Efficient Object Localization with Gaussianized Vector Representation

Xiaodan Zhuang
Beckman Inst., ECE Dept.,
UIUC, Illinois, USA
xzhuang2@uiuc.edu

Xi Zhou
Beckman Inst., ECE Dept.,
UIUC, Illinois, USA
xizhou2@uiuc.edu

Mark A. Hasegawa-Johnson
Beckman Inst., ECE Dept.,
UIUC, Illinois, USA
jhasegaw@uiuc.edu

Thomas S. Huang
Beckman Inst., ECE Dept.,
UIUC, Illinois, USA
huang@ifp.uiuc.edu

ABSTRACT

Recently, the Gaussianized vector representation has been shown effective in several applications related to interactive multimedia, such as facial age estimation, image scene categorization and video event recognition. However, all these tasks are classification and regression problems based on the whole images. It is not yet explored how this representation can be efficiently applied in the object localization problem, which reveals the locations and sizes of the objects. In this paper, we present an efficient object localization approach for the Gaussianized vector representation, following a branch-and-bound search scheme introduced by Lampert et al. [4] In particular, we design a quality bound for rectangle sets characterized by the Gaussianized vector representation for fast hierarchical search. This bound can be obtained for any rectangle set in the image, with little extra computational cost, in addition to calculating the Gaussianized vector representation for the whole image. A localization experiment on a multi-scale car dataset shows that the proposed object localization approach based on the Gaussianized vector representation outperforms previous work using the histogram-of-keywords representation.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithm, Performance, Experimentation

IMCE'09, October 23, 2009, Beijing, China.

Keywords

Gaussianized vector representation, efficient object localization, branch and bound

1. INTRODUCTION

The Gaussian mixture model (GMM) is widely used for distribution modeling in speech recognition, speaker identification and computer vision. Gaussianized vector representation was recently proposed as an innovative image and video vector representation based on the GMM [12]. Variants of this Gaussianized vector representation have been successfully applied in several applications related to interactive multimedia, such as facial age estimation [11, 14], image scene categorization [12] and video event recognition [13].

While the Gaussianized vector representation proves effective in the above visual recognition tasks, all these are classification or regression problems working on the whole images. Another important problem is the object detection or localization problem, i.e., to find the rectangle bounding boxes for instances of a particular object with varying locations, widths and heights. However, it is not clear how to use the Gaussianized vector representation to capture localized information besides global information in an image. No work has yet explored applying the Gaussianized vector representation in the object localization problem.

A natural way to carry out localization is the sliding window approach [9]. However, an exhaustive search in an $n \times n$ image needs to evaluate $O(n^4)$ candidate bounding boxes, and is not affordable for a complicated representation such as the Gaussianized vector representation. Tricky heuristics about possible bounding box locations, widths and heights, or local optimization methods would have to be used, resulting in false estimates. This intrinsic tradeoff between performance and efficiency of the sliding window approach is not desirable particularly for applications in consumer electronics that are highly efficiency sensitive. Lampert et al. introduced a branch-and-bound search scheme[4], which finds the globally optimal bounding box efficiently without the above problems.

We present an efficient object localization approach based

on the Gaussianized vector representation. The branch-and-bound search scheme [4] is adopted to perform fast hierarchical search for the optimal bounding boxes, leveraging a quality bound for rectangle sets. We demonstrate that the quality function based on the Gaussianized vector representation can be written as the sum of contributions from each feature vector in the bounding box. Moreover, a quality bound can be obtained for any rectangle set in the image, with little computational cost, in addition to calculating the Gaussianized vector representation for the whole image.

We carry out an object localization experiment on a multi-scale car dataset. The results show the proposed object localization approach based on the Gaussianized vector representation outperforms a similar system using the branch-and-bound search based on the histogram-of-keywords representation. This suggests the Gaussianized vector representation can be effective for the localization problem besides the classification and regression problems reported previously.

The rest of this paper is arranged as follows. In Section 2, we describe the procedure of constructing Gaussianized vector representation. Section 3 details the proposed efficient localization method based on the Gaussianized vector representation. The experimental results on multi-scale car detection are reported in Section 4, followed by conclusions and discussion in Section 5.

2. GAUSSIANIZED VECTOR REPRESENTATION

The Gaussian mixture model (GMM) is widely used in various pattern recognition problems [8, 6]. Recently, the Gaussianized vector representation was proposed. This representation encodes an image as a bag of feature vectors, the distribution of which is described by a GMM. Then a GMM supervector is constructed using the means of the GMM, normalized by the covariance matrices and Gaussian component priors. A GMM-supervector-based kernel is designed to approximate Kullback-Leibler divergence between the GMMs for any two images, and is utilized for supervised discriminative learning using an SVM. Variants of this GMM-based representation have been successfully applied in several visual recognition tasks, such as facial age estimation [11, 14], scene categorization [12] and video event recognition [13].

As pointed out by [12], the success of this representation can be attributed to two properties. First, it establishes correspondence between feature vectors in different images in an unsupervised fashion. Second, it observes the standard normal distribution, and is more informative than the conventional histogram of keywords.

The Gaussianized vector representation is closely connected to the classic histogram of keywords representation. In the traditional histogram representation, the keywords are chosen by the k-means algorithm on all the features. Each feature is distributed to a particular bin based on its distance to the cluster centroids. The histogram representation obtains rough alignment between features vectors by assigning each to one of the histogram bins. Such a representation provides a natural similarity measure between two images based on the difference between the corresponding histograms. However, the histogram representation has some intrinsic limitations. In particular, it is sensitive to feature outliers, the choice of bins, and the noise level in the data. Besides, encoding high-dimensional feature vectors by a relatively small codebook results in large quantization errors and loss of discriminability.

Gaussianized vector representation enhances the histogram representation in the following ways. First, k-means clustering leverages the Euclidean distance, while the GMM leverages the Mahamalobis distance by means of the component posteriors. Second, k-means clustering assigns one single keyword to each feature vector, while the Guassinized vector representation allows each feature vector to contribute to multiple Gaussian components statistically. Third, histogram-of-keywords only uses the number of feature vectors assigned to the histogram bins, while the Gaussianized vector representation also engages the weighted mean of the features in each component, leading to a more informative representation.

2.1 GMM for feature vector distribution

We estimate a GMM for the distribution of all feature vectors in an image. The estimated GMM is a compact description of the single image, less prone to noise compared with the feature vectors. Yet, with increasing number of Gaussian components, the GMM can be arbitrarily accurate in describing the underlying feature vector distribution. The Gaussian components impose an implicit multimode structure of the feature vector distribution in the image. When the GMMs for different images are adapted from the same global GMM, the corresponding Gaussian components imply certain correspondence.

In particular, we obtain one GMM for each image in the following way.

First, a global GMM is estimated using feature vectors extracted from all training images, regardless of their labels. Here we denote z as a feature vector, whose distribution is modeled by a GMM, a weighted linear combination of K unimodal Gaussian components,

$$p(z;\Theta) = \sum_{k=1}^{K} w_k \mathcal{N}(z;\mu_k^{global},\Sigma_k).$$

$\Theta = \{w_1, \mu_1^{global}, \Sigma_1, \cdots\}$, w_k, μ_k and Σ_k are the weight, mean, and covariance matrix of the kth Gaussian component,

$$\mathcal{N}(z;\mu_k,\Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}}\, e^{-\frac{1}{2}(z-\mu_k)^T\Sigma_k^{-1}(z-\mu_k)}. \quad (1)$$

We restrict the covariance matrices Σ_k to be diagonal [7], which proves to be effective and computationally economical.

Second, an image-specific GMM is adapted from the global GMM, using the feature vectors in the particular image. This is preferred to direct seperate estimation of image-specific GMMs for the following reasons. 1) It improves robust parameter estimation of the image specialized GMM, using the comparatively small number of feature vectors in the single image. 2) The global GMM learnt from all training images may provide useful information for

the image specialized GMM. 3) As mentioned earlier, it establishes correspondence between Gaussian components in different images-specifc GMMs. For robust estimation, we only adapt the mean vectors of the global GMM and retain the mixture weights and covariance matrices. In particular, we adapt an image-specific GMM by *Maximum a Posteriori* (MAP) with the weighting all on the adaptation data. The posterior probabilities and the updated means are estimated as

$$Pr(k|z_j) = \frac{w_k \mathcal{N}(z_j; \mu_k^{global}, \Sigma_k)}{\sum_{k=1}^{K} w_k \mathcal{N}(z_j; \mu_k^{global}, \Sigma_k)}, \quad (2)$$

$$\mu_k = \frac{1}{n_k} \sum_{j=1}^{H} Pr(k|z_j) z_j, \quad (3)$$

where n_k is a normalizing term,

$$n_k = \sum_{j=1}^{H} Pr(k|z_j), \quad (4)$$

and $Z = \{z_1, \ldots, z_H\}$ are the feature vectors extracted from the particular image.

As shown in Equation 2, the image-specific GMMs leverage statistical membership of each feature vector among multiple Gaussian components. This sets the Gaussianized vector representation apart from the histogram of keyword representation which originally requires hard membership in one keyword for each feature vector. In addition, Equation 3 shows that the Gaussianized vector representation encodes additional information about the feature vectors statistically assigned to each Gaussian component, via the means of the components.

Given the computational cost concern for many consumer electronics applications, another advantage of using GMM to model feature vector distribution is that efficient approximation exists for GMM that does not significantly degrade its effectiveness. For example, we can prune out Gaussian components with very low weights in the adapted image-specific GMMs. Another possibility is to eliminate the additions in Equation 3 that involves very low priors in Equation 2. Neither of these approaches significantly degrades GMM's capability to approximate a distribution [8].

2.2 Discriminative learning

Suppose we have two images whose ensembles of feature vectors, Z_a and Z_b, are modeled by two adapted GMMs according to Section 2.1, denoted as g_a and g_b. A natural similarity measure is the approximated Kullback-Leibler divergence

$$D(g_a||g_b) \leq \sum_{k=1}^{K} w_k D(\mathcal{N}(z; \mu_k^a, \Sigma_k)||\mathcal{N}(z; \mu_k^b, \Sigma_k)), \quad (5)$$

where μ_k^a denotes the adapted mean of the kth component from the image-specific GMM g_a, and likewise for μ_k^b. The right side of the above inequality is equal to

$$d(Z_a, Z_b) = \frac{1}{2} \sum_{k=1}^{K} w_k (\mu_k^a - \mu_k^b)^T \Sigma_k^{-1} (\mu_k^a - \mu_k^b). \quad (6)$$

$d(Z_a, Z_b)^{\frac{1}{2}}$ can be considered as the Euclidean distance in another high-dimensional feature space,

$$d(Z_a, Z_b) = \|\phi(Z_a) - \phi(Z_b)\|^2$$
$$\phi(Z_a) = [\sqrt{\frac{w_1}{2}} \Sigma_1^{-\frac{1}{2}} \mu_1^a; \cdots; \sqrt{\frac{w_K}{2}} \Sigma_K^{-\frac{1}{2}} \mu_K^a]. \quad (7)$$

Thus, we obtain the corresponding kernel function

$$k(Z_a, Z_b) = \phi(Z_a) \bullet \phi(Z_b). \quad (8)$$

A Support Vector Machine (SVM) is used with the above kernel to distinguish objects from backgrounds. The classification score for a test image is

$$g(Z) = \sum_t \alpha_t k(Z, Z_t) - b, \quad (9)$$

where α_t is the learnt weight of the training sample Z_t and b is a threshold parameter. $k(Z, Z_i)$ is the value of a kernel function for the training Gaussianized vector representation Z_i and the test Gaussianized vector representation Z, y_i is the class label of Z_i used in training,

$$y_i = \begin{cases} +1, & \text{if } Z_i \text{ is the object} \\ -1, & \text{if } Z_i \text{ is not the object} \end{cases} \quad (10)$$

The support vectors and their corresponding weights are learned using the standard quadratic programming optimization process. In this work, we use the SVM training tools implemented in Libsvm [2].

3. LOCALIZATION WITH GAUSSIANIZED VECTOR REPRESENTATION

3.1 Branch-and-bound search

Localization of an object is essentially to find the subarea in the image on which a quality function f achieves its maximum, over all possible subareas. One way to define these subareas is the bounding box, which encodes the location, width and height of an object with four parameters, i.e., the top, bottom, left and right coordinates (t, b, l, r).

The sliding window approach is most widely used in object localization with bounding boxes [9, 3]. To find the bounding box where the quality function f reaches its global maximum, we need evaluate the function on all possible rectangles in the image, whose number is on the order of $O(n^4)$ for an $n \times n$ image. To reduce the computational cost, usually only rectangles at a coarse location grid and of a small number of possible widths and heights are considered. On the other hand, different approaches can be adopted to use a local optimum to approximate the global one, when the quality function f has certain properties, such as smoothness. All these approaches make detection tractable at the risk of missing the global optimum, and with demand for well informed heuristics about the possible location and sizes of the object.

The branch-and-bound search scheme was recently introduced [4] to find the globally optimal bounding box without the heuristics and assumptions about the property of the quality function. It hierarchically splits the parameter space of all the rectangles in an image, and discards large

parts if their upper bounds fall lower than an examined rectangle.

For localization based on bounding boxes, a set of rectangles is encoded with $[T, B, L, R]$, each indicating a continual interval for the corresponding parameter in (t, b, l, r). The approach starts with a rectangle set containing all the rectangles in the image, and terminates when one rectangle is found that has a quality function no worse than the bounds $\hat{f}$ of any other rectangle set.

At every iteration, the parameter space $[T, B, L, R]$ is split along the largest of the four dimension, resulting in two rectangle sets both pushed into a queue together with their upper bounds. The rectangle set with the highest upper bound is retrieved from the queue for the next iteration.

The steps of the branch-and-bound search scheme can be summarized as follows:

1. Initialize an empty queue Q of rectangle sets. Initialize a rectangle set $\mathbf{R}$ to be all the rectangles: T and B are both set to be the complete span from zero to the height of the image. L and R are both set to be the complete span from zero to the width of the image.

2. Obtain two rectangle sets by split the parameter space $[T, B, L, R]$ along the largest of the four dimension.

3. Push the two rectangle sets in Step 2 into queue Q with their respective quality bound.

4. Update R with the rectangle set with the highest quality bound in Q.

5. Stop and return R if $\mathbf{R}$ contains only one rectangle R. Otherwise go to Step 2.

The quality bound $\hat{f}$ for a rectangle set $\mathbf{R}$ should satisfy the following conditions:

1. $\hat{f}(\mathbf{R}) \geq max_{R \in \mathbf{R}} f(R)$

2. $\hat{f}(\mathbf{R}) = f(R)$, if R is the only element in $\mathbf{R}$

Critical for the branch-and-bound scheme is to find the quality bound $\hat{f}$. Given the proven performance of the Gaussianized vector representation in classification tasks shown in previous work [11, 13, 14, 12], we are motivated to design a quality bound based on this representation, to enable localization based on this representation.

3.2 Quality function

For the Gaussianized vector representation, the binary classification score in Equation 9 informs the confidence that the evaluated image subarea contains the object instead of pure background. Therefore, we can use this score as the quality function for the Gaussianized vector representation.

In particular, according to Equation 8 and Equation 9, the quality function f can be defined as follows,

$$f(Z) = g(Z) = \sum_t \alpha_t \phi(Z) \bullet \phi(Z_t) - b, \tag{11}$$

which can be expanded using Equation 7,

$$\begin{aligned} f(Z) &= \sum_t \alpha_t \sum_{k=1}^{K} \sqrt{\frac{w_c}{2}} \Sigma_c^{-\frac{1}{2}} \mu_k \\ &\quad \bullet \sqrt{\frac{w_k}{2}} \Sigma_c^{-\frac{1}{2}} \mu_k^i - b \\ &= \sum_t \alpha_t \sum_{k=1}^{K} \frac{w_k}{2} \Sigma_k^{-1} \mu_k \bullet \mu_k^t - b. \end{aligned} \tag{12}$$

According to Equation 3, the adapted mean of an image-specifc GMM is the sum of the feature vectors in the image, weighted by the corresponding posterior. Therefore,

$$\begin{aligned} f(Z) &= \sum_t \alpha_t \sum_{k=1}^{K} \frac{w_k}{2} \Sigma_k^{-1} \frac{1}{n_k} \sum_{j=1}^{H} Pr(k|z_j) z_j \bullet \mu_k^t - b. \\ &= \sum_{j=1}^{H} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} Pr(k|z_j) z_j \bullet \frac{w_k}{2} \Sigma_k^{-1} \sum_t \alpha_t \mu_k^t \right\} - b. \end{aligned} \tag{13}$$

3.3 Quality bound

We define the "per feature vector contribution" as the contribution of each feature vector in a subarea to the confidence that this subarea is the concerned object. In particular, the "per feature vector contribution" is defined as in Equation 14.

$$W_j = \sum_{k=1}^{K} \frac{1}{n_k} Pr(k|z_j) z_j \bullet \frac{w_k}{2} \Sigma_k^{-1} \sum_t \alpha_t \mu_k^t. \tag{14}$$

Therefore, Equation 13 can be rewritten as Equation 15, showing that the quality function can be viewed as the sum of contribution from all involved feature vectors.

$$f(Z) = \sum_j W_j - b. \tag{15}$$

Given a test image, if we approximate the term n_k with their values calculated on the whole image, the per feature vector contributions $W_j, j \in 1, ..., H$ are independent from the bounding box within the test image. This means that we can precompute W_j and evaluate the quality function on different rectangles by summing up those W_j that fall into the concerned rectangle.

We design a quality bound for the Gaussianized vector representation in a way similar to the quality bound for histogram of keywords proposed in [4]. For a set of rectangles, the quality bound is the sum of all positive contributions from the feature vectors in the largest rectangle and all negative contributions from the feature vectors in the smallest rectangle. This can be formulated as

$$\begin{aligned} \hat{f}(\mathbf{R}) &= \sum_{W_{j_1} \in R_{max}} W_{j_1} \times (W_{j_1} > 0) \\ &\quad + \sum_{W_{j_2} \in R_{min}} W_{j_2} \times (W_{j_2} < 0). \end{aligned} \tag{16}$$

where $[T, B, L, R]$ are the intervals of t, b, l, r and R_{max} and R_{min} are the largest and the smallest rectangles.

We demonstrate that Equation 16 satisfies the conditions of a qualify bound for the branch-and-bound search scheme defined in Section 3.1.

First, the proposed $\hat{f}(\mathbf{R})$ is an upper bound for all rectangles in the set $\mathbf{R}$. In particular, the qualify function evaluated on any rectangle R can be written as the sum of postive contributions and negative contributions from feature vectors in this rectangle,

$$f(R) = \sum_{W_{j_1} \in R} W_{j_1} \times (W_{j_1} > 0) + \sum_{W_{j_2} \in R} W_{j_2} \times (W_{j_2} < 0). \quad (17)$$

Obviously, given a rectangle set $\mathbf{R}$, the first term in Equation 17 is maximized by taking all the positive contributions from the largest rectangle in the set. The second term in Equation 17 is negative and its absolute value can be minimized by taking all the negative contributions in the smallest rectangle.

Second, when the rectangle set $\mathbf{R}$ contains only one rectangle, $R_{min} = R_{max} = R$. Equation 16 equals Equation 17,

$$\hat{f}(\mathbf{R}) = f(R).$$

This quality bound defined by Equation 16 is used in the branch-and-bound scheme discussed in Section 3.1 to achieve fast and effective detection and localization. Note that since the bound is based on sum of per feature vector contributions, the approach can be repeated to find multiple bounding boxes in an image, after removing those features claimed by the previously found boxes. This avoids the problem of finding multiple non-optimal boxes near a previously found box as in the sliding window approach.

Note that estimating W_j in Equation 14 involves no more computation than the calculation in a binary classifier using the Gaussianized vector representation of the whole image. To further expedite the localization, we can use two integral images [10] to speed up the two summations in Equation 16 respectively. This makes the calculation of $\hat{f}(\mathbf{R})$ independent from the number of rectangles in the set $\mathbf{R}$.

4. EXPERIMENT

In this paper, we carry out an object localization experiment using the proposed efficient object localization approach based on the Gaussinized vector representation. We compare the detection performance with a similar object localization system based on the generic histogram of keywords.

4.1 Dataset

We use a multi-scale car dataset[1] for the localization experiment. There are 1050 training images of fixed size 100×40 pixels, half of which exactly showing a car and the other half showing other scenes or objects. Since the proposed localization approach has the benefit of requiring no heuristics about the possible locations and sizes of the bounding boxes, we use a test set consisting of 107 images with varying resolution containing 139 cars in sizes between 89×36 and 212×85. This dataset also includes ground truth annotation for the test images in the form of bounding rectangles for all the cars. The training set and the multi-scale test set are consistent with the setup used in [4].

A few sample test images of the dataset is shown in figure 1. Note that some test images contain multiple cars and partial occlusion may exist between different cars as well as between a car and a "noise" object, such as a bicyclist, a pedestrian or a tree.

Figure 1: Sample images in the multi-scale car dataset

4.2 Metric

The localization performance is measured by recall, precision and F-measure, the same way as in [1] and [4]. A hypothesized bounding box is counted as a correct detection if its location coordinates and size lie within an ellipsoid centered at the true coordinates and size. The axes of the ellipsoid are 25% of the true object dimensions in each direction. For multiple detected bounding boxes satisfying the above criteria for the same object, only one is counted as correct and the others are counted as false detections.

4.3 Gaussianized vectors

The feature vectors for each image are extracted as follows. First, square patches randomly sized between 4×4 and 12×12 are extracted on a dense pixel grid. Second, an 128-dimensional SIFT vector is extracted from each of these square patches. Third, each SIFT vector is reduced to 64 dimensions by Principal Component Analysis. Therefore, each image is converted to a set of 64-dimensional feature vectors.

These feature vectors are further transformed into Gaussianized vector representations as described in Section 2. Each image is therefore represented as a Gaussianized vector. In particular, we carry out the experiment with 32, 64, 128 Gaussian components in the GMMs respectively.

4.4 Results

To keep the setting the same as in [4], we search each test image for the three best bounding boxes, each affiliated with the quality function score. In particular, the branch-and-bound search scheme is applied to each test image three times. After each time, those features claimed by the found boxes are removed as discussed in Section 3.1.

The ROC curves, characterizing precision vs. recall, are obtained by changing the threshold on the quality function score for the found boxes. The equal error rate (EER) equals 1 − F-measure when precision equals recall.

The ROC curves and the EER are presented in Figure 2 and Figure 3 respectively. We compare the results with a localization system using the same banch-and-bound scheme, but based on the generic histogram of keywords with 1000 entry codebook generated from SURF descriptors at different scales on a dense pixel grid [4].

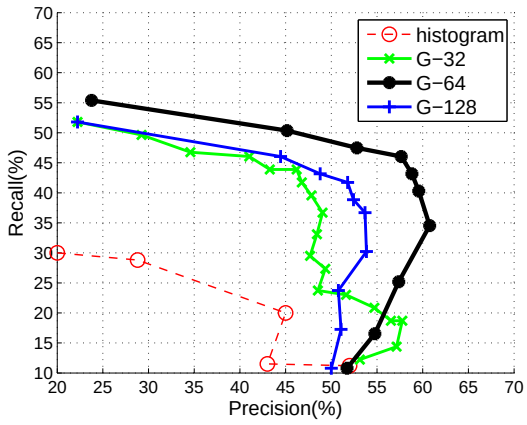

Figure 2: ROC curves for multi-scale car detection. ("G-n" denotes the result using n components in the Gaussianized vector representation. "Histogram" denotes the performance using the generic histogram-of-keywords approach by Lampert et al.)

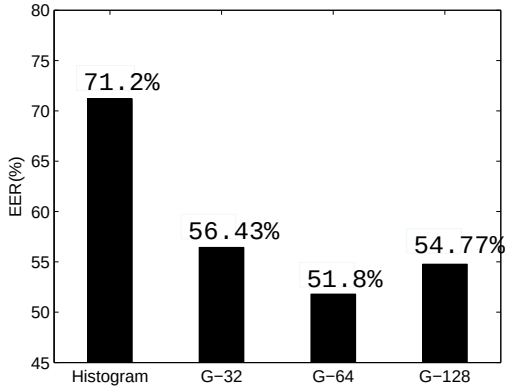

Figure 3: Equal Error Rates for multi-scale car detection.

We can see that the Gaussianized vector representation outperforms the histogram of keywords in this multi-scale object detection task. In particular, using 64 Gaussian components gives the best performance.

In Figure 4, we present a few examples of correct detection and erroneous detection in the best setting in Figure 3. Each test image is accompanied by a "per-feature-contribution" map. Negative and positive contributions are denoted by blue and red, with the color saturation reflecting absolute values. The quality function evaluated on a bounding box is the sum of all the per-feature-contributions, as discussed in Section 3.

The examples of correct detection demonstrate that the system can effectively localize one or multiple objects in complex backgrounds.

The three examples of erroneous detection probably occur for different reasons: 1) The car is a bit atypical, resulting in fewer features with highly positive contributions. 2) The two cars and some ground texture form one rectangle area with highly positive contributions, bigger than the two true bounding boxes. 3) The car is highly confusable with the background, resulting in too many highly negative contributions everywhere, preventing any rectangle to yield a high value for the quality function.

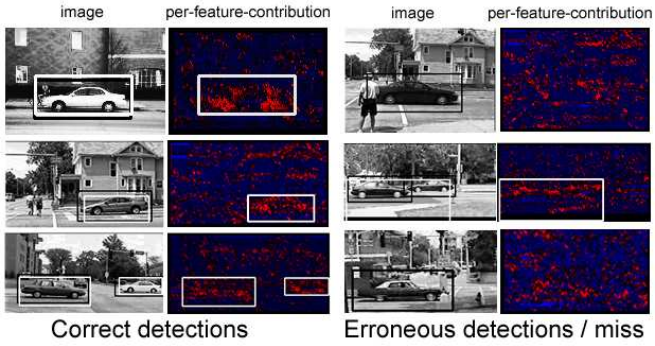

Figure 4: Examples of good and bad localization based on Gaussianized vector representation. (The black and white bounding boxes in the images are the ground truth and the hypotheses respectively. Best viewed in color.)

5. CONCLUSION & DISCUSSION

In this work, we present an efficient object localization approach based on the Gaussianized vector representation. We design a quality bound for rectangle sets characterized by the Gaussianized vector representation. This bound can be obtained for any rectangle set within the image boundaries, with little extra computational cost, in addition to calculating the Gaussianized vector representation for the whole image classification problem. Adopting the branch-and-bound search scheme, we leverage the proposed quality bound for fast hierarchical search. The proposed object localization approach based on the Gaussianized vector representation outperforms a similar localization system based on the generic histogram-of-keywords representation on a multi-scale car dataset. This is the first work using the Gaussianized vector representation in object detection and localization.

Extension to the histogram-of-keywords, in particular, the spatial pyramid kernel, has been introduced to improve both classification and localization performance [5, 4]. Beyond the scope of this paper, we have extended the Gaussianized vetor representation into a spatial hierarchical variant with superior results in classification and regression tasks. A natural extension of this work will be to derive a similar quality bound based on the hierarchical Gaussianized vector for further improved localization performance. We have further research in this direction as future work.

6. ACKNOWLEDGMENTS

This research is funded by NSF grant IIS 08-03219.

7. REFERENCES

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] C. Chang and C. Lin. LIBSVM: A library for support vector machines. http://www.csie.ntu.edu.tw/˜cjlin/libsvm, 2001.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[4] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. of CVPR*, 2008.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[6] H. Permuter, J. Francos, and I. Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 3, pages III–569–72 vol.3, April 2003.

[7] D. Reynolds, T. Quatieri, and R. Dunn. Speaker Verification using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[9] H. A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. In *NIPS 8*, pages 875–811, 1996.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, 2001.

[11] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. S. Huang. Regression from patch-kernel. In *CVPR*, 2008.

[12] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. S. Huang. A Novel Gaussianized Vector Representation for Natural Scene Categorization. In *ICPR*, 2008.

[13] X. Zhou, X. Zhuang, S. Yan, S. Chang, M. Hasegawa-Johnson, and T. S. Huang. SIFT-Bag Kernel for Video Event Analysis. In *ACM Multimedia*, 2008.

[14] X. Zhuang, X. Zhou, M. Hasegawa-Johnson, and T. S. Huang. Face Age Estimation Using Patch-based Hidden Markov Model Supervectors. In *ICPR*, 2008.