual patches in the descriptor training images that it represents. The distance is a Euclidean distance computed in the 128 dimensional space.



(a) (b)

Figure 6: A representation of a shared object class after sharing 18 mug images (a) Plots of all the shared descriptors (b) Plots of the shared descriptors after variance based filtering

$$Dist(d_1, d_2) = \sum_{i=1}^{128} \| d_1^i - d_2^i \|$$

Where d_1 *and* d_2 represent any 128 dimension descriptors used in SIFT.

$$\mathrm{var}(d) = \frac{\sum_{i=1}^{m} Dist(d, f_i)}{m}$$

Where d represents the 128 dimension shared descriptor and f represents the set of m descriptors from the m patches it represents.

Figure 4: The formula used to compute the variance of the Shared descriptor with descriptors of the patches it represents

Computing these variance metrics for each shared descriptor and taking only the descriptors which have a variance less than a set threshold results in taking only the descriptors which are good indicators of that object class rather than any of the specific objects. Figure 5(a) shows all the shared descriptors found after sharing a set of 18 mug images. Figure 5(b) shows the set remaining after applying variance based filtering. This filtered set of shared descriptors now represents the dictionary of descriptors for the class of mugs.

3. Matching Descriptors in Feature Space

In the original SIFT descriptor comparison approach, every descriptor in one image is compared to every other descriptor in the second image via the distance metric (Euclidean distance) shown in Figure 4. The base approach being the same, the altered matching algorithm also takes into account the variance of the shared descriptor. When comparing a new image to the dictionary of features of an object, the dictionary is scanned to find the best match for each SIFT keypoint in the new image. Further, the variance of the shared descriptor which was found as the best match in the dictionary is used to decide whether the keypoint in the new image qualifies as a match or not.

There were a lot of experiments carried out using a variety of different decision techniques to label a keypoint as a match or not. One successful technique labeled a keypoint as a match if in feature space the Euclidean distance to the closest shared descriptor was less than the variance of that descriptor, which implies that the given keypoint is close in feature space to the shared descriptor and the keypoints that it represents in feature space.

The other technique which was quite successful was a modification of the matching technique described in **[1]** which uses the distance to the two closest shared descriptors in feature space and checks if these two distances are within a certain percentage of each other. This would indicate that the keypoint is also close to another point in the object feature space and this indicates a high probability that the keypoint belongs to the object feature space. Figure 7 shows the matches resulting from matching the mug dictionary on the right to a new mug which it has not been trained upon in the descriptor training stage.

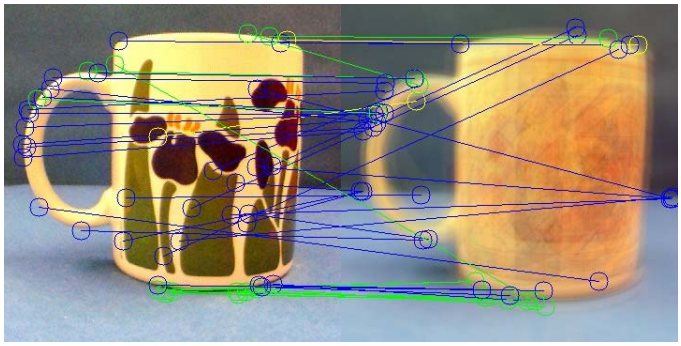


Figure 7: Matching pairs using a variance based threshold method between the mug dictionary and a new mug it has never seen before

4. Multinomial Naïve Bayes (Bag of Features)

Continuing with the analogy to text based search, the dictionary of an object class is used in conjunction with distance metrics discussed in the previous section to tabulate which individual shared descriptors or words were found in the new image or document.

Traditionally the Multinomial Naïve Bayes model for text simply counts the number of times a word from the dictionary appears in the given document. Similar to this, initially the distance metrics were used to find the count of the number of times a shared descriptor appears in the given image. This approach is called the bag of features model. Hence, the feature vector for a single image is the same length as that of the dictionary of the object class being searched and each entry in the feature matrix indicates the count of the number of times that shared descriptor was found in the new image which that row represents.

$$X \quad \varepsilon \quad R^{M*N}$$

$$X_{ij} \quad \varepsilon \quad \{0,1.....,N\}$$

$$Y \quad \varepsilon \quad R^{M}$$

$$Y_i \quad \varepsilon \quad \{0,1\}$$

Where X is the feature matrix, Y are the labels representing whether the image belongs to the object class or not. N is the length of the dictionary of the object class. M is the total number of training images. X[i,j] indicates the number of times shared descriptor 'j' from the dictionary was found in the image 'i'.

Figure 8: Representation of the feature matrix X and labels Y

The training set for the mug experiment comprised of 15 positive images which contained mugs in them and 15 negative images which were random patches from the background with and without other objects. It should also be noted that the 15 mugs used for the positive training samples were all different from the 18 mugs used for descriptor training. Figure 9(a) shows some examples of positive training images and 9(b) shows examples from the negative training images.

(a)

(b)

Figure 9: Training Set (a) Examples of Positive Images (b) Examples of Negatives Images

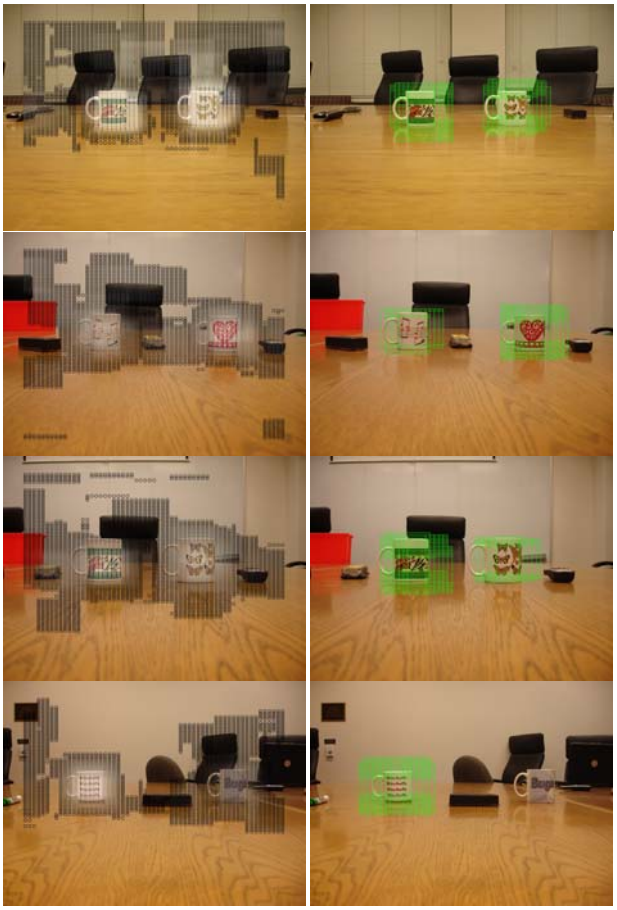
The training set is used to compute the prior probability of a sample being positive or negative and also the individual positive and negative probabilities for each shared descriptor, as is done in the text classification model of Multinomial Naïve Bayes.

After testing the above defined Multinomial Naïve Bayes classifier on a set of test images, the simple metric of simply counting the number of times a shared descriptor was found in the image did not turn out to be very accurate, since a group of false but weak features could generate a false positive. Further experiments were conducted with using different metrics to create the feature vector for a given image. One of the successful ones takes the contribution of every keypoint on the image to the feature vector instead of just the positive matches.

The count of positive matches is now replaced by the sum of the distances of that particular shared descriptor to the keypoints in the image for which it is the closest in feature space. Thus, each entry in the feature matrix is now a real number and not simply an integer count. This metric allows the classifier to differentiate between a set of strong keypoints at smaller distances in feature spaces versus a set of weak keypoints at larger distances especially when both keypoints are nearly the same number.

To test the object classifier, test images were collected using a digital camera of table scenes with mugs in them. One possible limitation of SIFT is that objects within the images need to be of a certain size to get enough features off them. Thus, when attempting to recognize objects from the SIFT features of an image the objects within the image must be at least 200 by 200 pixels. Hence, the test images were all 3200 by 2700 in resolution.

In order to localize the objects within the images, the test image of the scene was dividing into closely spaced windows. The SIFT features falling within the window were then matched to the dictionary of the object and a feature vector was generated for each window within the image. The Multinomial Naïve Bayes classifier then used this feature vector and the probabilities computed during the training step to assign a probability of the occurrence of the object to each window. Thresholds were set onto these probabilities to localize areas of high probabilities which ideally would contain the object who's classifier was run.



(a) (b)

Figure 10: Results on the test images. Images in column (a) are belief maps and images in column (b) are results after applying a threshold of probability on the belief maps

5. Results

After the training step, 32 high resolution digital images of mug scenes were taken in an office environment and the process of applying probabilities to each window within those images was used to compute the belief maps shown in Column (a) of Figure 10, where white indicated high probability and black indicated low probability on a grayscale mapping to probability. Column (b) shows the result of putting a threshold on these probabilities to localize the object instances within the image. Figure 11 shows a precision-recall curve which was plotted by computing the precision and recall on the test set of 32 images by varying the threshold of probability used to localize objects in the belief map.

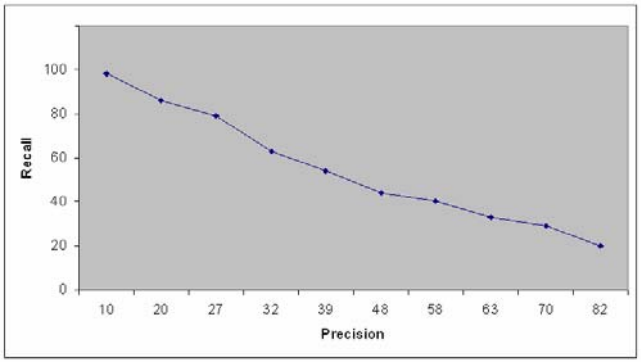


Figure 11: Precision-Recall curve for the test set of 32 images

6. Future Work

The current implementation does not support multi-view object recognition. Hence, the next step shall be to share images of an object class taken from different angles and then test them for multi-view detection.

Another aim shall be to test the Shared SIFT algorithm over a range of objects with different basic shapes to confirm that the variance filtering approach selects shared descriptors that represent a generic description of objects of a class.

To enhance the accuracy of the matching process of descriptors in feature space, rather than relying on a combined variance for all the dimensions, a metric similar to the one in figure 4 will be used to compute the variance for each descriptor in each of the 128 dimensions. This will also change the distance formula used to in feature space and distances in each dimensions will be normalized by the variance of each shared descriptor in that dimension. Further, logistic regression will be used to get variance normalized distances of matches of each shared descriptor from positive and negative training images and then use them to find a threshold between the distance to an object descriptor and other descriptors. Thus, variance normalized distances of matches in test images will be used to evaluate the sigmoid function based on the parameters obtained off the training set to classify the keypoint as a match or not.

Lastly, to improve performance for testing new images, instead of using the current approach of closely spaced windows, a better approach will be to cluster the SIFT features together into circular windows and assign probabilities only to each of these clusters. This clustering technique based on the Hough transform will lead to a significant improvement in performance.

7. Acknowledgement

This project has been made in collaboration with Stephen Gould and Prof. Andrew Ng. It would not have been possible to translate this raw idea into an actual project without Stephen's sound mathematical ideas and Prof. Andrew's guidance and direction. Paul Baumstarcks's help with the Multinomial Naïve Bayes classifier at an important stage in the algorithm's conception is also acknowledged.

References

[1] Lowe, D. G., "*Distinctive Image Features from Scale-Invariant Keypoints*", International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[2] A. Torralba, K. Murphy, and W. Freeman. "*Sharing visual features for multiclass and multiview object detection". Technical report*, CSAIL Technical report, MIT, 2004.

[3] Erik C. and Jochen T., "*Shared Features for Scalable Appearance-Based Object Recognition*", IEEE 2005 Workshop on Applications of Computer Vision (WACV 2005)

[4] Pingkun Y., Saad K., and Mubarak S., "*3D Model based Object Class Detection in An Arbitrary View*",School of Electrical Engineering and Computer Science, University of Central Florida