House Hunting: Image-based Geo-Localization of Buildings Within a City

Ryan R. Zunker
Lake Forest College
555 North Sheridan Road
Lake Forest, Illinois 60045
zunkerrr@lakeforest.edu

Atreyee Sinha
Edgewood College
1000 Edgewood College Drive
Madison, Wisconsin 53711
asinha@edgewood.edu

Sugata Banerji
Lake Forest College
555 North Sheridan Road
Lake Forest, Illinois 60045
banerji@lakeforest.edu

ABSTRACT

Image-based geo-localization is an emerging field of computer vision that has drawn much attention over the past decade. The problem, however, is not trivial and presents a number of significant difficulties. In this paper we present a novel technique of addressing this problem by combining several Machine Learning techniques and demonstrate its effectiveness in the case of a particularly challenging city. In particular, we train a classifier to isolate buildings from images, then propose a novel EFM-HOG representation to match shape of buildings between images, and finally combine all of this to demonstrate geo-localization and retrieval results on a dataset that we created for this purpose.

KEYWORDS

Computer Vision, CBIR, Semantic Segmentation, Geo-Localization, Shape Matching, Enhanced Fisher Model, EFM-HOG

1 INTRODUCTION AND BACKGROUND

Today, nearly everyone carries a high-resolution camera on their person at all times. Identifying buildings in photos taken by these cameras can be useful for solving problems related to several areas such as tourism and law enforcement.

The image-based 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 [1] where the authors explore the distinguishable architectural features of cities to [2] and [3] where the scale is global Earth. But our work brings the problem to the scale of identifying individual buildings on Google Street View [4] and tries to solve it. This is most similar to the work of [5], but our method uses very few (< 10) boxes per image and uses a novel EFM-HOG representation.

The Histograms of Oriented Gradients (HOG) descriptor [6] is very popular among researchers for shape-based image matching. However, an approach like [7] seems to be more applicable for our situation since the angle the query image was photographed from affects its shape. While we were inspired by the idea of using each exemplar as a positive training set from [7], we did not use Support Vector Machines (SVM) for shape matching. Instead, we chose to introduce the novel idea of enhancing the shape features by the Enhanced Fisher Model (EFM) process [8] because it produces a low-dimensional representation which is important from the computational aspect. The EFM feature extraction method has achieved good success rates for the task of image classification and retrieval [9].

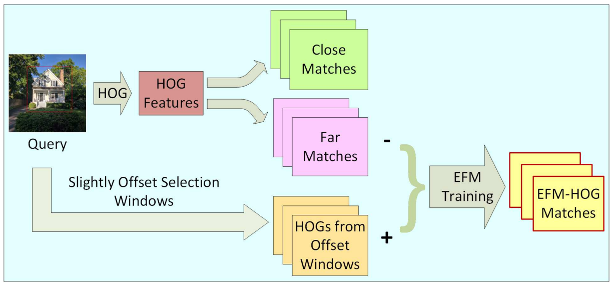

Figure 1: The process of generating the retrieval set using the proposed EFM-HOG match technique.

Semantic segmentation of outdoor scene images into a small number of semantic categories has been addressed successfully by [10]. 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 bounding box around the houses, and hence we used fewer features. Local Binary Patterns (LBP) [11] is known to provide good features for not only texture but also object and scene classification [12][13] and so LBP was chosen as the texture feature. We do not use deep neural networks for this work due to the lack of a sufficient amount of labeled ground truth data.

2 PROPOSED METHOD

The proposed method, as outlined in Figure 1, works by matching HOG features [6] of the buildings in the query image with the HOG features extracted from the buildings in Google Street View images. This is tested on images taken in the city of Lake Forest, Illinois. The task of HOG matching, however, is nontrivial due to several factors. First, the query images shot with smartphone cameras have different camera parameters, angles and lighting conditions from the Google Street View reference images. Second, in majority of the reference images, the buildings occupy only a small portion of the image, the rest being filled with vegetation. Hence it is important to select a region of interest (ROI) containing the building before features can be extracted. Finally, there are other differences

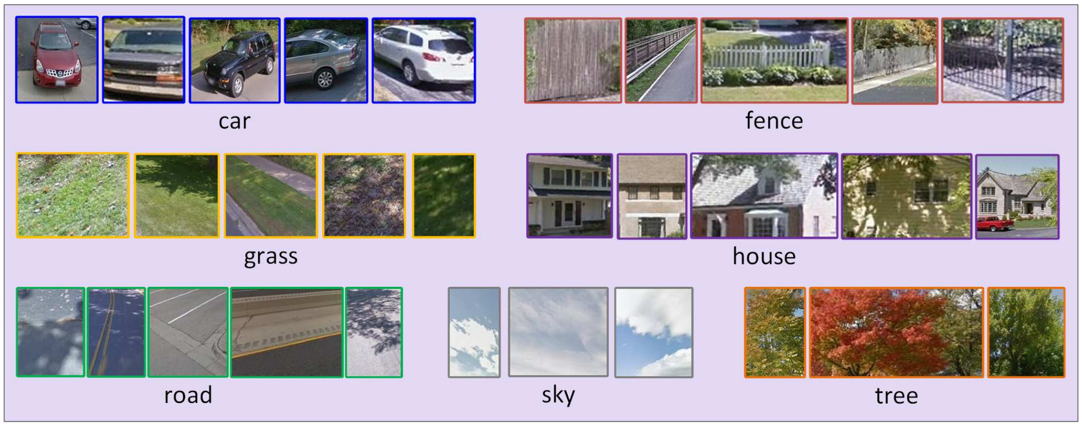

Figure 2: Manually selected training patches for the 7 SVM classifiers for coarse semantic segmentation.

between the query and the reference images due to changing seasons, passage of time and new constructions. These factors combine to make this problem an extremely challenging one regarding the city of Lake Forest, IL, USA. The following few sections explain the steps used to address these challenges.

2.1 Region of Interest (ROI) Selection

The city of Lake Forest has a large number of trees and most of the houses are far from the road in the middle of large estates. This makes the houses occupy a small area in the Google Street View images which are shot from a moving car. We needed some primitive form of semantic segmentation to separate the houses from the vegetation, road and other objects. 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, Histograms of Oriented Gradients (HOG) [6] and Local Binary Patterns (LBP) [11]. 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 a support vector machine (SVM) [14] classifier for each class.

2.1.1 The Linear SVM Classifier. The support vector machine (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 [14]. The SVM implementation used for our experiments is the one that is distributed with the VlFeat package [15]. We use the one-vs-all method to train an SVM for each semantic category. The parameters of the support vector machine 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 2. We divide each reference image into 100 uniformly sized patches over a 10×10 regular grid and pass each patch through all 7 classifiers to assign one final label to each patch. Finally, we draw minimal bounding boxes around the house and fence category patches (if any) and extract HOG features from them. This process is shown in Figure 3.

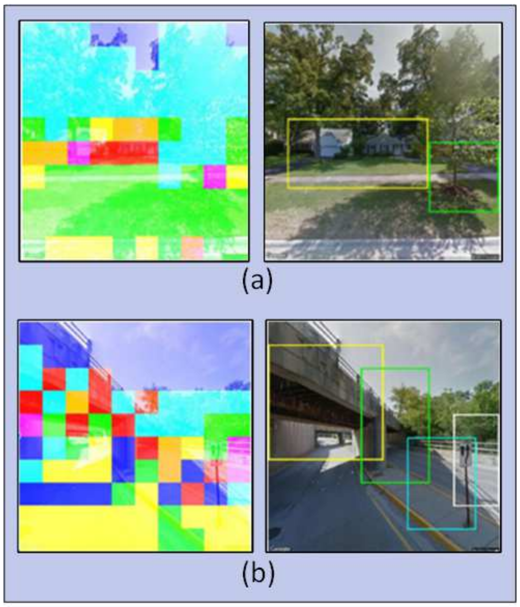

Figure 3: Two examples of ROI selection from our reference dataset. In (a), the algorithm selects a correct box and a wrong box. In (b), the algorithm selects a man-made structure (bridge) but the HOG features from that area are unlikely to produce meaningful matches. In the left-side images in both (a) and (b) different colors signify different semantic categories. The red patches indicate the house category and magenta indicates fence.

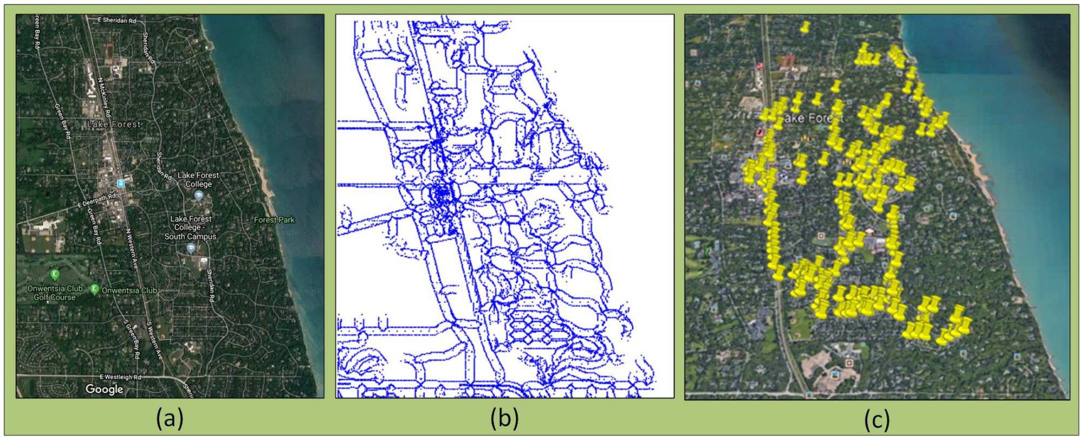

Figure 4: The locations of the images in our dataset. (a) shows the map of Lake Forest. (b) shows the distribution of the Google Street View images collected. (c) shows the locations of our query images.

2.2 Histograms of Oriented Gradients (HOG)

The idea of histograms of oriented gradients (HOG) is based on the observation that local features such as shapes of buildings and humans can be represented well by the angular distribution of local intensity gradients in the image [6]. HOG features are extracted from an image based on a series of normalized local histograms of image gradient orientations in a dense grid [6].

For this work, we used the HOG implementation distributed with the MATLAB Computer Vision System Toolbox. HOG features are extracted for a manually selected rectangle of the query image, and also from the rectangles around the house and fence category patches of the Google Street View images. We resize each rectangular window to 320 × 320 pixels and use a 32 × 32 cell size for the HOG feature extraction.

2.3 Matching

We initially used simple Euclidean distance to match the HOG features of the query image and the HOG features of the reference images. However, we found this does not always fetch images of the same building as the closest matches. To improve this result, we used the Enhanced Fisher Model (EFM) [8] based dimensionality reduction and feature extraction to extract more discriminative features for a novel shape matching algorithm.

2.3.1 The Enhanced Fisher Model (EFM). Principal component analysis, or 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 [16]. 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 [8]. The FLD method, if implemented in an inappropriate PCA space, may lead to overfitting. The EFM method, which applies an eigenvalue spectrum analysis criterion to choose the number of principal components to avoid overfitting, improves the generalization performance of the FLD.

In our experiments, we need a positive training set and a negative training set for training the EFM algorithm. We do it as follows. For the positive training set, we take the user-defined rectangle around a building in the query image and generate several more rectangles by shifting that rectangle on all sides around the original one. This procedure is inspired from [7] but it also reduces the dependence on the exact rectangle chosen by the user and decreases the chance of overfitting by slightly augmenting the positive training set. The HOG features from all these rectangles, which contain similar but not identical images, as our positive training set for the EFM algorithm. The negative training set is formed by HOG features from rectangles from the reference set that rank poorly on our simple HOG-Euclidean distance matching algorithm. We use 11 positive training samples and 110 negative training samples for this task. The maximum number of features that can be extracted by the EFM method is one less than the number of classes. Since this is a two-class problem, the EFM algorithm generates just one single feature. We use the difference of this feature from the query rectangle and from the reference rectangles as the distance measure for matching.

3 EXPERIMENTS

In this section, we will first give a brief description of the two datasets used for our experiments, and then discuss the search and retrieval performance of our novel EFM-HOG matching algorithm.

3.1 Dataset

For testing our proposed algorithm, we use the city of Lake Forest, Illinois. We created two datasets: one for query images and one for reference images. We generated our own images for the query dataset by walking around the city and taking photos with smartphone cameras. This dataset has 308 images. For the reference dataset we downloaded Google Street View [4] images from around the city. We download 8 overlapping Street View images from points 8 feet apart along every road in Lake Forest. This process downloaded 126, 000 images. From our 308 query images, we selected 128 images that contained buildings that were also visible

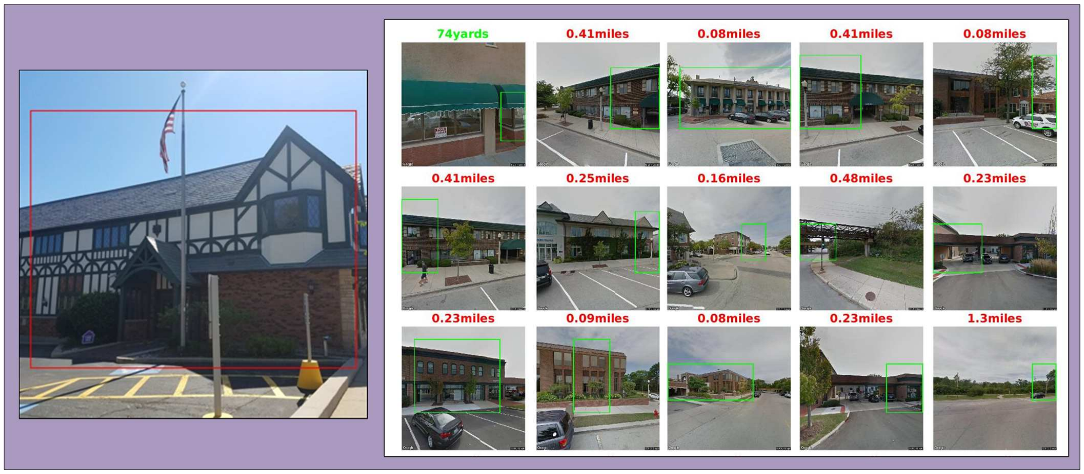

Figure 5: Top 15 images retrieved using the query on the left, and plain HOG matching. Green text indicates a result under 100 yards. No exact matches are found in the top 15 results.

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 images to bring the total up to 10, 000 images. This was our final reference set for the experiments. Figure 4 shows the Google Maps view of Lake Forest, the distribution of our reference set and the distribution of our query images.

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

3.2 Results

We ran two sets of experiments on our dataset. The first one does the retrival 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 Figure 5 and Figure 6. Figure 5 shows the top 15 traditional HOG matches for an example query image while Figure 6 shows the same number of matches using the EFM-HOG match for the same query. As can be seen, HOG fails to find even a single

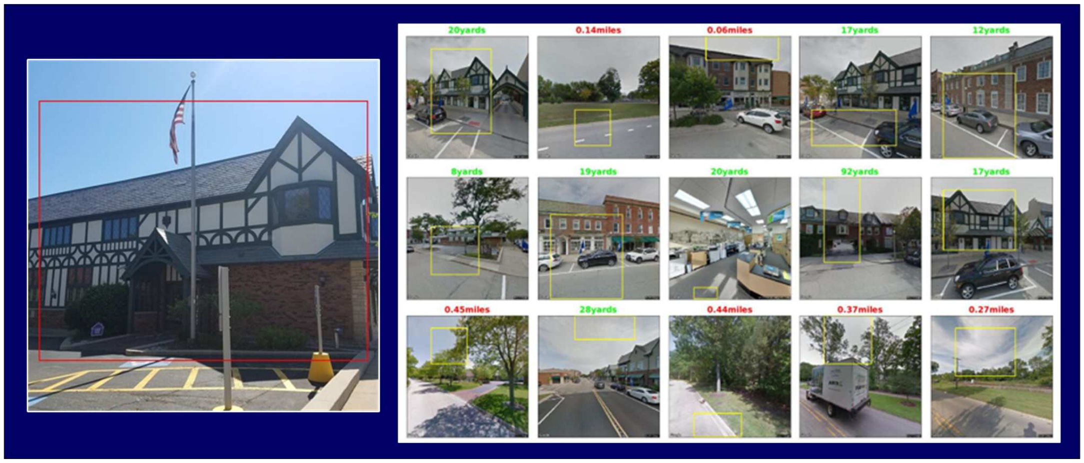

Figure 6: Top 15 images retrieved using the query on the left, and EFM-HOG matching. Green text indicates a result under 100 yards. The first retrieved image is of the exact same building.

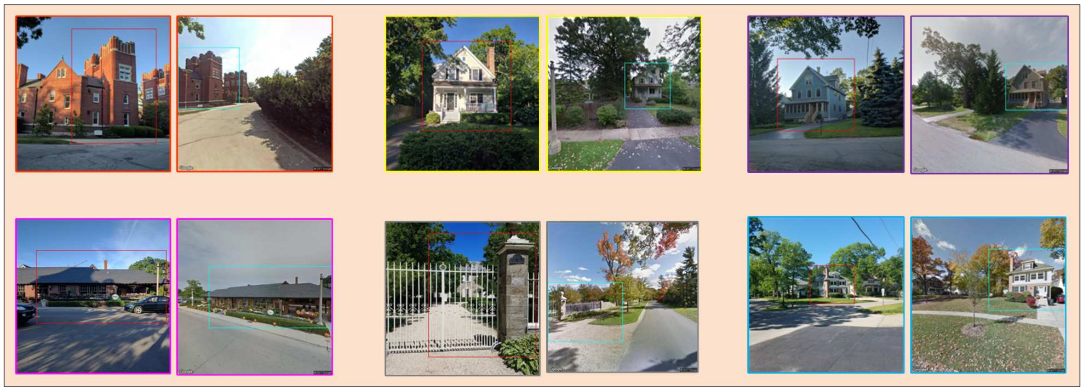

Figure 7: Successfully geo-located query images along with the retrieved Google Street View images that are exact matches for the query.

match for the building in the query image while EFM-HOG finds several.

Our EFM-HOG match program retrieved at least one image that was closer than 100 yards 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. One such query image and the top 15 retrieved images along with their geographic distances are shown in Figure 6. A few more successfully geo-localized buildings are shown in Figure 7. 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 in the retrieved images themselves were generated by our coarse semantic segmentation algorithm which is also a measure of the success of this algorithm.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a novel method of representing shape features from buildings using the EFM-HOG feature, and demonstrated its effectiveness for geo-localization on a city database 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 and demonstrate the proposed method to be superior for retrieval.

In future, we can try to develop a strategy for using a convolutional neural network (CNN) classifier for the semantic segmentation step which we could not do here due to lack of sufficient labeled data. We also plan to extend our dataset to cover other cities.

5 ACKNOWLEDGMENTS

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

6 REFERENCES

[1] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, “What makes Paris look like Paris?,” *ACM Transactions on Graphics*, vol. 31, no. 4, p. 101, 2012.

[2] J. Hays and A. A. Efros, “Im2gps: estimating geographic information from a single image,” in *CVPR*, pp. 1–8, 2008.

[3] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, “Image sequence geolocation with human travel priors,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 253–260, Sep. 2009.

[4] “Google StreetView.” https://www.google.com/maps.

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

[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, pp. 886–893, 2005.

[7] T. Malisiewicz, A. Gupta, and A. A. Efros, “Ensemble of exemplar-SVMs for object detection and beyond,” in *ICCV*, pp. 89–96, 2011.

[8] 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, pp. 132–137, 2000.

[9] A. Sinha, S. Banerji, and C. Liu, “Novel color Gabor-LBP-PHOG (GLP) descriptors for object and scene image classification,” in *ICVGIP*, p. 58, 2012.

[10] G. Singh and J. Košecká, “Introspective Semantic Segmentation,” in *WACV*, pp. 714–720, 2014.

[11] 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, pp. 51 – 59, 1996.

[12] 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, pp. 173 – 185, 2013.

[13] S. Banerji, A. Sinha, and C. Liu, “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*, pp. 490–497, 2013.

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

[15] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” 2008.

[16] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, second ed., 1990.