Discriminative Regions: A Substrate for Analyzing Life-logging Image Sequences

Mohammad Moghimi^{1,2}, Jacqueline Kerr¹, Eileen Johnson¹, Suneeta Godbole¹, and Serge Belongie²

¹ UC San Diego, La Jolla, CA 92093

² Cornell University, Ithaca, NY 14850
mmoghimi@cs.cornell.edu

Abstract. Life-logging devices are becoming ubiquitous, yet still processing and extracting information from the vast amount of data that is being captured is a very challenging task. We propose a method to find discriminative regions which we define as regions that are salient, consistent, repetitive and discriminative. We explain our fast and novel algorithm to discover the discriminative regions and show different applications for discriminative regions such as summarization, classification and image search. Our experiments show that our algorithm is able to find discriminative regions and discriminative patches in a short time and extracts great results on our life-logging SenseCam dataset.

1 Introduction

We are entering the age of wearable computing. Wearable devices are becoming more powerful and ubiquitous. Wearable cameras such as Google’s Project Glass, Narrative and SenseCam are adopted more and more by people. However, indexing the enormous amount of pictures that is being captured by these devices remains a challenge. Furthermore, these visual logs of people’s everyday lives provide a rich source of data for information extraction, including a variety of different Computer Vision tasks. In this paper, we propose a method to find regions of interest in the life-logging image sequences and also showcase a few applications of this representation.

In this work we are using SenseCam, a chest-mounted wearable camera, that periodically takes pictures every 20-30 seconds. This results in about two thousand images per day. Thus, there is a need for algorithms to analyze and extract information from these image sequences to facilitate the search process. The process would ideally be unsupervised or supervised with minimal human input.

We propose an algorithm that highlights regions of interest in life-logging images. We define these regions to be *salient* i.e. conspicuous regions, *consistent* i.e. reliably appearing in a few consecutive frames, *repetitive* i.e. frequently appearing in the image set, and *discriminative* i.e. are specific to a particular scene.

Saliency and consistency constraints help focus on regions that are in the foreground. We design a novel robust method to find consistent regions based on a forward-backward search in the feature space. Then those consistent regions are ranked based on their discriminative power and the number of their appearances. The motivation behind discriminative regions is that the parts that are visible everywhere are not very informative. Consider a scene where the user is working in his/her office. In this case, there are parts of the image that specifically belong to the office while others are shared with

Fig. 1. Discriminative Regions. First column is the input image. Second column shows consistent regions. Third column shows four discriminative patches discovered for set a of images from the same scene as the input image and Fourth column highlights the discriminative regions.

several other scenes. The idea is that the interesting regions are those that are specific to a scene and these regions should contain features that discriminate a given scene from others. Based on this definition, the office objects such as monitor, keyboard, etc that are visible in the office become the discriminative regions for the office scene.

The organization of this paper is as follows. Sec. 2 reviews the related works. Sec. 3 and Sec. 4 explain our algorithm that finds regions of interest that satisfy the four requirements. Sec. 5 discusses how we find discriminative regions after discovering discriminative patches. Later in Sec. 6, we show a few applications we build on top of these discriminative regions. Sec. 7 discusses our experiments and results and we conclude in Sec. 8.

2 Background

Ego-centric, First Person or Wearable Computer Vision has recently emerged as an area of great interest to computer vision researchers. Recently there have been works on finding objects and their relations to activities [6], [5] and [11]. These methods usually require extensive amounts of annotation of object segments, bounding boxes or other labels for each image or video frame in the sequence.

Another relevant line of work is object discovery [12] [7] [2] [10] [14] [15]. Most of these works address category level object discovery or object category discovery. These methods are designed for cases where multiple instances of each category are available in the dataset whereas in life-logging image sequences where there is usually one instance of each object category. The challenge in discovering objects in life-logging

focuses more on finding objects in a variety of imaging conditions e.g. viewing angle, illumination, and occlusion rather handling intra class variation of objects such as different shapes or sizes of objects. Among these methods, Kang et al. work [9] is the most relevant paper. Their idea is to find groups of mutually consistent image segments. They first ran segmentation on all of the images of their dataset to get many small segments and designed an algorithm find co-occurring segments. Later, those segments are joined to form object segments. In this process, since their goal was instance-level object discovery, they also imposed appearance and geometry constraints in their optimization. They tested their methods on lab controlled image sets. The difference between their work and this paper is that we are working with real life-logging images and this introduces many challenges such as the size of the dataset, uncontrolled illumination and occlusion. The other difference is that we focus more on finding candidate segment while their work is on how to connect these segments to form objects.

Our discriminative patch discovery method is similar to [3] and [13]. Singh et al. randomly subsample patches from a dataset of images with labels and apply an iterative discriminative learning method to find the patch clusters that demonstrate high discriminative power. Their iterative SVM learning is based on Ye et al. work on clustering [16]. We adapt the discriminative patch discovery idea and propose a simpler and faster algorithm to find discriminative patches.

3 Consistent Regions

As mentioned earlier, finding consistent regions is a crucial step in finding image regions containing objects of interest. We define consistent regions as parts that are visible across several consecutive frames in the image sequence. For example, in Fig. 3, the consistent regions are bike handle and hands while everything else lie on the background. The key factor is that the frame rate is variable between 20-30 as opposed to $\frac{1}{24}$ in SD videos. Due to the low frame rate, optical flow algorithms, object tracking and background subtraction methods fail to find the consistent regions.

Life-logging scenes can be classified into two groups: stationary scenes and dynamic scenes. Stationary or low motion scenes are dominant in the life-logging images. These scenes are mostly sitting cases including working with computer, watching TV, eating meals or sleeping and less occasional standing or other postures. Images of these scenes show very small change and large portions of these images are consistent. The other type of scenes are dynamic scenes in which usually the camera is translating in space. This includes biking, driving or walking. In the biking or driving scenes, there are parts of the image that are changing while some parts remain consistent. The consistent parts of the image may change their position in the next frames but they remain visible for at least a few frames. Walking or running are another types of dynamic scenes where the whole scene is significantly changing, do not have consistent regions.

There are several solutions for standard frame rate videos. Optical flow or background subtraction methods might solve this problem when the object movements are small. Fig. 3 shows a sample of three consecutive frames. In these frames, the bike's handle bar, some of parts of bike's body and hands constitute the consistent regions while the road, trees and sky compose the background. Conventional foreground-background segmentation methods fail because of two reasons: (1) the appearance of those regions change between consecutive frames due to lighting changes, shadows, viewing angle

Fig. 2. Consistent Patch Detection. Two patches from the right image are selected A and B . Their nearest neighbors in HOG feature space are labeled as A' and B' in the left image. Finally A'' and B'' are the nearest neighbors of A' and B' respectively.

and deformations. (2) the movement of these regions between these frames can be reasonably large e.g. the center of the bike handle may reside at the center of one frame and move to the far left in the next frame. Thus, we need a method that is able to handle more variations in the appearance and the movement of objects than the conventional methods do.

We propose a forward-backward search method to address these two issues which is based on searching small regions of the image in the images before or after it. To handle the location change, we search for each region in the next or previous frames in all of the possible locations. And we use HOG features to encode the structure of the region and cope with the variations in the appearance. Fig. 2 shows two consecutive frames from a biking scene. Consider a patch from the middle frame e.g. the center of the bike handle. We search for this patch in the previous frame to find its nearest neighbor. We do not constrain the search space to a local region and search throughout the previous frame for the best match. Our goal is to find another instance of the given patch in the previous frame if it is visible. One solution is to look at the distance between the given patch (A) and its nearest neighbor (A') in the feature space and accept the match as a correct match if the distance is lower than a threshold. But our experiments show that this approach is not very reliable, as the distances to the nearest neighbor for the wrong matches are sometimes lower than the correct matches. We propose not to rely on the distances in the feature space. Instead, we do another search for A' in the middle frame A'' . If the starting patch A is a consistent patch, then A'' would be the same as or very close to A . And if the starting patch e.g. B is not a consistent patch, then its nearest neighbor, B' , is a wrong match and B'' falls at a different location.

To implement the proposed method, first we extract dense HOG features from a grid of points from middle and left frames and create a kd-tree on the extracted features for each frame. Using the kd-trees, two searches are done for each patch in the given image. If the resulting patch is spatially close to the given patch, they the given patch is considered a consistent patch. The same procedure is done on the middle and right frames and the final consistency map is the intersection of the two maps. Fig. 3 shows three consistency maps where the middle one is the intersection of the left and right maps. Finally the last image shows the middle image with the overlaid consistency mask.

Fig. 3. Consistent Region Detection Sample. The top row show three consecutive images from a biking scene. The second row shows the consistency maps. The right binary map is generated using first two images and the left map is generated using the last two images. The middle map is the intersection of the two maps which is overlaid on top of the original image in the third row.

4 Discriminative Regions

After finding the salient and consistent regions, our goal is to group these regions to form clusters. Some of these clusters may represent meaningful regions and some clusters do not, even with very low intra cluster error. We define meaningful clusters as those who demonstrate discriminative power. For the discrimination, we use two sets (labels). Since we use image labels, this work can be considered as a weakly supervised method. Later we discuss ways to automatically extract information we need for these labels.

We assume that we have two image sets: $\mathbb{P}$ and $\mathbb{N}$. $\mathbb{P}$ contains a group of images from a particular scene. The scenes can be dynamic or static and they usually correspond to human activities such as biking, walking, driving, working in the office and watching TV. $\mathbb{N}$ is the universal or negative set and contains everything but $\mathbb{P}$. For example $\mathbb{P}$ may represents images of a biking scene while $\mathbb{N}$ is the set of all other images. Later in the Sec. 7, we discuss different combinations of $\mathbb{P}$ and $\mathbb{N}$.

$$\mathbb{U} = \mathbb{P} \cup \mathbb{N} \subset \mathbb{R}^d \quad (1)$$

More specifically $\mathbb{P}$ and $\mathbb{N}$ are the sets of extracted features from patches randomly sampled from the consistent and salient regions. We used histogram of oriented gradi-

Fig. 4. Discriminative Patch Discovery. (a) and (c) show examples of $\mathbb{P}$ and $\mathbb{N}$ respectively. (b) shows the discriminative patches that are discovered by our algorithm. Each column represents a discriminative patch and the patches in each column are the nearest neighbors of the patch in the first row. Finally, (d) shows the discriminative patches after the merging step.

ents (HOG) [1] and color (a^* and b^* channels of La^*b^* color space) to represent each patch. Furthermore, all of the selected patches from an image share the same labels as their image source. There is a one-to-one mapping between labels and sets. All of the patches in $\mathbb{P}$ have +1 label while patches in $\mathbb{N}$ have -1 label.

We propose to solve the discriminative patch discovery by looking at nearest neighbors of each feature point. $N_k(x)$ is defined as set of k nearest neighbors to x based on the distance function d . If we look at the distribution of labels in $N_k(x)$, we are able to find patches that most likely appear in either $\mathbb{P}$ or $\mathbb{N}$ and patches and appear in both sets.

$$N_k(x) = \{x' | x' \in \mathbb{U}, \forall y \in \mathbb{U} - N_k(x) : d(x, x') < d(x, y)\} \quad (2)$$

and $|N_k(x)| = k$

Specifically, we measure the label proportion in $N_k(x)$ by counting the number of element in each set (i.e. $\mathbb{P}$ or $\mathbb{N}$) and dividing the two numbers. The larger the value of division, the more discriminative the patch. The idea of iterative discriminative learning [3], [13] is to first run unsupervised clustering such as k-means and then at each iteration train an SVM classifier to separate each cluster from the rest and take the top matches of each classifier each cluster as positive in the next iteration to train another set of classifiers. Following their method, we implemented the iterative SVM learning but we did not see improvements over our nearest neighbor based discriminative patch discovery. There are two main reasons. The first reason is in the nature of our data. Since we are using life-logging images, there is inherent repetition of objects in the image sequence that help to achieve good clusters with our method. The second reason is that our results is much cleaner than k-means clustering which makes it hard for the iterations to improve the quality.

$$D(x) = \frac{|N_k(x) \cap \mathbb{P}|}{|N_k(x) \cap \mathbb{N}|} \quad (3)$$

Fig. 5. Frame selection to visualize spatial relationship among discriminative patches. Note that highlighted regions are discriminative patches found in the top ranked frames.

After calculation of the discriminativeness value $D(x)$ for all the patches in the $\mathbb{P}$, we sort patches based on their discriminativeness value in the decreasing order and pick those with highest values. In our experiments, we use $D(x) = 4$ as a cut-off value to select the discriminative patches. $D(x) \geq 4$ is equivalent to have more than or equal to 80% of the nearest neighbor points in $\mathbb{P}$.

$$\mathbb{D} = \{x | D(x) > \text{threshold}\} \quad (4)$$

Consider a patch that has a high discriminativeness value. It is expected for its nearest neighbors to have high discriminativeness values as well. The reason is that in that case the discriminative patch resides in a part of the feature space that is mainly filled with patches from $\mathbb{P}$. Because of this, some of the elements of $\mathbb{D}$ become very similar. Thus, we need to remove and merge some of the clusters. The last step is to merge the clusters that have significant amount of overlap. If two clusters are to be merged, we remove the cluster center that has less discriminativeness value from $\mathbb{D}$ and add its cluster members to the nearest neighbors of the cluster with higher discriminativeness value.

5 Spatial Relations Among Patches

Depending on the definition of objects and their sizes, the discriminative patches that we find may represent whole objects or object parts. When the patches are smaller than object, we need to link them together to build objects or discriminative regions. For that we need to find the spatial relationship between discriminative patches.

We experimented with different methods to find spatial relationships among patches. The best method turned out to be frame selection. In this method, we find images that contain high number of discriminative patches and deduce about the patches geometry based on their placements in the candidate frames. Specifically, we rank the frames based on the number of appearances on the discriminative patches in them. This is a very fast process since we only need to go through discriminative patches and their nearest neighbors and all of those patches have pointers to their original image frames. Fig. 5 shows four frames that very highly ranked for the biking scene.

Fig. 6. Example images and a sample of histogram of daily activities.

6 Applications

There are many uses for the discriminative regions we present in this paper. These regions depict information about the scene and the activity that is being performed. These regions are of importance in many applications such as classification of images, summarization and content-based image search.

Summarization. Since discriminative regions contain object and foreground information, they carry high level information about the activity that is being done from the image sequence and can be used for summarization of high level concepts. The idea is to select a minimal set of frames and cover all the discriminative patches. In this way most of the activities in the image sequences will be summarized with a small number of images.

Classification and Statistics. Discriminative Regions, by definition, are suitable for classification. Our algorithm is able to highlight regions that have discriminative power and can be used for classification. One example classifier, would be a linear classifier on the histogram of discriminative patches found in images.

The result of classification can be turned into histogram of activities or scenes e.g. Fig. 6. This will give a nice overview of a given day and can be accompanied by analysis of physical activities and recommendations about personal health.

Labeling and Search. The idea of labeling and search is similar to face labeling in Google Picasa and iPhoto. First these softwares detect faces in image galleries and cluster them into face clusters. They then ask the user to label representative faces from each cluster and using those label they label other faces in the image gallery.

In life-logging image sequences, we are not only looking for faces but anything else that are frequently appearing in the image set. After finding discriminative patches, we ask the user to label the representative patch from each cluster. Since the discriminative patches we find might be a part of an object, we show the whole image and highlight the discriminative patch and ask the user to label the enclosing object. The reason is that a cropped patch might not be informative enough to be perceived by the user but given the context it becomes easy to label. In this process, the user can choose one of the previously chosen labels or add a new label. In this way, we control the number of labels and discourage the user from adding excessive number of labels. Then, our algorithm propagate the labels to all the instances in the clusters. Fig. 7 demonstrates a few images with the propagated labels. Once we establish labels for images, we can

Fig. 7. Example images with the two propagated labels: bike and hand. We manually labeled top cluster centers and ran our algorithm to propagate the labels to all of the members of those clusters. Green boxes are hand labels and red boxes are bike labels.

use them for search and ranking. For Fig. 7, we ran our software and labeled the top 100 discriminative patches and our algorithm propagated the labels to all the candidate frames.

7 Experiments and Results

7.1 Data Acquisition and Annotation

The participants were adult cyclists recruited through a university based cycle-to-work network. Eligible participants were aged 18-70 years, were university employees, routinely bicycled for transportation. Each participant wore the SenseCam during waking hours for 35 days. They were instructed to perform their normal daily living activities, and turn off SenseCam in private time (e.g., bathroom). Excluding night time or private pictures that were removed before doing our experiments, our dataset contains over 360,000 images.

All of the images in our dataset, are labeled with position labels (i.e. Sitting, Standing Still, Walking/Running, Biking), and activity labels (such as Household Activity, Administrative Activity, Television, Other Screen Use and Eating). To the best of our knowledge, this becomes the largest SenseCam dataset with reliable labels. Our dataset is available on our lab’s website³.

7.2 Implementation

The first step is the consistency detection. For that, we randomly select 400 images from $\mathbb{P}$ and $\mathbb{N}$. Then, first we extract dense HOG from a grid of 70×50 points from all of the selected images. We chose to extract $8 \times 8 \times 31$ bin HOG from patches cropped around each point on the grid using HOG implementation of Girshick et al. [8]. Note that, throughout this paper we use a fixed patch size of 100×100 pixels. This size

³ <http://vision.ucsd.edu/%7emohammad/sensecam.dataset>

was selected to represent a cover a good size for object parts while the images have a resolution of 640×480 pixels.

For consistency detection of each image, we need to extract HOG features from three images. After that, we search for all of the feature points in the given frame and frames before and after that and vice versa. Thus, we need to perform a total of four searches that are implemented using [4]. Assuming we have run this process for a given image, for the next image, we only need to run the feature extraction on the new image and do two searches corresponding to the new image. Using this technique, we achieved speed of about 5 seconds per image on a MacBook Pro with 2.9 GHz CPU. We also do this process for all images but with two frame distance i.e. frames $i - 2$, i and $i + 2$. The consistency maps are later combined to be used in the patch sampling process.

Having the consistency maps computed, we extract 50 patches from each image that is selected for each study. The extracted patches have to satisfy two requirements: saliency and consistency. We use a low level saliency definition which is defined as a minimum standard deviation as well as a minimum on the length of the vectorized HOG feature. For meeting the consistency requirement, we randomly sample based on the binary consistency maps. The result of this step is about 20000 HOG-color feature points.

Then we compute the distance between all pairs using [4] and pick the top 50 nearest neighbors for each point. These nearest neighbor are then used for the ranking of discriminative patches. The whole process takes about 5 minutes assuming that the consistency maps are pre-computed.

7.3 Experiments

The first experiment is to show the effect of consistency maps. In this experiment, $\mathbb{P}$ is a set of biking image of a particular person where $\mathbb{N}$ is a set of image from an office scene. Fig. 8 show the discriminative patches that are discovered in this experiment. The top row is the discriminative patch discovery without using consistency detection. For second and third rows we used three and five images for consistency detection. As it suggests, using consistency maps helps to focus more at the foreground object rather than background regions such as sky, trees or road. It also shows that using more frames for consistency detection, results in better foreground object clusters. Additionally, since the patch sampling is more restricted and number of discriminative patches become less.

The second experiment is studying the effect of type of scene. For this experiment, we have used five different labels: Car, Biking, Watching TV, Walking/Running and Sitting. The image where chosen the image sets of two different persons. In each case $\mathbb{P}$ is defined to be the set of images having each of the five labels from one person's data and $\mathbb{N}$ is the images from the same label but from the other person. Fig. 9 shows the discovered discriminative patches and discriminative regions from top image selected by our frame selection process.

The Car and Biking scenes are similar in nature. There are some parts are visible in most of the images and some parts that are in the background. Fig. 9 shows that our algorithm is able to detect bike's parts, hands as well as parts of car's dashboard. The Watching TV and Sitting scenes are stationary scenes and our algorithm discovered monitor, TV, TV stand and well as other objects in the two rooms. Walking/Running is a completely dynamic scenes and set of discriminative patches are very small compared

Fig. 8. Effect of using consistency detection. The left part shows the discriminative patches without using consistency detection and the right part demonstrates the effect of consistency detection with five images. Using five images reduces the resulting discriminative patches that are mostly foreground patches.

to other labels and our algorithm focused at fence, some parts of sky and lamps that seems to be specific for one person.

8 Conclusion

In recent years, there has been a surge of interest in wearable cameras in the industry. Yet there is a need for efficient and accurate image analysis techniques for processing life-logging images. In this paper, we presented a method to extract discriminative regions with minimal supervision. We also hinted at how these discriminative regions can serve as a substrate for life-logging applications. Finally, we showed a few applications we can build on top of discriminative regions such as summarization and improved search and object part detection.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. vol. 1. IEEE (2005) 6
2. Deselaers, T., Alexe, B., Ferrari, V.: Localizing objects while learning their appearance. In: ECCV. Springer (2010) 2
3. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes paris look like paris? ACM Transactions on Graphics 31(4) (2012) 3, 6
4. Dollár, P.: Piotr’s Image and Video Matlab Toolbox (PMT). <http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html> 10
5. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: ICCV. IEEE (2011) 2
6. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR. IEEE (2011) 2
7. Fritz, M., Schiele, B.: Decomposition, discovery and detection of visual categories using topic models. In: CVPR. IEEE (2008) 2
8. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5. <http://people.cs.uchicago.edu/~rbg/latent-release5/> 9
9. Kang, H., Hebert, M., Kanade, T.: Discovering object instances from scenes of daily living. In: ICCV. IEEE (2011) 3

Fig. 9. Discriminative patches and discriminative regions highlights for different scenes. The scenes are Car, Biking, Watching TV, Walking/Running and Sitting from top to bottom respectively.

10. Kim, G., Faloutsos, C., Hebert, M.: Unsupervised modeling of object categories using link analysis techniques. In: CVPR. IEEE (2008) 2
11. Pirsaviash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: CVPR. IEEE (2012) 2
12. Russell, B.C., Freeman, W.T., Efros, A.A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR. vol. 2. IEEE (2006) 2
13. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV. Springer (2012) 3, 6
14. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering object categories in image collections (2005) 2
15. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: PAMI. vol. 2, pp. 101–108. IEEE (2000) 2
16. Ye, J., Zhao, Z., Wu, M.: Discriminative k-means for clustering. NIPS 20 (2007) 3