Finding Better Matches: Improving Image Retrieval with EFM-HOG

Sugata Banerji
Mathematics and Computer Science
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045, USA
Email: banerji@lakeforest.edu

Ryan R. Zunker
Mathematics and Computer Science
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045, USA
Email: zunkerrr@lakeforest.edu

Atreyee Sinha
Computing and Information Sciences
Edgewood College
1000 Edgewood College Drive
Madison, WI 53711, USA
Email: asinha@edgewood.edu

***Abstract*—Retrieving images from a dataset, which are similar to a query image, is an important high-level vision problem. Different tasks define similarity based on various low-level features, such as shape, color, or texture. In this article, we focus on the problem of image retrieval of similarly shaped objects, with the query being an object selected from a test image at run-time. Towards that end, we propose a novel shape representation and associated similarity measure, which exploits the dimensionality reduction and feature extraction methods of Principal Component Analysis (PCA) and Enhanced Fisher Model (EFM). We demonstrate the effectiveness of this representation on three shape-matching problems using multiple large-scale image datasets and also compare its retrieval performance with the Histograms of Oriented Gradients (HOG). Furthermore, to test the performance of our presented descriptor on the non-trivial task of image-based geo-localization, we create a large-scale image dataset and conduct extensive experiments on it. Finally, we establish that our proposed EFM-HOG not only works well on this new dataset, but also significantly improves upon the conventional HOG results.**

Keywords–Histogram of Oriented Gradients; Enhanced Fisher Model; Content-Based Image Retrieval; Shape Matching; EFM-HOG.

I. Introduction

With the enormous popularity of digital devices equipped with cameras, along with the wide access to high speed Internet and cloud storage, several applications based on image search and retrieval have emerged. Such applications include augmented reality, geo-localization, security and defense, educational uses, to name a few. Billions of images are uploaded and shared over social media and web sharing platforms everyday, giving rise to a greater need for systems that can retrieve images similar to a query image from a dataset. Traditional approaches of content-based image retrieval are based upon low level cues such as shape, color and texture features. In this extended work, we address three image retrieval problems, which are all based on shape similarity. Specifically, we select a window surrounding an object of interest from a query image and want to be able to retrieve other images in the dataset, which have similarly shaped objects. Towards that end, we investigate and propose a novel EFM-HOG representation and retrieval technique [1] that is based on shape features, dimensionality reduction, and discriminant analysis. It is also robust to the slight changes in the window object selection.

The Histograms of Oriented Gradients (HOG) feature is very popular for shape matching. Simple HOG matching, however, poses significant challenges in effective image retrieval due to the fact that the apparent shape of the query object may change considerably between images due to differences in lighting, camera parameters, viewing angle, scale and occlusion. In this work, we introduce a novel technique of improving the HOG features to reduce the number of false matches.

Shape matching can be used in a variety of different scenarios and we demonstrate the effectiveness of the proposed method in three such scenarios here. The first is a simple object retrieval problem where the goal is to retrieve images of objects belonging to the same class as, or more broadly speaking, similar in shape to the query image. We use the publicly available PASCAL VOC 2012 [2] image dataset for this task. The second task that we use our method for is building image retrieval and landmark recognition. Here it is important to fetch not other similarly-shaped buildings but other pictures of the exact same building. It is also important to fetch multiple instances of the building in top search results so that the building can be identified without doubt using a k-nearest neighbors method. We use another large public database, the Oxford Buildings dataset [3] for this task.

The third problem that we try to solve using the proposed method is that of image-based geo-localization at the scale of a city. To make this problem more challenging than landmark recognition, we created a new dataset [4] based on Google StreetView images of the city of Lake Forest, Illinois, USA. This dataset has $10,000$ images and it is more challenging than a similar dataset built from a big city due to an extremely high amount of vegetation cover. Here, we had to address the problem of isolating the buildings in the images and discard most of the trees and other things. We had to design a form of coarse semantic segmentation as a preprocessing step on this dataset for this purpose.

The rest of this paper is organized as follows. Section II presents a short survey of other methods employed in shape-based image retrieval, with a brief mention of other researchers working on related problems. Section III and its subsections outline in detail the method proposed in this paper. The problems addressed and the datasets used are described in detail in Section IV. The experiments performed and results obtained are detailed in Section V. Finally, we list our conclusions and directions for future research in Section VI.

II. Related Work

The HOG feature vector [5], proposed nearly two decades ago for pedestrian detection problems, has been very popular

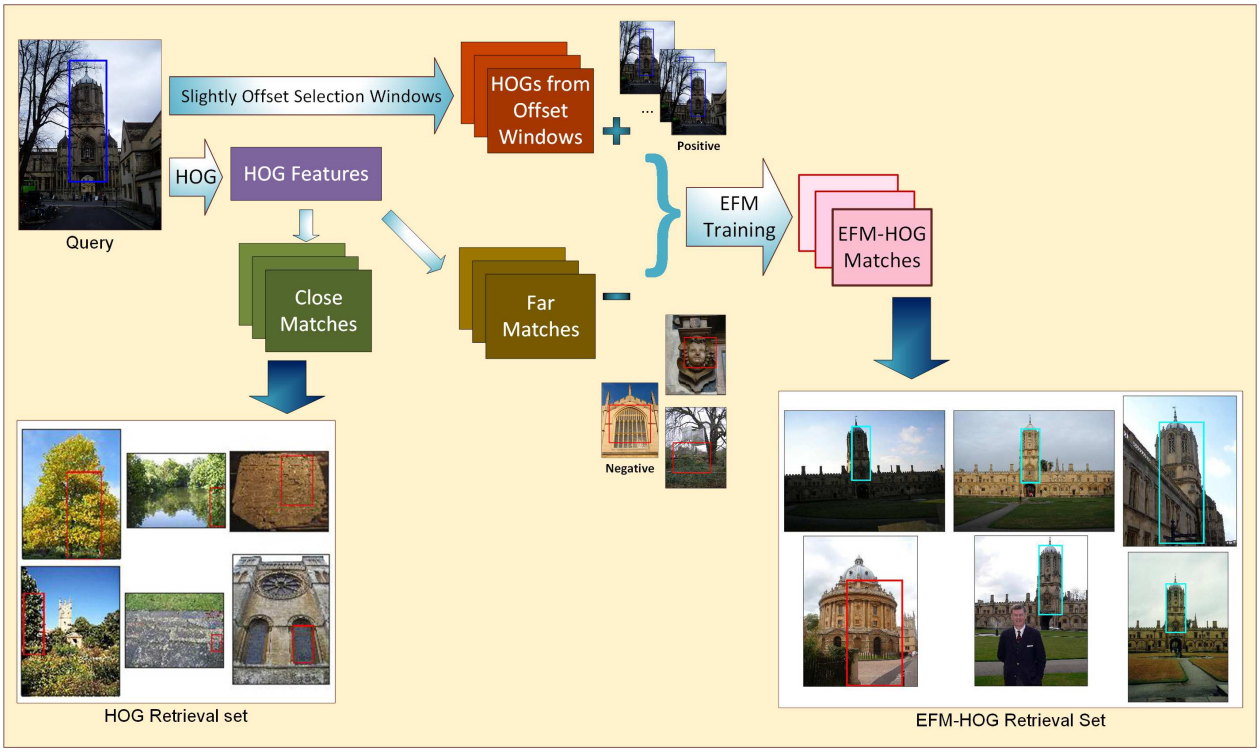

Figure 1: The process of generating the retrieval set using the proposed EFM-HOG match technique.

among Computer Vision researchers for representing shape. It has successfully been combined with other techniques [6] and fused with other descriptors [7] for classifying both indoor and outdoor scene images. HOG has also given rise to other extremely successful object detection techniques, such as Deformable Part Models (DPM) [8]. More complicated descriptors [9] [10] [11] have been used for image retrieval with reasonable success. However, such methods are time consuming and more processor-intensive as compared to simple HOG matching. In recent years, handcrafted features have declined in popularity due to the success of deep neural networks in object recognition [12] [13], but such methods are not without their drawbacks. Deep neural networks require a lot of processor time and run better on specialized hardware. They also require far greater number of training images that are available in a small or medium-sized dataset to avoid overfitting. For these reasons, enhancing simple handcrafted features like HOG can be effective for solving small-scale retrieval problems more effectively than methods of greater complexity.

The use of HOG for shape matching is fraught with challenges as mentioned above in Section I. The difficulties of using HOG for shape-based image retrieval are particularly evident for content generated by users in the wild, but are also applicable to more controlled images such as Google StreetView [14] images due to seasonal differences in vegetation and lighting. In effect, every query image is an exemplar of its own class and a retrieval system must be trained to treat it that way. In [15], this idea is handled using a Support Vector Machine (SVM) [16]. Instead of an SVM, here we introduce the novel idea of enhancing the HOG features by the Enhanced Fisher Model (EFM) process [17] because it produces a low-dimensional representation, which is important from the computational aspect. Principal Component Analysis (PCA) has been widely used to perform dimensionality reduction for image indexing and retrieval [17]. The EFM feature extraction method has achieved good success rates for the task of image classification and retrieval [7]. In the proposed method, which is represented schematically in Figure 1, we show this method to be effective in isolating the query object from the background clutter as well.

The geo-localization problem has been addressed by many researchers with varying degrees of success since the end of the last decade. The works range in scale between [18] where the authors explore the distinguishable architectural features of cities to [19] [20] where the scale is global Earth. But our work brings the problem to the scale of identifying individual buildings on Google StreetView [14] and tries to solve it. This is most similar to the work of [21], but our method uses very few (< 10) boxes per image using our proposed EFM-HOG representation.

We design a coarse semantic segmentation algorithm and use it as a preprocessing step on the dataset that we built from Google StreetView images for testing our technique. Semantic segmentation of outdoor scene images into a small number of semantic categories has been addressed successfully by [22]. While they use color histograms in the RGB and HSV color space, texture, shape, perspective and SIFT features at the superpixel level to assign pixel-level semantic labels, this was not necessary in our case. HOG features are extracted from rectangular windows and it was sufficient to achieve enough coarse semantic segmentation to draw a rectangular

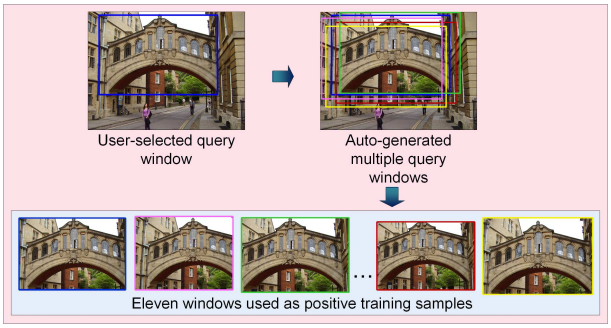

Figure 2: Auto-generation of offset windows to be used as positive training samples during querying. The window dimensions and offsets shown are only representative.

bounding box around the houses, and hence we used fewer features. Local Binary Patterns (LBP) [23] is known to provide good features for not only texture but also object and scene classification [24] [25] and so LBP was chosen as the primary feature to represent patches. We also use HSV color histogram and HOG feature vectors and concatenate them to LBP for this purpose. While deep neural networks have proven very successful for the semantic segmentation task [26] [27] [28] [29], they require a large number of training images with labeled ground truth. Even weakly supervised methods [30] require a large number of images labeled at the bounding-box level and do not work well with non-convex regions such as vegetation. Since we do not have a sufficient number of images with labeled ground truth data, and neural networks trained on other cities were found to perform poorly on our new Lake Forest StreetView dataset, we do not use deep neural networks for this work.

III. Proposed Method

The proposed method, as outlined in Figure 1, works by matching HOG features [5] from a selected window in a query

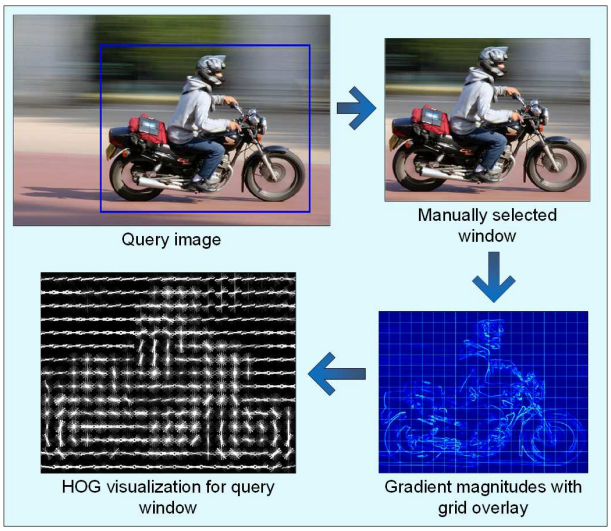

Figure 3: Formation of the HOG descriptor from a query image window.

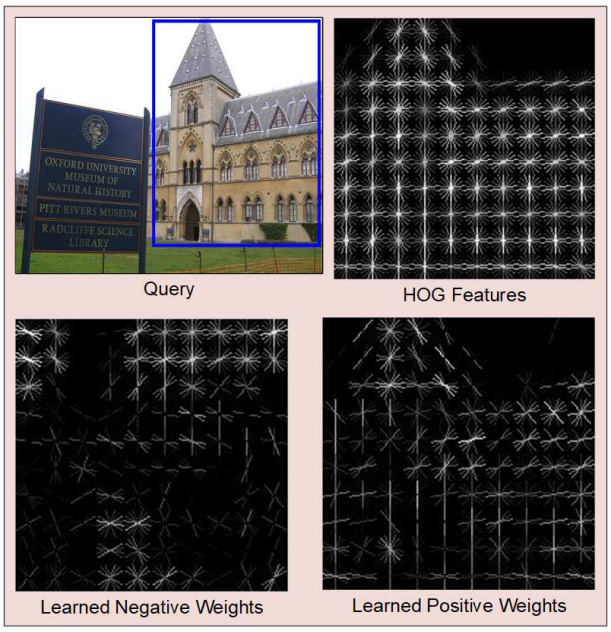

Figure 4: The positive and negative weights learned from the HOG features through the EFM discriminative feature extraction process.

image surrounding an object of interest with the HOG features extracted from the similarly shaped objects in other images of the dataset. The following few subsections explain in detail the various steps needed for the proposed feature extraction and retrieval process.

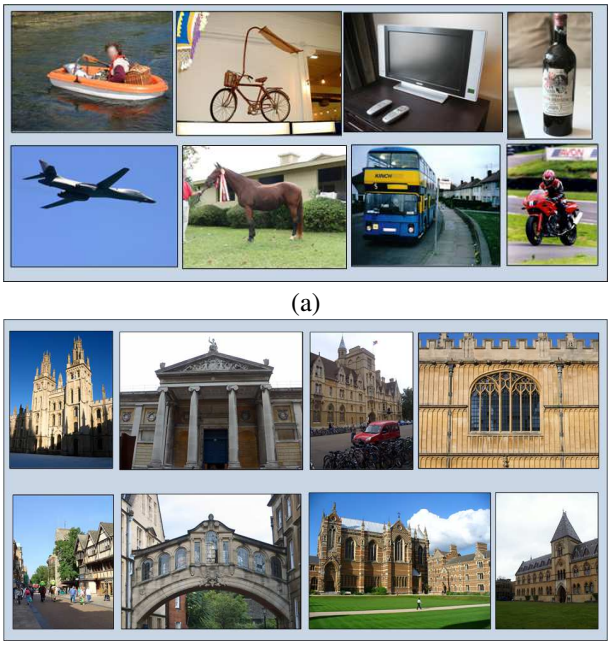

Figure 5: Some sample query images from (a) the PASCAL VOC 2012 dataset, and (b) the Oxford Buildings dataset.

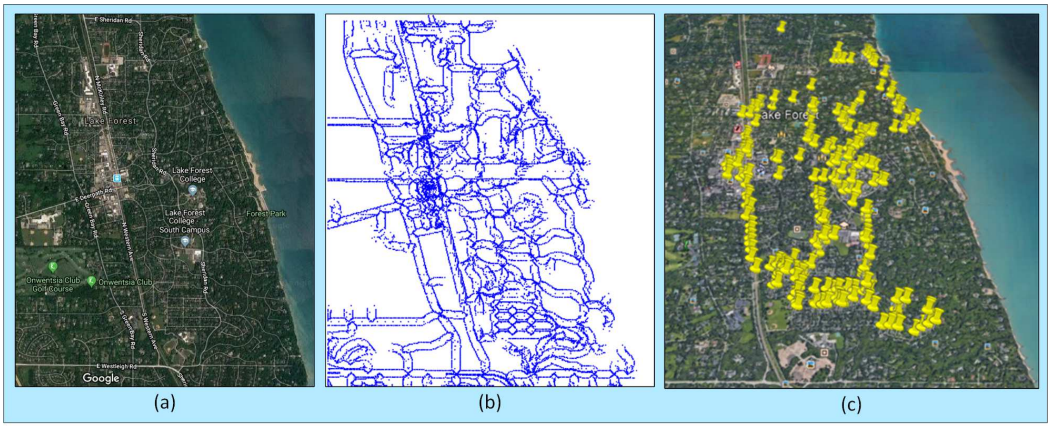

Figure 6: The locations of the images in our dataset. (a) shows the map of Lake Forest. (b) shows the distribution of the Google StreetView images collected. (c) shows the locations of our query images.

A. Window Generation

We start with generating objectness windows from each image. We use different methods on different datasets for this purpose. For the PASCAL VOC 2012 dataset [2] and the Oxford Buildings dataset [3], the method used by [31] is found to work well. This method designs an objectness measure and explicitly trains it to distinguish windows containing an object from background windows. It uses five objectness cues – namely, multi-scale saliency, color contrast, edge density, superpixels straddling, and location and size – and combines them in a Bayesian framework. We select the 25 highest-scoring windows from each image in our dataset and extract HOG features from these windows.

For the Lake Forest StreetView dataset introduced by us, the objectness method by [31] does not work well. For this dataset, we generate windows of interest by a combination of a patch-wise semantic segmentation and heuristics-based algorithm described in more detail in Section IV-C1. This method typically produces less than 10 windows per image. Whatever method we use for generating the windows, the window coordinates are pre-calculated and stored for each image file in a dataset.

While testing our system, the user generates a window on the query image manually roughly enclosing the object of interest. Then, we automatically select 10 slightly offset versions of this window. Eight of these are generated by moving the user-selected window to the right, left, up, down, up-right, up-left, down-right and down-left by 5%, respectively. Two windows are generated by expanding and contracting the user's selection by 5%, respectively. Features are now extracted from these 10 as well as the original window for further processing. This process is represented in Figure 2.

B. The HOG Descriptor

The idea of HOG rests on the observation that local features such as object appearance and shape can often be characterized well by the distribution of local intensity gradients in the image [5]. HOG features are derived from an image based on a series of normalized local histograms of image gradient orientations in a dense grid [5]. The final HOG descriptors are formed by concatenating the normalized histograms from all the blocks into a single vector.

Figure 3 demonstrates the formation of the HOG vector for a window selected from an image. We use the HOG implementation in [32] for both generating the descriptors and rendering the visualizations used in this paper.

C. Dimensionality Reduction

PCA, which is the optimal feature extraction method in the sense of the mean-square-error, derives the most expressive features for signal and image representation. Specifically, let $\mathcal{X} \in \mathbb{R}^N$ be a random vector whose covariance matrix is defined as follows [33]:

$$S = \mathcal{E}\{[\mathcal{X} - \mathcal{E}(\mathcal{X})][\mathcal{X} - \mathcal{E}(\mathcal{X})]^t\} \quad (1)$$

where $\mathcal{E}(\cdot)$ represents expectation and t the transpose operation. The covariance matrix S is factorized as follows [33]:

$$S = \Phi \Lambda \Phi^t \quad (2)$$

where $\Phi = [\phi_1 \phi_2 \cdots \phi_N]$ is an orthogonal eigenvector matrix and

$$\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$$

a diagonal eigenvalue matrix with diagonal elements in decreasing order. An important application of PCA is the extraction of the most expressive features of $\mathcal{X}$. Towards that end, we define a new vector $\mathcal{Y}$: $\mathcal{Y} = P^t \mathcal{X}$, where $P = [\phi_1 \phi_2 \ldots \phi_K]$, and $K < N$. The most expressive features of $\mathcal{X}$ thus define the new vector $\mathcal{Y} \in \mathbb{R}^K$, which consists of the most significant principal components.

D. EFM

The features obtained after dimensionality reduction by PCA as discussed in Section III-C are the most expressive features for representation. However, they are not the optimum features for classification. Fisher's Linear Discriminant (FLD), a popular method in pattern recognition, first applies PCA for dimensionality reduction and then discriminant analysis

for feature extraction. Discriminant analysis often optimizes a criterion based on the within-class and between-class scatter matrices S_w and S_b, which are defined as follows [33]:

$$S_w = \sum_{i=1}^{L} P(\omega_i)\mathcal{E}\{(\mathcal{Y} - M_i)(\mathcal{Y} - M_i)^t|\omega_i\} \quad (3)$$

$$S_b = \sum_{i=1}^{L} P(\omega_i)(M_i - M)(M_i - M)^t \quad (4)$$

where $P(\omega_i)$ is *a priori* probability, ω_i represents the classes, and M_i and M are the means of the classes and the grand mean, respectively. One discriminant analysis criterion is J_1: $J_1 = tr(S_w^{-1}S_b)$, and J_1 is maximized when Ψ contains the eigenvectors of the matrix $S_w^{-1}S_b$ [33]:

$$S_w^{-1}S_b\Psi = \Psi\Delta \quad (5)$$

where Ψ, Δ are the eigenvector and eigenvalue matrices of $S_w^{-1}S_b$, respectively. The discriminating features are defined by projecting the pattern vector $\mathcal{Y}$ onto the eigenvectors of Ψ:

$$\mathcal{Z} = \Psi^t\mathcal{Y} \quad (6)$$

$\mathcal{Z}$ thus contains the discriminating features for image classification.

The FLD method, however, often leads to overfitting when implemented in an inappropriate PCA space. To improve the generalization performance of the FLD method, a proper balance between two criteria should be maintained: the energy criterion for adequate image representation and the magnitude criterion for eliminating the small-valued trailing eigenvalues of the within-class scatter matrix. The EFM improves the generalization capability of the FLD method by decomposing the FLD procedure into a simultaneous diagonalization of the within-class and between-class scatter matrices [17]. The simultaneous diagonalization demonstrates that during whitening, the eigenvalues of the within-class scatter matrix appear in the denominator. As shown by [17], the small eigenvalues tend to encode noise, and they cause the whitening step to fit for misleading variations, leading to poor generalization performance. To enhance performance, the EFM method preserves a proper balance between the need that the selected eigenvalues account for most of the spectral energy of the raw data (for representational adequacy), and the requirement that the eigenvalues of the within-class scatter matrix (in the reduced PCA space) are not too small (for better generalization performance). For this work, the number of eigenvalues was empirically chosen.

TABLE I: The number of images in each class of the PASCAL VOC 2012 dataset

Object Category	Number of Images
aeroplane	670
bicycle	552
bird	765
boat	508
bottle	706
bus	421
car	1161
cat	1080
chair	1119
cow	303
dining table	538
dog	1286
horse	482
motorbike	526
person	4087
potted plant	527
sheep	325
sofa	507
train	544
TV/monitor	575

TABLE II: The number of images containing each landmark in the Oxford Buildings dataset

Landmark	Good	OK	Junk
All Souls Oxford	24	54	33
Ashmolean Oxford	12	13	6
Balliol Oxford	5	7	6
Bodleian Oxford	13	11	6
Christ Church Oxford	51	27	55
Cornmarket Oxford	5	4	4
Hertford Oxford	35	19	7
Keble Oxford	6	1	4
Magdalen Oxford	13	41	49
Pitt Rivers Oxford	3	3	2
Radcliffe Camera Oxford	105	116	127

E. Training

The EFM feature extraction method uses positive and negative training samples to find the most discriminative features. In our setting, there is only one query image to be used as a positive sample. This is similar to the Exemplar-SVM training scenario used by [15], but to make the training more robust to selection error by the user and to prevent overfitting, we use 11 windows instead of just the one selected by the user as described in Section III-A.

We rank all region of interest (ROI) windows from all images in the dataset in terms of Euclidean distance in the HOG space from the original query window. For the negative training samples, we use 110 windows that are ranked low, i.e., are very distant in the HOG space. Experimentally, we found that the windows that are ranked last (i.e., farthest from the query) in the dataset are not very good candidates for negative training samples, since they are often outlier windows that contain large blank areas like the sky. Instead, windows that have a rank of $1,000$ to $5,000$ when sorted by increasing HOG distance were seen to perform well. We also tried training the system with different numbers of negative samples and found a number close to 100 performs the best. These windows are mostly background regions like ground and vegetation. The positive and negative weights for the HOG features learned by this method can be seen in Figure 4.

For an n-class problem, the EFM process for discriminatory feature extraction reduces the dimensionality of any vector to $n - 1$. Since our problem is a two-class problem, EFM produces one feature per window. We compute the score of each window by finding the absolute value of the difference between the window EFM feature and the average positive training set EFM feature. Ranking the images by their best-scoring windows gives us the retrieval set.

IV. Shape Retrieval Tasks and Datasets

To prove the effectiveness of our proposed EFM-HOG descriptor and the associated distance measure, we apply it

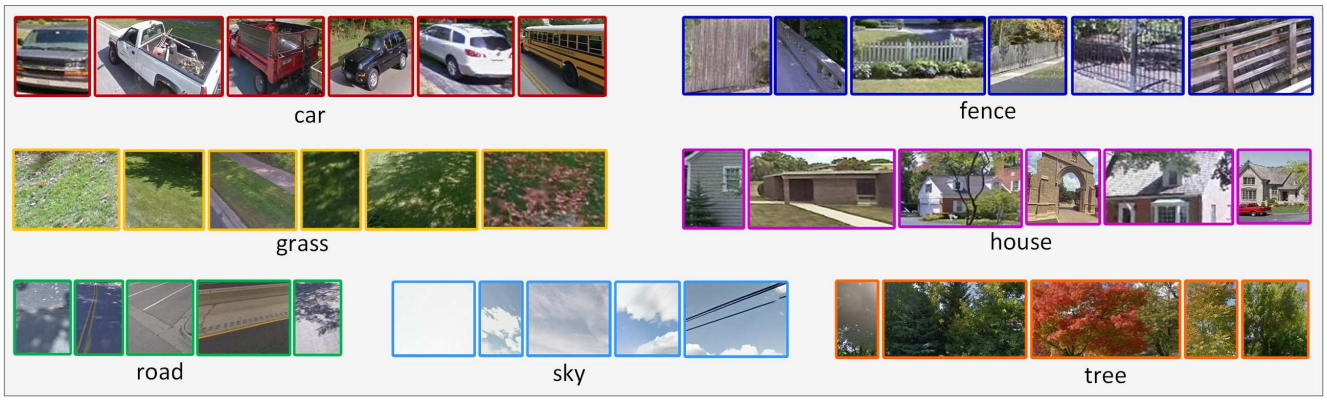

Figure 7: Manually selected training patches used to train the seven SVM classifiers for coarse semantic segmentation.

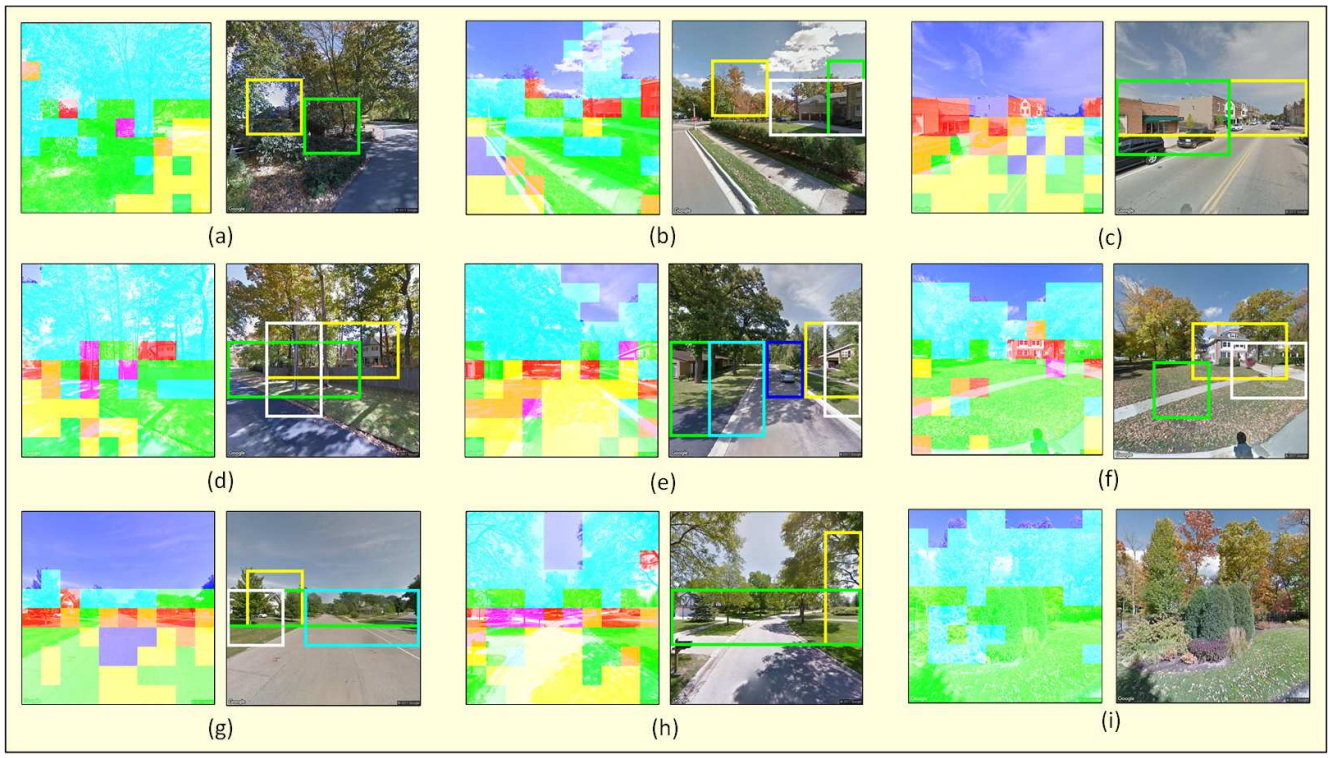

Figure 8: Some examples of ROI selection from our reference dataset. The left-side images in (a) through (i) show different semantic categories by using different colors. The red patches indicate the house category and magenta indicates fence. The right-side images show the output of our ROI-window generation algorithm. Colors of the rectangles in the right-side images have no significance. Note that in (i), no buildings are found, and so no windows are generated.

to three distinct problems. For each of these tasks, we use a different dataset with properties suitable for the problem being addressed. In this section, we will give a brief description of the different problems addressed and datasets used for our experiments, and then we will discuss the performance of our novel EFM-HOG matching algorithm on these datasets in the next section.

A. *Object Search and Retrieval*

The first problem that we address is that of object search and retrieval. In this problem, the user selects a bounding box around an object in a query image and we attempt to retrieve similar objects from the dataset. For this task, we use PASCAL VOC 2012 dataset [2]. We only use the training/validation data from this dataset to test our retrieval algorithm. This data consists of $11,540$ images from 20 classes (many images have multiple classes present). The classes in this image dataset are aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train and TV/monitor. The classes and the number of images in them are shown in Table I. Figure 5(a) shows some images from this dataset. We create five randomly selected

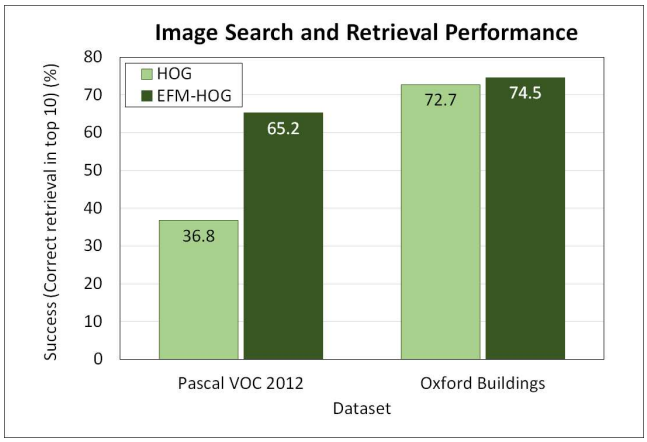

Figure 9: Mean retrieval accuracy (measured by the presence of a relevant image in the top 10 retrieved images).

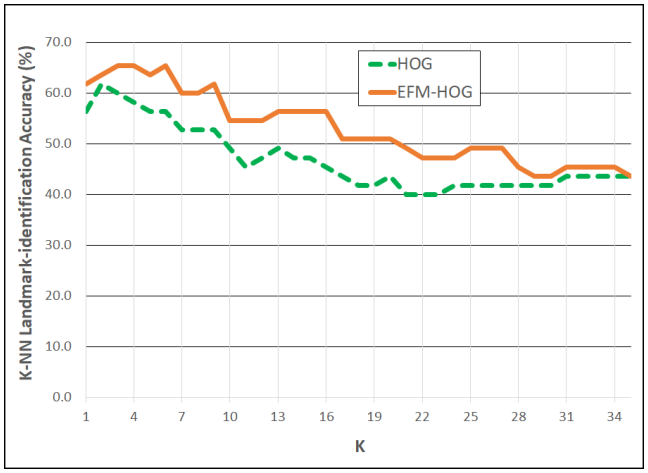

Figure 11: The mean landmark-recognition performance on the Oxford Buildings dataset by using the k-nearest neighbors method with varying k.

100-image test sets from the dataset and perform a five-fold cross-validation. A successful retrieval experiment is one where the program retrieves at least one relevant image (an image containing the query object) within the top 10 results. The performance of our descriptor on this dataset is discussed in Section V.

B. *Landmark Recognition in the Wild*

The second problem that we address is that of landmark recognition in the wild. This is more challenging than the object search and retrieval problem because the images here are mostly outdoor images, and buildings are not always as easily distinguishable from their surroundings as object images are. The dataset that we use for this problem is the Oxford Buildings dataset [3], which consists of $5,062$ images of 11 different Oxford landmarks and distractors collected from Flickr [34]. 55 images from this dataset were used as queries for testing our retrieval system. Flickr images are completely user-generated, which means there is a great variation in camera type, camera angle, scale and lighting conditions. This makes this dataset very difficult for image retrieval in general and landmark-recognition in particular. Figure 5(b) shows some of our query images from this dataset. For each query, the images that contain the query landmark are further classified into *good*, *OK* and *junk* categories, with progressively poorer views of the query landmark. Table II shows the landmark-wise distribution of *good*, *OK* and *junk* images in this dataset.

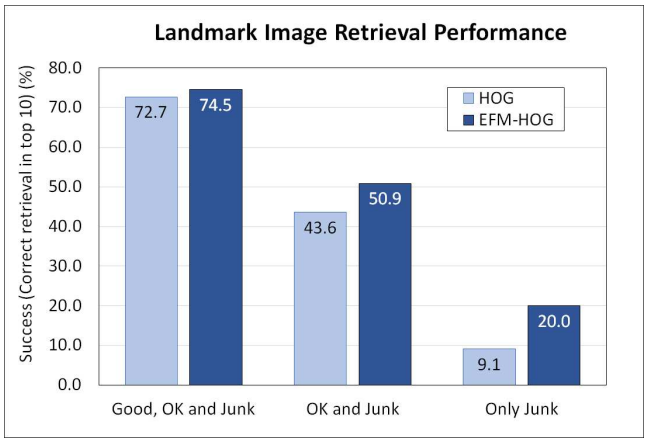

Figure 10: Landmark image-retrieval accuracy (measured by the presence of a relevant image in the top 10 retrieved images) on the Oxford Buildings dataset. The three sets of values show the success rate of HOG and EFM-HOG while varying the quality of available matches in the reference set.

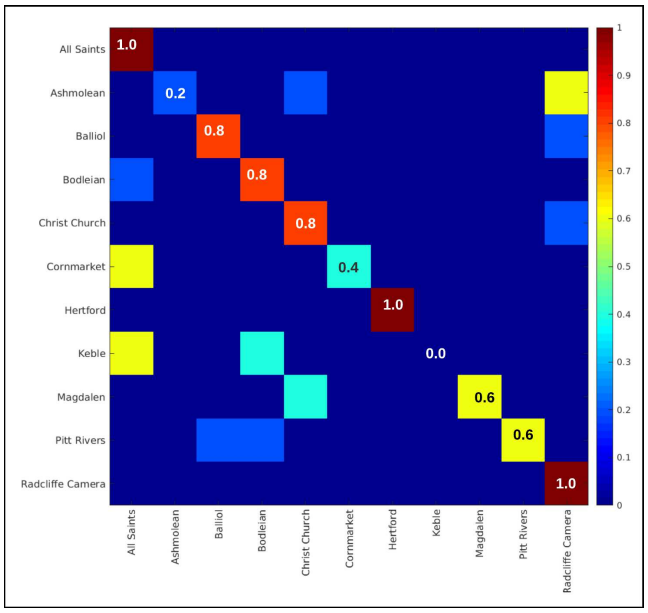

Figure 12: The confusion matrix for the landmark-recognition performance of the EFM-HOG descriptor on the Oxford Buildings dataset by using the k-nearest neighbors method with k=3.

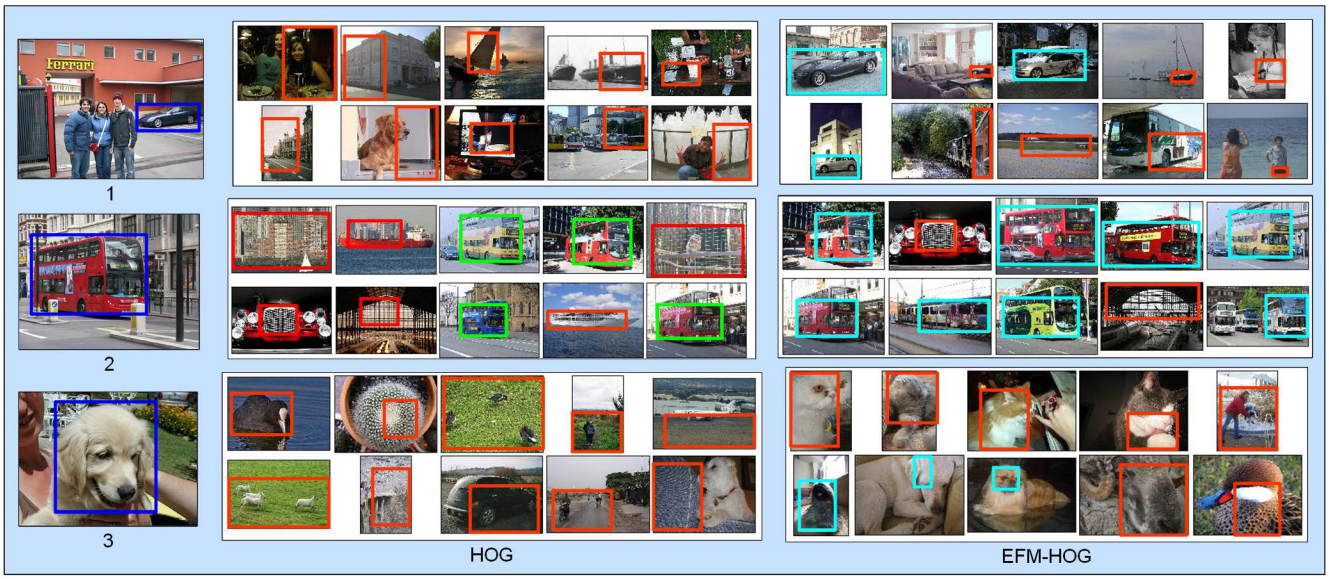

Figure 13: Comparison of image retrieval results for HOG and the proposed EFM-HOG on the PASCAL VOC 2012 dataset.

C. Image-based Geo-Localization

Finally, to test our proposed algorithm for image-based geo-localization of buildings within an entire city using StreetView images, we built a new image dataset [4], which we call the Lake Forest StreetView dataset, from the city of Lake Forest, Illinois. We collected two sets of images for this purpose: one for query images, and one for reference images. We acquired our own images for the query dataset by walking around the city and taking photos of buildings with smartphone cameras. This dataset has 308 images. For the reference dataset, we downloaded Google StreetView [14] images from around the city. We downloaded eight overlapping StreetView images from points eight feet apart along every road in Lake Forest. This process downloaded $126,000$ images. From our 308 query images, we selected 128 images spread over the whole city of Lake Forest that contained buildings that were also visible in at least one of the reference images. To do this, we wrote a program that uses the GPS tags on each query image to retrieve the geographically nearest 100 images from the reference dataset. We then visually inspected this retrieved set to determine if the query image building was visible in any of them. Finally, we combined these retrieved sets together, eliminated duplicates, and added a few thousand random distractor StreetView images to bring the total up to $10,000$ images. This was our final reference set for the experiments. Figure 6(a) shows the Google Maps view of Lake Forest. Figure 6(b) shows the distribution of our reference set, which is composed of downloaded Google StreetView images. Each blue point in this image represents the location of a Google StreetView photo. It can be seen that our image dataset follows the streets and there are large areas without any images in between, which are private estates and parks. Finally, Figure 6(c) shows the distribution of our query images using yellow markers. These markers were generated directly using the GPS tags of the query images, which were taken using smartphone cameras.

We ran retrieval experiments on this set using each of the 128 query images. We manually drew rectangles around buildings in each of the query images, which were then used to extract the EFM-HOG features for matching. The process of selecting multiple windows that are slightly offset from the original reduces the impact of slight variations between manually drawn rectangles in two experiments, but still the manually drawn rectangle boundaries were saved to preserve repeatability between experiments. Degree of success or failure of a retrieval was measured by the mean geographical distance of retrieved images from the query, and also by the presence or absence of the query building in the retrieval set.

1) ROI Selection for Geo-localization Problem: HOG matching starts with selecting a bounding box around the ROI, which in this case would be the buildings. The task of selecting the buildings in Lake Forest, however, is non-trivial due to a characteristic of the city itself. The city of Lake Forest has a very large number of trees and most of the houses are far from the road in the middle of large estates. The Google StreetView images are shot with a wide-angle camera mounted on a moving car. The combination of a wide-angle lens and the large distance from the road causes the houses to appear very small in the images, and the vegetation or parts of the road closer to the camera appear much larger. In majority of the reference images, the buildings occupy only a small portion of the image, the rest being filled with vegetation, sky or portions of the road. Hence, selecting an ROI containing the building becomes an important preprocessing step before features can be extracted.

The object detection program used on the other two datasets did not work well on this dataset, and we needed some coarse form of semantic segmentation to separate the houses from the vegetation, road and other objects. We did not use a deep neural network for this purpose because of two reasons. First, we did not need pixel-level separation of categories since HOG features are extracted from rectangular windows anyway. Second, we did not have labeled segmentation ground truth training images and networks trained on images from other

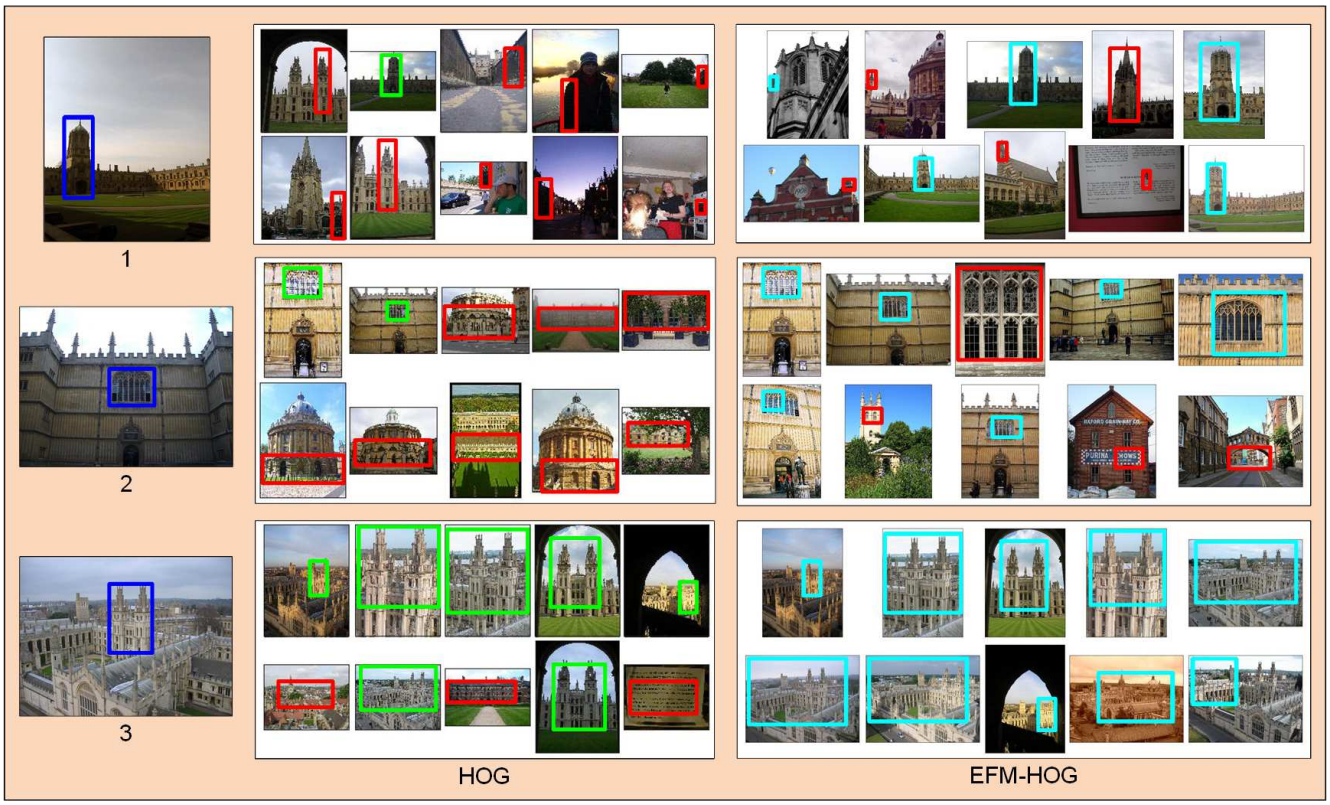

Figure 14: Comparison of landmark image retrieval results for HOG and the proposed EFM-HOG on the Oxford Buildings dataset.

cities did not generalize well to Lake Forest. So, we built our own semantic segmentation technique for this step.

On visual inspection of the images, we decided there were seven major semantic classes, namely sky, grass, tree, road, house, fence and vehicles. We manually selected rectangular patches from each of these classes and extracted three sets of features from each patch. These features are color histogram in the HSV color space, HOG and LBP. These three sets of features are concatenated to get our feature vector to train the classifiers for coarse semantic segmentation. For this task, we trained an SVM [16] classifier for each class.

2) The Linear SVM Classifier: The SVM is a particular realization of statistical learning theory. The approach described by SVM, known as structural risk minimization, minimizes the risk functional in terms of both the empirical risk and the confidence interval [16]. The SVM implementation used for our experiments is the one that is distributed with the VlFeat package [32]. We use the one-vs-all method to train an SVM for each semantic category. The parameters of the SVM are tuned empirically using only the training data, and the parameters that yield the best average precision on the training data are used for classification of the test data.

We created the training data for our SVMs in the form of rectangular windows selected manually from the reference images. 100 training patches were used per class. Some of these patches are shown in Figure 7. We divide each reference image into 100 uniformly sized patches over a 10×10 regular grid and pass each patch through all seven classifiers to assign one final label to each patch. Finally, we draw minimal bounding boxes around the house and fence category patches (if any) with some padding around them, and extract HOG features from them. This process is shown in the different parts of Figure 8. The bounding boxes in Figure 8(a) look like they are enclosing vegetation, but there are underlying house and fence outlines visible here through the trees. Bounding boxes are not perfect, and sometimes they either enclose objects other than houses, like the blue box in Figure 8(e), or the green box in Figure 8(f), or are too large, like the green box in Figure 8(h). False positives, however, are less of a problem to our technique than false negatives, and extra bounding boxes are better than missing houses. Also note that in Figure 8(i), there are no buildings, and no bounding boxes are generated, which is a strong indication that our segmentation algorithm is successful in cutting down on the number of undesirable windows in a large number of cases. If we had used the objectness code that we ran on the other two datasets, we would have got 25 windows from this image as well. The different colors of the bounding boxes in the right-side images in Figure 8 have no special significance. The various colors have been used to differentiate between the rectangles.

V. Experiments and Results

In this section, we describe in detail the experiments that we performed on our three datasets and the results that we obtained in each of the three tasks that we attempted. For each of the tasks, we used a different dataset and compared the results of our EFM-HOG descriptor with that obtained

Figure 15: Comparison of image-based geo-localization results for HOG and the proposed EFM-HOG on the Lake Forest StreetView dataset.

by conventional HOG features. We also provide samples of query images along with top matching images retrieved by both algorithms from all three datasets for a qualitative comparison of the results.

A. *The Object Search and Retrieval Task*

The proposed image representation is tested on three different tasks, the first of which is object search and retrieval. Here, an image is used as a query to retrieve similar scenes from the dataset. For this, the user selects a rectangular ROI from the query image, and HOG features from this rectangular window are matched with the 25 highest scoring objectness windows from each image in the database, both in the raw HOG space and in the EFM-HOG space after the proposed training and feature extraction procedure. The closest matches based on Euclidean distance are retrieved in order of their distance from the query window. Finding an instance of the query class object in the top 10 retrieved images is considered a success. Figure 9 compares the retrieval success rates of the HOG descriptor and the proposed EFM-HOG representation on the PASCAL VOC 2012 dataset. For this dataset, the retrieval experiment is performed on five random splits and the average success rate is found to be 65.2% for EFM-HOG as compared to 36.8% for HOG. We also find that the conventional HOG performs quite well for clearly segmented objects, such as airplanes in the sky, but the EFM-HOG performs much better for images of objects with a cluttered background.

We also experimented on the Oxford Buildings dataset from the retrieval point of view. Figure 9 also compares the retrieval success rates of the HOG descriptor and the proposed EFM-HOG representation on this dataset alongside the PASCAL VOC 2012 performance. Specifically, in 41 cases out of 55 queries in the Oxford buildings dataset (74.5% cases), the query landmark is retrieved within top 10 images by the proposed method, as opposed to 40 by HOG (72.7% cases). This is actually a very small difference, but this can be explained by the nature of this dataset. For all landmark query images in this dataset, there are at least some *Good* images in the dataset that show clear views of the landmarks with no occlusions. HOG is actually pretty effective at retrieving these images. To better demonstrate the effectiveness of the proposed method, we repeat this experiment with just the *OK* and *Junk* images, and then just the *Junk* images for each query. In these experiments, we find that the HOG method retrieves a relevant image in the top 10 much less frequently than the EFM-HOG method. With just the *Junk* files, EFM-HOG performs more than twice as well as HOG. These results are shown in Figure 10.

B. *The Landmark-Recognition Task*

The second experiment that we performed with the new EFM-HOG descriptor on the Oxford Buildings dataset was a landmark-recognition task where the system tries to label each query image with its correct landmark label. Some images

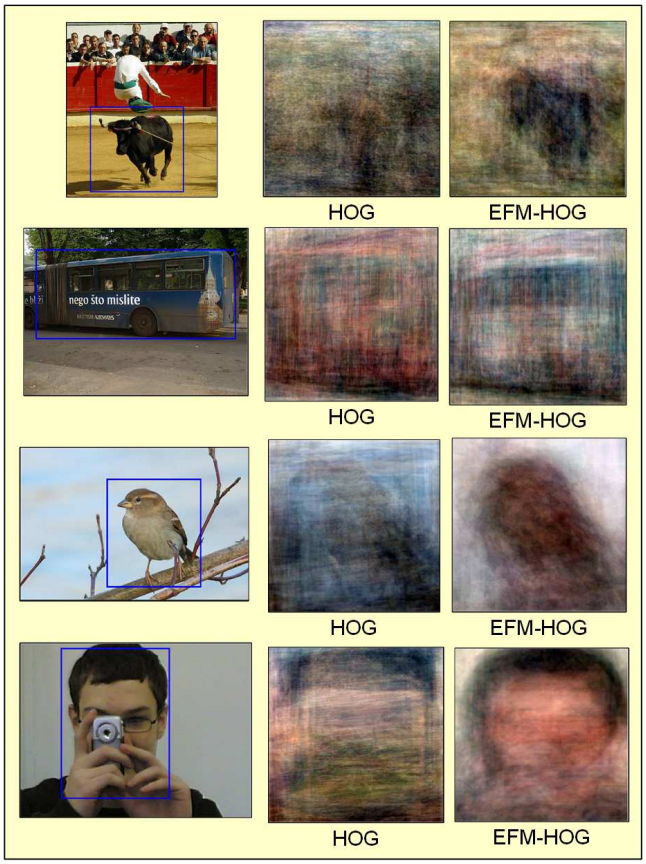

Figure 16: The means of the top 100 retrieved windows for HOG and EFM-HOG for 4 query images from the PASCAL VOC 2012 dataset.

in this dataset belong to one of the 11 landmarks listed in Table II, the others belong to none of the classes and are used as distractors. We did this task by retrieving relevant images in a manner similar to the retrieval task, and then performing the k-nearest neighbors (k-NN) classification on the top k results. The same experiments are repeated for the conventional HOG descriptor as well. As can be seen from Figure 11, the proposed EFM-HOG outperforms HOG all values of k between 1 and 35. The highest EFM-HOG landmark-recognition performance of 65.5% is achieved at k=3. A further breakdown of the landmark-recognition performance of the EFM-HOG descriptor is seen in Figure 12. Here, the rows represent real landmark labels of the queries and the columns represent predicted labels. The results are averaged over the 5 query images for each landmark and the k-NN classifier has been used with k=3.

Some HOG and EFM-HOG retrieval results on the PASCAL VOC 2012 dataset are shown in Figure 13. In this figure, the query images are shown on the left of each row with blue bounding boxes followed by the two retrieval sets obtained by using HOG and EFM-HOG. A red bounding box in a retrieved image indicates that the retrieved image is not from the same class from the query image (shown on the left). Correct matches for HOG are shown with green bounding boxes and correct matches for EFM-HOG are shown with cyan bounding boxes.

Some comparative retrieval results between HOG and EFM-HOG on the Oxford Buildings dataset are shown in Figure 14. In this figure too, the query images are shown on the left of each row with blue bounding boxes. The retrieval set in the middle of each row is obtained by using HOG and the retrieval set on the right is obtained by using EFM-HOG. A red bounding box in a retrieved image indicates that the retrieved image has a different landmark building label from the query image (shown on the left). Correct label matches for HOG are shown with green bounding boxes and correct label matches for EFM-HOG are shown with cyan bounding boxes.

C. The Geo-localization Task

We ran two sets of experiments on our Lake Forest StreetView dataset for this task. The first set does the retrieval with traditional HOG and the second set uses the proposed EFM-HOG matching. The improvement in retrieved result sets achieved by the proposed EFM-HOG technique can be seen by comparing the results shown in Figure 15. In this figure, the retrieved images have their geographical distance from the query written above them. Green text signifies a retrieved image closer than 0.1 miles, and in all the examples here, an exact match. As can be seen in all the examples, HOG fails to find even a single match for the building in the query image in the top 10 retrieval results while EFM-HOG finds one in all three.

Our EFM-HOG match program retrieved (within the top 20 results) at least one image that was closer than 100 yards (0.0568 miles) of our query in 40 out of the 128 queries that we used. In 17 of these images the exact building was found and matched. Three such query images and the top 10 retrieved images along with their geographic distances are shown in Figure 15. In the result images, a geographic distance written in green indicates an actual match. HOG is unable to retrieve a match in the top ten results in any of the three queries shown while EFM-HOG fetches one result among the top ten in all three.

D. Qualitative Analysis of Retrieved Image Windows

We also manually inspected our retrieved image windows and ran some experiments to do a qualitative analysis of the results. Figure 16 shows an interesting aspect of our retrieval technique. Here, we show the image means of the first 100 windows retrieved by both HOG and EFM-HOG on the PASCAL VOC 2012 dataset. The figure shows that the EFM-HOG means contain clearer shapes, which indicates that the EFM-HOG retrieves more similar shapes than HOG, even when the results are irrelevant to the query.

A few successfully geo-localized buildings from our retrieval experiments on the Lake Forest StreetView dataset are shown in Figure 17. In each of the image pairs shown in this figure, the left image with a red bounding box shows the query taken with a smartphone camera, and the right one with a cyan bounding box shows a retrieved image from the Lake Forest StreetView dataset. In one of these images, the match is successful even with only a small section of the fence visible in the query, which shows the technique is quite robust. The rectangles around the buildings in the retrieved images themselves were generated by our coarse semantic

Figure 17: Successfully geo-located query images along with the retrieved Google StreetView images that are exact matches for the query.

segmentation algorithm, which is also a qualitative measure of the success of this algorithm.

VI. Conclusion and Future Work

We have presented in this paper a new image descriptor based on HOG and discriminant analysis that uses a novel approach to fetch scenes with similar shaped objects. We have conducted experiments using over $5,000$ images from the Oxford Buildings dataset and over $11,500$ images from the PASCAL VOC 2012 dataset and concluded the following: (i) HOG features are not always sufficiently discriminative to perform meaningful retrieval, (ii) the discriminative nature of HOG features can be improved with the EFM for feature extraction and dimensionality reduction, and (iii) HOG features perform well for clearly isolated objects with little background clutter, but the EFM-HOG performs better for real-world images with cluttered backgrounds.

We furthermore demonstrated the effectiveness of our proposed EFM-HOG descriptor for geo-localization on a $10,000$-image Lake Forest StreetView dataset that we built from scratch. We also developed a coarse semantic segmentation strategy to automatically isolate buildings and draw bounding boxes around them as a preprocessing step before the HOG feature extraction. Finally, we compare the proposed EFM-HOG representation and the traditional HOG representation to demonstrate that our method is superior for retrieval. We intend to use this method with other image retrieval tasks in the future, so that a more thorough understanding of its strengths and weaknesses can be achieved.

It is evident that the successful geo-localization in the Lake Forest StreetView dataset depends heavily on the quality of the bounding boxes generated by our coarse semantic segmentation algorithm, and improving that algorithm will significantly improve the results. In future, we plan to develop a more robust strategy for semantic segmentation. Superpixel-based and deep neural network-based semantic segmentation may also be used if we can get a sufficient number of labeled images. We also plan to extend our dataset to cover other cities.

Acknowledgment

The authors would like to thank Professor Jana Košecká at the Department of Computer Science, George Mason University, Fairfax, Virginia for some valuable input on the EFM-HOG method and the experiments conducted.

The authors would also like to thank the Richter Scholars Program at Lake Forest College for partially supporting the research presented in this paper.

References

[1] S. Banerji and A. Sinha, "EFM-HOG: Improving Image Retrieval in the Wild," in Proceedings of the Fourth International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2019), June 2019, pp. 6–11.

[2] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, 2010, pp. 303–338.

[3] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[4] R. R. Zunker, S. Banerji, and A. Sinha, "House Hunting: Image-based Geo-Localization of Buildings within a City," in Proceedings of the Fifth International Conference on Computing and Data Engineering, 2019, pp. 100–104.

[5] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[6] S. Banerji, A. Sinha, and C. Liu, "Scene Image Classification: Some Novel Descriptors," in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2012, pp. 2294–2299.

[7] A. Sinha, S. Banerji, and C. Liu, "Novel Color Gabor-LBP-PHOG (GLP) Descriptors for Object and Scene Image Classification," in Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, 2012, pp. 58:1–58:8.

[8] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, 2010, pp. 1627–1645.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, November 2004, pp. 91–110.

[10] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in Proceedings of the European Conference on Computer Vision, 2006, pp. 404–417.

[11] K. E. A. Van De Sande, C. G. M. Snoek, and A. W. M. Smeulders, "Fisher and VLAD with FLAIR," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2014, pp. 2377–2384.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of the Twenty-sixth Conference on Neural Information Processing Systems, 2012, pp. 1106–1114.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of the Third International Conference on Learning Representations, 2015.

[14] "Google StreetView," https://www.google.com/maps, accessed on Mon, December 7, 2020.

[15] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of Exemplar-SVMs for Object Detection and Beyond," in Proceedings of the International Conference on Computer Vision, 2011, pp. 89–96.

[16] V. Vapnik, The Nature of Statistical Learning Theory. Springer-Verlag, 1995.

[17] C. Liu and H. Wechsler, "Robust Coding Schemes for Indexing and Retrieval from Large Face Databases," IEEE Transactions on Image Processing, vol. 9, no. 1, 2000, pp. 132–137.

[18] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes Paris look like Paris?" Communications of the ACM, vol. 58, no. 12, November 2015, pp. 103–110.

[19] J. Hays and A. A. Efros, "IM2GPS: estimating geographic information from a single image," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[20] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, "Image sequence geolocation with human travel priors," in Proceedings of the Twelfth IEEE International Conference on Computer Vision, September 2009, pp. 253–260.

[21] R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Locality in generic instance search from one example," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2014, pp. 2099–2106.

[22] G. Singh and J. Košecká, "Introspective Semantic Segmentation," in Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 714–720.

[23] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," Pattern Recognition, vol. 29, no. 1, 1996, pp. 51–59.

[24] S. Banerji, A. Sinha, and C. Liu, "New image descriptors based on color, texture, shape, and wavelets for object and scene image classification," Neurocomputing, vol. 117, no. 0, 2013, pp. 173–185.

[25] ——, "A New Bag of Words LBP (BoWL) Descriptor for Scene Image Classification," in Proceedings of The Fifteenth International Conference on Computer Analysis of Images and Patterns, 2013, pp. 490–497.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015, pp. 3431–3440.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, 2017, pp. 2481–2495.

[28] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, 2018, pp. 834–848.

[29] W. Wang, Y. Fu, Z. Pan, X. Li, and Y. Zhuang, "Real-time driving scene semantic segmentation," IEEE Access, vol. 8, 2020, pp. 36 776–36 788.

[30] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1742–1750.

[31] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, November 2012, pp. 2189–2202.

[32] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008, accessed on Mon, December 7, 2020.

[33] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed. Academic Press, 1990.

[34] "Flickr," http://www.flickr.com, 2004, accessed on Mon, December 7, 2020.