Object Classification and Localization Using SURF Descriptors

Drew Schmitt, Nicholas McCoy

December 13, 2011

This paper presents a method for identifying and matching objects within an image scene. Recognition of this type is becoming a promising field within computer vision with applications in robotics, photography, and security. This technique works by extracting salient features, and matching these to a database of pre-extracted features to perform a classification. Localization of the classified object is performed using a hierarchical pyramid structure. The proposed method performs with high accuracy on the Caltech-101 image database, and shows potential to perform as well as other leading methods.

1 Introduction

There are numerous applications for object recognition and classification in images. The leading uses of object classification are in the fields of robotics, photography, and security. Robots commonly take advantage of object classification and localization in order to recognize certain objects within a scene. Photography and security both stand to benefit from advancements in facial recognition techniques, a subset of object recognition.

Our method first obtains salient features from an input image using a robust local feature extractor. The leading techniques for such a purpose include the Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF).

After extracting all keypoints and descriptors from the set of training images, our method clusters these descriptors into N centroids. This operation is performed using the standard K-means unsupervised learning algorithm. The key assumption in this paper is that the extracted descriptors are independent and hence can be treated as a "bag of words" (BoW) in the image. This BoW nomenclature is derived from text classification algorithms in classical machine learning.

For a query image, descriptors are extracted using the same robust local feature extractor. Each descriptor is mapped to its visual word equivalent by finding the nearest cluster centroid in the dictionary. An ensuing count of words for each image is passed into a learning algorithm to classify the image.

A hierarchical pyramid scheme is incorporated into this structure to allow for localization of classifications within the image.

In Section 2, the local robust feature extractor used in this paper is further discussed. Section 3 elaborates on the K-means clustering technique. The learning algorithm framework is detailed in Section 4. A hierarchical pyramid scheme is presented in Section 5. Experimental results and closing remarks are provided in Section 6.

2 SURF

Our method extracts salient features and descriptors from images using SURF. This extractor is preferred over SIFT due to its concise descriptor length. Whereas the standard SIFT implementation uses a descriptor consisting of 128 floating point values, SURF condenses this descriptor length to 64 floating point values.

Modern feature extractors select prominent features by first searching for pixels that demonstrate rapid changes in intensity values in both the horizontal and vertical directions. Such pixels yield high Harris corner detection scores and are referred to as keypoints. Keypoints are searched for over a subspace of $\{x, y, \sigma\} \in \mathbb{R}^3$. The variable σ represents the Gaussian scale space at which the keypoint exists. In SURF, a descriptor vector of length 64 is constructed using a histogram of gradient orientations in the local neighborhood around each keypoint. Figure 1 shows the manner in which a SURF descriptor vector is constructed. David Lowe provides the inclined reader with further information on local robust feature extractors [1].

The implementation of SURF used in this paper is provided by the library OpenSURF [2]. OpenSURF is an

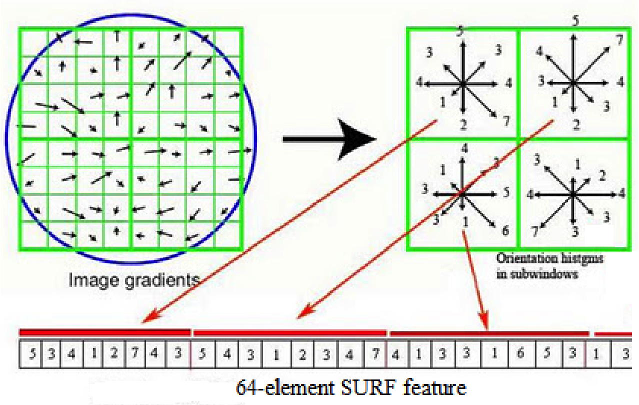

Figure 1: Demonstration of how SURF feature vector is built from image gradients.

open-source, MATLAB-optimized keypoint and descriptor extractor.

3 K-means

A key development in image classification using keypoints and descriptors is to represent these descriptors using a BoW model. Although spatial and geometric relationship information between descriptors is lost using this assumption, the inherent simplification gains make it highly advantageous.

The descriptors extracted from the training images are grouped into N clusters of visual words using K-means. A descriptor is categorized into its cluster centroid using a Euclidean distance metric. For our purposes, we choose a value of $N = 500$. This parameter provides our model with a balance between high bias (underfitting) and high variance (overfitting).

For a query image, each extracted descriptor is mapped into its nearest cluster centroid. A histogram of counts is constructed by incrementing a cluster centroid's number of occupants each time a descriptor is placed into it. The result is that each image is represented by a histrogram vector of length N. It is necessary to normalize each histogram by its L2-norm to make this procedure invariant to the number of descriptors used. Applying Laplacian smoothing to the histogram appears to improve classification results.

K-means clustering is selected over Expectation Maximization (EM) to group the descriptors into N visual words. Experimental methods verify the computational efficiency of K-means as opposed to EM. Our specific application necessitates rapid training and image classification, which precludes the use of the slower EM algorithm.

4 Learning Algorithms

Naive Bayes and Support Vector Machine (SVM) supervised learning algorithms are investigated in this paper. The learning algorithms are used to classify an image using the histogram vector constructed in the K-means step.

4.1 Naive Bayes

A Naive Bayes classifier is applied to this BoW approach to obtain a baseline classification system. The probability $\phi_{y=c}$ that an image fits into a classification c is given by

$$\phi_{y=c} = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = c\}. \tag{1}$$

Additionally, the probability $\phi_{k|y=c}$, that a certain cluster centroid, k, will contain a word count, x_k, given that it is in classification c, is defined to be

$$\phi_{k|y=c} = \frac{\left(\sum\limits_{i=1}^{m} 1\{y^{(i)} = c\}x_k^{(i)}\right) + 1}{\left(\sum\limits_{i=1}^{m} 1\{y^{(i)} = c\}n_i\right) + N}. \tag{2}$$

Laplacian smoothing accounts for the null probabilities of "words" not yet encountered. Using Equation 1 with Equation 2, the classification of a query image is given by

$$\arg\max_{c}\left(\phi_{y=c}\prod_{i=1}^{n}\phi_{i|y=c}\right). \tag{3}$$

4.2 SVM

A natural extension to this baseline framework is to introduce an SVM to classify the image based on its BoW. Our investigation starts by considering an SVM with a linear kernel

$$K(x, y) = x^T y + c, \tag{4}$$

due to its simplicity and computational efficiency in training and classification. An intrinsic flaw of linear kernels is that they are unable to capture subtle correlations between separate words in the visual dictionary of length N.

To improve on the linear kernel's performance, non-linear kernels are considered in spite of their increased complexity and computation time. More specifically the χ^2 kernel given by

$$K(x,y) = 1 - 2\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}, \tag{5}$$

is implemented.

Given that an SVM is a binary classifier, and it is often desirable to classify an image into more than two distinct groups, multiple SVM's must be used in conjunction to produce a multiclass classification.

A one-vs-one scheme can be used in which a different SVM is trained for each combination of individual classes. An incoming image must be classified using each of these different SVM's. The resulting classification of the image is the class that tallies the most "wins". The one-vs-one scheme involves making $\binom{N}{2}$ different classifications for N classes, which grows factorially with the number of classes. This scheme also suffers from false positives if an image is queried that does not belong to any of the classes.

A more robust scheme is the one-vs-all classification system in which an SVM is trained to classify an image as either belonging to class c, or belonging to class $\neg c$. For the training data $\{(x_i, y_i)\}_{i=1}^{m}$, $y_i \in 1, ..., N$, a multiclass SVM aims to train N separate SVM's that optimize the dual optimization problem

$$\max_{a} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x^{(i)}, x^{(j)}), \tag{6}$$

using John Platt's SMO algorithm [3]. In Equation 6, $K(x,z)$ corresponds to one of the Kernel functions discussed above.

A query image is then classified using

$$sgn\left\{\sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, z)\right\}, \tag{7}$$

where $sgn(x)$ is an operator that returns the sign of its argument and z is the query vector of BoW counts.

Figure 2 represents this concept visually. When the query image is of class A, the A-vs-all SVM will classify the image correctly, and thus the overall output will place the image into class A. When the query image is of a different class, D, which is not already existent in the class structure, the query will always fall into the "all" class on the individual SVM's. Hence, the query will not be falsely categorized into any class.

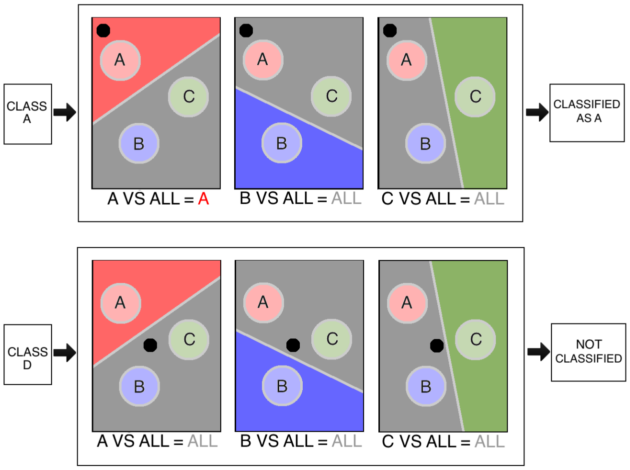

Figure 2: Portrayal of one-vs-all SVM. When query image is of type A, the A-vs-all SVM will correctly classify it. When the query image is not of class A, B, or C, it will likely not be classified into any.

It is important to reiterate that each multiclass SVM only distinguishes between classes c and $\neg c$. A different SVM is trained in this manner for each class. Thus, the number of SVM's needed in a one-vs-all scheme only grows linearly with the number of classes, N. This system also does not suffer from as many false positives because images that do not belong to any of the classes are usually classified as such in each individual SVM.

The specific multiclass SVM implementation used in this paper was MATLAB's built-in version as described by Kecman [4].

5 Object Localization

The methods described thus far are sufficient for the role of classifying an image into a class when an object is prominently displayed in the forefront of the image. However, in the case when the desired object is a small subset of the overall image, this object classification algorithm will fail to classify it correctly. Additionally, there is mo-

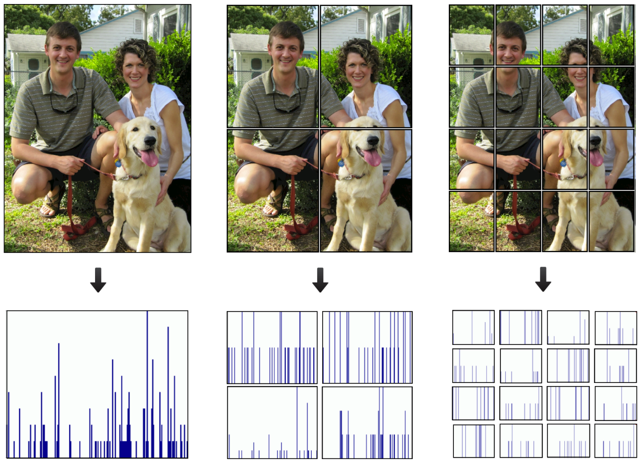

Figure 3: Visual representation of partitioning an image into sub-images and constructing the histograms.

tivation to localize an object in a scene using classification techniques. The solution to these shortcomings is object localization using a hierarchical pyramid scheme. Figure 3 illustrates the general idea behind extracting descriptors using a pyramid scheme.

First, the set of image descriptors, D, are extracted from the image using SURF. Next, the image is segmented into L pyramid levels, where L is a user-selected parameter that controls the granularity of the localization search. Each level l, has subsections $0 \leq i \leq 4^{(l-1)}$, where $0 \leq l \leq (L-1)$. At each level l, the entire set of image descriptors, D, are segmented into a subgroup $d \in D$ for section i which can be found as

$$i = \left[\text{idiv}\left(\frac{p.col - 1}{\frac{C}{2^l}} \right) + \text{idiv}\left(\frac{p.row - 1}{\frac{R}{2^l}} \right) \right] 2^l + 1, \quad (8)$$

for a given pixel p. The notation $\text{idiv}(x)$ represents an integer division operator. The symbols R and C are the maximum number of rows and columns, respectively, in the original image. Then, for pixel p the votes at each level of the pyramid can be tallied into an Nx1 map computed using

$$\text{voteMap}(c) = \sum_{l=0}^{L-1} 2^{l-1} \mathbf{1}\{\text{label}_{pyr}(l,i) = c\}. \quad (9)$$

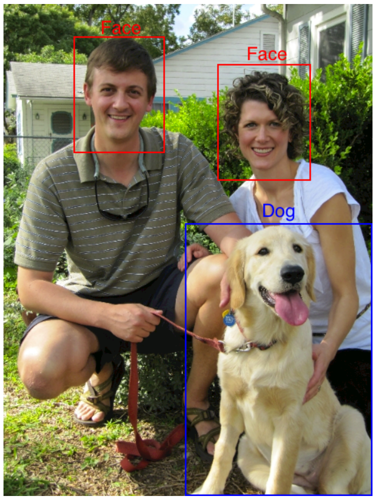

Figure 4: Results showing both image classification and localization.

The resulting effect is that pixel p is most highly influenced by the label of its lowest-level containing subsection, in $l = (L-1)$, and less influenced by the label of its highest-level containing subsection, in $l = 0$. The resulting label given to pixel p can then be calculated as

$$\text{label}_{pix}(p) = \arg\max_{c} \text{voteMap}(c). \quad (10)$$

6 Results and Future Work

Figure 4 shows the classification and localization results of our proposed algorithm on a generic image consisting of multiple classes of objects.

A more rigorous test of our method was done using a subset of the CalTech-101 database [5]. Images falling into the four categories of airplanes, cars, motorbikes, and faces were trained and tested using our method. Figure 5 shows the improvement in percent correct classifications in classification of Naive Bayes, linear SVM, and nonlinear SVM as the training set size increases.

The f-score, computed using the precision, P, and recall, R, of the algorithm by

$$f = \frac{2PR}{P+R}, \quad (11)$$

is perhaps a better indicator of performance because it is a statistical measure of a test's accuracy. Figure 6

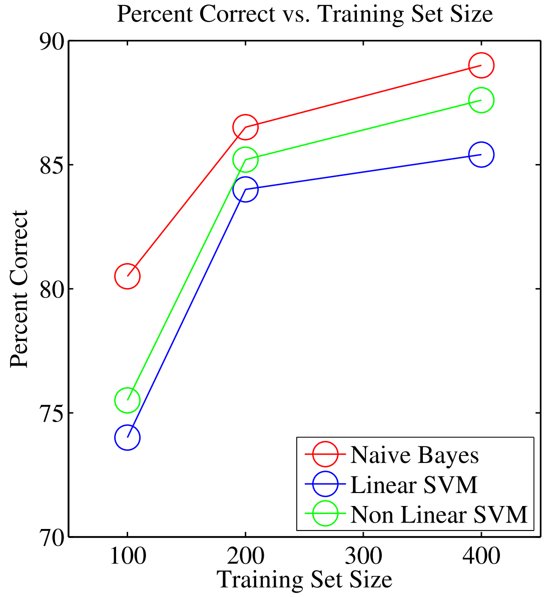

Figure 5: Percent correct classifications of supervised learning classifiers.

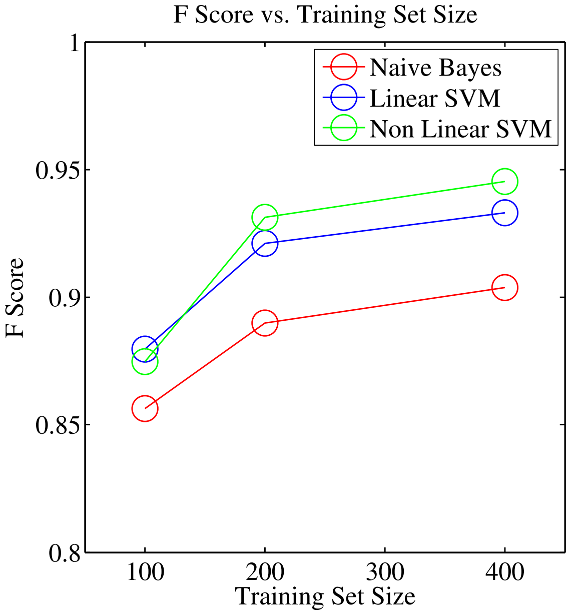

Figure 6: f-score of classifications of supervised learning classifiers.

shows a visible improvement in the f-score for all three classification algorithms as the training set size increases. The nonlinear SVM maintains the largest f-score over all training set sizes, which aligns with our hypothesized result.

Future work for this research should focus on replacing K-means with a more robust clustering algorithm. One option is Linearly Localized Codes (LLC) [6]. The LLC method performs sparse coding on extracted descriptors to make soft assignments that are more robust to local spatial translations [7]. Furthermore, there is still open-ended work to be done on the reconstruction of objects using the individually labeled pixels from the pyramid localization scheme. Hrytsyk and Vlakh present a method of conglomerating pixels into their neighboring groups in an optimal fashion [8].

References

[1] D. Lowe. Towards a computational model for object recognition in IT cortex. Proc. *Biologically Motivated Computer Vision*, pages 2031, 2000.

[2] C. Evans. Opensurf. http://www.chrisevansdev.com/computer-vision-opensurf.html, retrieved 11/04/2011.

[3] J. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, 1998.

[4] V. Kecman. Learning and Soft Computing, *MIT Press*, Cambridge, MA. 2001.

[5] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples. *CVPR*, 2004.

[6] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, 2009.

[7] T. Serre, L. Wolf, and T.Poggio. Object recognition with features inspired by visual cortex. *CVPR*, 2005.

[8] N. Hrytsyk, V. Vlakh. Method of conglomerates recognition and their separation into parts. *Methods and Instruments of AI*, 2009.