ANALYZING SEDENTARY BEHAVIOR IN LIFE-LOGGING IMAGES

Mohammad Moghimi a, Wanmin Wu b, Jacqueline Chen b, Suneeta Godbole b, Simon Marshall b, Jacqueline Kerr b, Serge Belongie c

aDepartment of Computer Science and Engineering, UC San Diego
bDepartment of Family and Preventive Medicine, UC San Diego
cDepartment of Computer Science, Cornell University and Cornell NYC Tech

ABSTRACT
We describe a study that aims to understand physical activity and sedentary behavior in free-living settings. We employed a wearable camera to record 3 to 5 days of imaging data with 40 participants, resulting in over 360,000 images. These images were then fully annotated by experienced staff with a rigorous coding protocol. We designed a deep learning based classifier in which we adapted a model that was originally trained for ImageNet [1]. We then added a spatio-temporal pyramid to our deep learning based classifier. Our results show our proposed method performs better than the state-of-the-art visual classification methods on our dataset. For most of the labels our system achieves more than 90% average accuracy across different individuals for frequent labels and more than 80% average accuracy for rare labels.

Index Terms—Wearable camera, Visual Classification, Deep Learning, Large Scale Image Analysis

1. INTRODUCTION
It is well understood that physical activity (PA) has a significant impact on health. Based on established scientific evidence, the 2008 Physical Activity Guidelines for Americans states that adults should get at least 150 minutes of moderate to vigorous physical activity (MVPA) every week [2]. Recently, new evidence shows that sedentary behavior (SB), or prolonged, unbroken sitting, is a distinct, important risk factor on health, independent of PA levels [3]. Common examples of SB include television (TV) watching, driving, workplace sitting, and computer use. Insufficient PA and/or excess of SB are associated with considerably increased risk for obesity, type 2 diabetes, cardiovascular disease, depression, and all-cause mortality.

Precisely measuring PA/SB is a fundamental, yet non-trivial task. Conventional self-reporting measures are subject to bias and errors. Recent advancements in wearable motion sensors technologies, particularly accelerometer, have enabled objective assessment of PA/SB. Machine-learning techniques have been used to classify everyday activities using annotated motion sensor data (e.g., [4]).

In this study, we investigate using a wearable camera (SenseCam [5]) for recording detailed PA/SB in life-logging images. SenseCam is a neck-worn, palm-sized device that automatically takes pictures about every 20-30 seconds. The images objectively capture daily activities in free-living settings and thus become a powerful tool for behavioral scientists to understand specific activity patterns.

After recent success of deep learning methods in object classification [6], researchers adapted these models for similar object recognition datasets such as Caltech-256 [7] and Caltech-256 [8]. We are going to take this further by applying object classifiers for the purpose of scene classification. We used a model that is pre-trained for object classification and applied it to SenseCam images. In some cases, it finds appropriate objects and sometimes it does not. Fig. 1 shows two example queries to the deep learning system. Objects that are detected in the two input images include Keyboard/Monitor and Tripod/Ribbon. Although, tripod and ribbon seem to be irrelevant, it is not hard to see the similarities between the scene and these object. We leverage these similarities and make connection between these predictions and the scenes. For example, prediction of tripod suggests road lines in a biking scene. The idea is somewhat similar to that of Object-Bank [9]. The difference is that the features that we propose to extract are based on a different classifier not specifically trained with relevant objects and applied at a coarser level.
We collected over 360,000 images from 40 participants over the course of a 3-5 days in real-life settings. The manual annotation effort on these images took about 18 man months. To help significantly accelerate this process, we developed a deep learning based method to automatically and efficiently process the images and classify PA/SB type. Furthermore, we extended the models with temporal and spatial pyramids to improve the performance. Using our novel image representation method, we achieved average accuracy of more than 80% for most of the classification labels and more than 95% accuracy for some of the binary classification problems for individual persons.

2. RELATED WORK

Image analysis has been researched for decades, but life-logging images pose a number of challenges: (1) the sheer amount of images captured (e.g., thousands per day) demands highly efficient processing algorithms and architectures. (2) the machine learning methods should focus on high level features for to extract high level concepts (e.g., using visual occurrence of a computer monitor to determine working with a computer).

MyLifeBits [10] was a pioneer project in this research space, but image analysis was not a focus then. More recently, Doherty et al. [11] presented an approach that classified lifestyle concepts such as tree, road, and door in SenseCam images. They used MPEG-7 features: ColorLayout and ScalableColor, the former being a set of DCT coefficients of the YCbCr image and the latter being the color histogram. This low-level feature extraction approach has also been used in their other reported work on life-logging images [12, 13]. We used high-level features instead and as we show in Sec. 4, high-level features are more suitable for object recognition (e.g., TV watching, computer use) in understanding PA/SB patterns. Recently, Pirsiavash et al. [14] presented a method for analyze activities in first-person videos using high level object based descriptors.

Our Contributions To the best of our knowledge, we gathered the largest life-logging dataset with reliable annotations. We then designed a deep learning based method to analyze the images. Instead of training a new network from scratch, we used a pre-trained model that was trained for object recognition [7] and showed that with the modifications, it worked better than the variety of the baseline algorithms on the dataset. We further improved the model by applying spatial and temporal pyramids.

3. CLASSIFICATION METHODS

In this section, we describe the image representation methods we propose to build our classification framework. Our goal is to have a diverse and powerful set of baselines to compare with our proposed deep learning based method.

Bag of Visual Words (BOW) Bag of visual words is a popular classic visual classification method that we used as a baseline in our experiments. We used dense SIFT [15] to extract interest points and used k-means clustering to create a codebook of visual words. Then each image was represented by a histogram over these visual words which was implemented with a fast kd-tree search data structure.

Fisher Encoding Fisher encoding has shown its promise on a number of popular object recognition and detection datasets such as PASCAL VOC, Caltech 256 [16] and ImageNet Large Scale Visual Recognition Challenge [17]. We encoded the same features that we extracted for the BOW method with Fisher Encoding.

Color For color representation, we used the same framework as our BOW method with the difference that we used RGB values for each point instead of SIFT features. For this, we created a bag of RGB points and clustered the points into 1000 clusters which were then used to create histogram representations.

GIST GIST is a holistic image feature proposed by Oliva et al. [18] based on a set of Gabor filters. We used 24 kernels at 3 scales separately applied to the RGB channels.

Deep Learning In 2012, Krizhevsky et al. presented a deep learning architecture that achieved the best results on the largest annotated object recognition dataset (ImageNet) [6]. Their model consists of 5 convolutional and 3 fully connected layers with an additional set of pooling and normalization layers. Recently, Girshick et al. achieved state-of-the-art results on PASCAL VOC detection dataset by using a network that is trained to classify ImageNet categories [19]. We claim that the ImageNet deep network is not only good for object recognition purposes but also for obtaining state-of-the-art results on non object centric data such as our life-logging dataset.

We use a model that is trained to classify ImageNet 2012 categories [7] and run feed forward passes on our SenseCam images and use the outputs of last layer to represent the input image. The last layer shows the likelihood of assigning the input to the 1000 object classes used to train the network. The representations are used to train a linear SVM model. To represent smaller object, we ran the feed forward path on multiple windows i.e. the whole image, four corners and center windows. We concatenated the outputs of the deep net to represent the image.

Classification We train linear support vector machines (SVM) [20] to classify different types of features that we extract from the images. Sec. 4.2 discusses the implementation details to address the time efficiency issues and explain why we use linear SVMs. In the test phase, given a linear SVM model, one only needs to compute a dot product between the SVM weight vector and the feature vector and use a threshold on the dot product value to determine a specific label.

Multi-frame encoding The pipeline that we have described so far is based on the single frame representation. To capture the dynamic changes in the image sequence, we propose to
use multiple frames as input to the classification. To encode the camera at a certain time, in addition to the closest frame to that time, we encode $k$ frames before and after and concatenate the encoded feature vectors. This simple yet effective method is capable of detecting the amount of change in the image sequence.

The basic idea behind feature concatenation is that instead of detecting the change using low level features, it is performed at the highest level with the encoded features. When there is no change between a frame and the frames before and after it (i.e., scene is stationary), the encoded features for those frames become similar. By examining the differences between elements of the final feature vector, a linear classifier is able to detect when change occurs. However, the change does not have to be detected with a separate classifier as change detection can be merged with the classification of different labels. Thus, apart from focusing on visual appearance of the scene, meanwhile, the classifier can detect change or motion which is essential for detecting some of the labels.

4. EXPERIMENTS

4.1. Data Acquisition and Annotation

We chose to use SenseCam, a wearable device built by Microsoft Research. It is a mature technology and has been used in many other research studies. 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, and were able to complete surveys in English. Participants provided informed consent and agreed to wear the SenseCam devices during waking hours for 3–5 days. Approximately 50% of the them wore the device during the weekend, and on at least 1 work day; the remainder wore the unit on weekdays only. They were instructed to perform their normal daily living activities, and turn off the SenseCam in private time (e.g., in the bathroom). Excluding night time or private pictures that were removed prior to our experiments, our dataset contains over 360,000 images.

We recruited staff to manually code these images using a rigorous protocol to ensure annotation reliability and accuracy. The annotation was performed using the SenseCam CLARITY browser. The images were annotated by position labels (i.e., Sitting, Standing Still, Walking/Running, Biking), and activity labels (e.g., 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.

Dataset Sharing Because of the sensitivity of the images, we only share extracted features to allow reproduction of the experiment results. Instructions on how to access the dataset is available on our lab’s website\(^1\). In addition, we provide the service to run new feature extractors upon request.

4.2. Results

Timing Feature extraction, encoding and classification are designed to be distributed processes. For feature extraction and encoding, we divided the task into 1000 distributed processes. Each process took about 5 and 3 minutes respectively on our computing cluster. That is equivalent to a total of 3.5 and 2 days of computing. The jobs finished in less an hour because the jobs were running in parallel on multiple machines. The codebook learning step took about 15 hours on one machine using randomly sampled feature points and SVM training took less than 2 minutes for each label and person. The SVM learning took about 1 day of combined time.

As stated earlier, we defined binary classification problems where the labels for training images were known and the classifier was expected to predict the labels for the new unseen testing images. We splitted the training and testing images with a uniform random distribution with a fixed proportion of training examples.

Our pilot experiments showed that SVMs performs better than other classifiers such as nearest neighbor classifier. We used liblinear [20] which is able to generate scores along with the binary predictions. The generated scores are in fact the inner product of the input feature vector with the learned weight vector. The pair of weight vector and offset (threshold) define the linear classifier. Having a fixed learned weight vector, we generated the ROC curve for the classifier by sweeping the threshold.

In all of our experiments, we trained and tested binary classifiers on individual persons’ data and generated an average ROC curve along with each label. To do this, 40 different classifiers were trained, and tested on the two different subsets of each person’s data. The measure we use to quantify the ROC curves is normalized accuracy. We normalized the accuracies based on the distribution of the test labels to be able to compare the performance of our classification method with respect to different labels. Note that a random classifier performs with 50% normalized accuracy even when there is a large unbalance between the two classes.

Fig. 2 shows the ROC curve of our method (deep learning, spatial and temporal pyramids) for different position labels. It shows that sitting and biking achieved the best results with average accuracy of 88.8% and 96.3% respectively. Standing Still showed the worst performance with the average accuracy

\(^1\)http://vision.ucsd.edu/~emohammad/sensecam_dataset
Table 1. This table compares the accuracy of different methods on our dataset. *stand st* and *stand mv* represent standing still and standing/moving.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sitting</th>
<th>Stand st</th>
<th>Stand mv</th>
<th>Walk/Run</th>
<th>Biking</th>
</tr>
</thead>
<tbody>
<tr>
<td>bag of words</td>
<td>79.9</td>
<td>69.7</td>
<td>62.0</td>
<td>73.3</td>
<td>92.8</td>
</tr>
<tr>
<td>fisher</td>
<td>86.8</td>
<td>76.2</td>
<td>76.3</td>
<td>83.2</td>
<td>94.4</td>
</tr>
<tr>
<td>decaf</td>
<td>93.4</td>
<td>76.8</td>
<td>83.3</td>
<td>89.9</td>
<td>97.1</td>
</tr>
<tr>
<td>pyramid decaf</td>
<td>92.8</td>
<td>80.3</td>
<td>85.3</td>
<td>89.7</td>
<td>98.5</td>
</tr>
<tr>
<td>bag of color</td>
<td>82.6</td>
<td>62.0</td>
<td>64.0</td>
<td>83.6</td>
<td>95.3</td>
</tr>
<tr>
<td>gist</td>
<td>75.2</td>
<td>62.9</td>
<td>62.7</td>
<td>74.1</td>
<td>90.1</td>
</tr>
</tbody>
</table>

Fig. 2. ROC curves for classification (decaf) of different position labels along with their average normalized accuracies.

Fig. 3. The effect of changing the number of frames in our multi-frame coding method. The sitting label is used for this experiment.

Fig. 4. The effect of changing the number of images used for training.

Fig. 5. The effect of changing the SVM regularization term.

of 74.5%. The main reason is that there were not enough samples for the Standing Still class however the raw accuracy was more than 99%. Table 1 compares our methods (*decaf* and *decaf_pyramid*) with other baselines. As the table shows, *decaf* worked the best for 2 labels and *decaf_pyramid* for the other 3.

Fig. 3 compares our multi-frame coding approach with single frame coding. In this case, the average accuracy improved from 89.4% to 93.4%. In this experiment, sedentary label was chosen for illustrative purposes. For this label, using 7 frames which is equivalent to 2–3 minutes results in the best classification accuracy. Other labels have different optimal points, but they are all about 7 frames. Using multi-frame coding improves accuracy by about 3–4% in average.

Fig. 4 shows the effect of changing the amount of training data. Most of the curves are almost flat which shows that the same accuracy can be achieved using as little as 10% of the training the classifier. This shows that the classifier can generalize well even with small amount of training data. In this experiment, the test set shrinks as the training set grows.

Fig. 5 shows the effect of the regularization term $\frac{1}{C}$ in the SVM classification. Our experiments showed that choosing low regularization term (i.e. high $C$) improves the overall accuracy. The reason is in the imbalance between the two classes. With high regularization parameter, the model tends to overfit to the dominating class and therefore it reduces the accuracy. Based on these experiments, we use $C = 100$ in all of our experiments.

5. CONCLUSIONS

In recent years, there has been a surge of interest in wearable cameras. This presents a need for efficient and accurate image analysis techniques for processing life-logging images. We presented a deep learning based classification method built on top of an ImageNet classifier and showed that it performs better than all baselines. Furthermore, we designed a spatial pyramid technique in time and space to further improve the classification results. We achieved more than 80% average accuracy for most of the labels and more than 90% accuracy for classification of some of the individuals’ data.
6. REFERENCES


