DETECTING FOOT-CHASES FROM POLICE BODY-WORN VIDEO[*]

RAFAEL AGUAYO[†] ALEJANDRO CAMACHO[‡] PIYALI MUKHERJEE[§] QI YANG[¶]
SPONSORS: HAYDEN SCHAEFFER [‖] AND P. JEFFREY BRANTINGHAM [**]

Abstract. Existing methods to record interactions between the public and police officers are unable to capture the entirety of police-public interactions. In order to provide a comprehensive understanding of these interactions, the Los Angeles Police Department (LAPD) intends to utilize *Body-Worn Video* (BWV) collected from cameras fastened to their officers. BWV provides a novel means to collect fine-grained information about police-public interactions. The purpose of this project is to identify foot-chases from the videos using machine-learning algorithms. Our proposed algorithm uses the *Bag-of-Intrinsic-Words* algorithm followed by classification via support-vector machines. Our training dataset consists of 100 training videos (20 foot-chase & 80 non-foot-chase), and a test dataset of 60 LAPD videos (4 foot-chase & 56 non-foot-chase). We achieved results of 91.6% testing accuracy.

1. Introduction. Studying the interaction between police and the public is often a difficult task because little information regarding police-public interaction is retained through activity logs and written reports [4]. In 2014, the Los Angeles Police Department (LAPD) implemented the use of chest-mounted *Body-Worn Video* (BWV) in small deployments, as seen in Figure 1.1, with the purpose of collecting more information regarding police-public interactions. BWV provides another line of evidence for outcomes of interactions.

BWV generates massive volumes of data that can be difficult to analyze. Due to the size of the BWV dataset, it is infeasible for police officers to view all the videos in order to find specific interactions, *e.g.* foot-chases. Since many BWV videos are likely to be used as evidence, an automated labeling mechanism can save valuable time and resources while maintaining confidentiality of the data. Our work focuses on devising a learning algorithm that can automatically detect whether a particular video contains a foot-chase or not. From our exploration of the literature, such a project is the first of its kind to have been attempted.

This paper is organized by the following sections. Section 2 discusses previous work done relevant to video and image processing. Section 3 describes the BWV data and preprocessing procedure. Section 4 discusses the mathematical background behind the feature-extraction methodology, and, our proposed Bag-of-Intrinsic-Words method. Section 5 presents our results and analysis on the BWV provided by the LAPD.

2. Previous and Related Work. In recent years, researchers have devised various methods for filtering, parsing, recognizing objects, and classifying video data. Since video data is composed of a sequence of frames, there is significant overlap between video and image processing techniques [3]. In [9], the authors presented a unified view of the different statistical structure of natural images. These models were designed to reflect certain properties of intrinsic systems. The authors in [6]

[†] UNIVERSITY OF CALIFORNIA, SAN DIEGO
[‡] CALIFORNIA STATE UNIVERSITY, FULLERTON
[§] COLUMBIA UNIVERSITY
[¶] UNIVERSITY OF SOUTHERN CALIFORNIA
[‖] CARNEGIE MELLON UNIVERSITY
[**] UNIVERSITY OF CALIFORNIA, LOS ANGELES

[*] This work was supported by the Institute of Pure and Applied Mathematics, University of California, Los Angeles.

Copyright © SIAM
Unauthorized reproduction of this article is prohibited

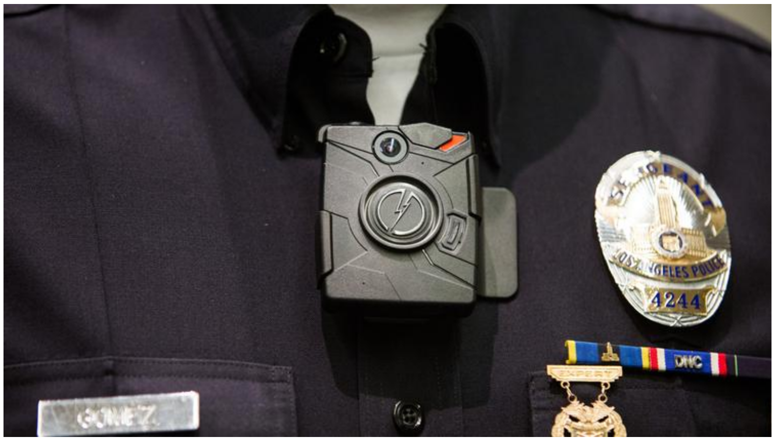

Fig. 1.1. *Sgt. Dan Gomez of the LAPD wearing a BWV camera. BWV provides a first-person perspective of the officer. (Marcus Yam for the Los Angeles Times. Copyright ©2015. Reprinted with permission.)*

considered an incremental-batch Bayesian probabilistic model in order to learn object categories from still images. Their approach allowed the model to learn the parameters in an incremental fashion. Thus, real-time learning was feasible.

There are several semi-supervised approaches in computer vision and imaging studies [18, 16, 12, 5, 22, 23] that use keypoints to extract information from videos. Keypoints in images can be identified in frames using descriptors such as SIFT (scale-invariant feature transform) and PCA-SIFT [11, 19, 2]. In [1], the authors developed SURF (Speeded-up Robust Features) from SIFT to identify points of interest using existing image edge-detectors and descriptors. Semi-supervised approaches have been used to extract quantitative information from images and videos using SURF [11, 19, 2, 1, 17]. SURF tests to see if distorted images of an original image contain the same points of interest. A qualitative comparison performed by [17] showed that SURF features are robust to noise, displacements, and most geometric and contrast-base transformations.

Bag-of-Visual-Words is a method proposed by [23] for classifying scenes. The idea behind [23] is derived from a text-based classification schema, where weighted terms and frequency are used to classify documents. The visual categories in images (such as blurriness, daytime or night shots, many or few entities in focus) and words in documents is analogous. Each visual property pertains to a category of images that this particular shot could belong to, just as a word in the document provides insight on the topics that the document could belong to. This analogy allows us to apply a tool for retrieving information [24] across the video data set just as we would to a document.

In order to identify valid visual ”words”, we looked at the work of auhtors in [8] and in [22]. In [8], the authors focused classification on a small region of an image showing that discriminatory localization classification works well with weakly-labeled data [8, 20, 14]. In [22], the authors evaluated previously proposed local spatio-temporal features for action recognition using a standard bag-of-features support vector machine (SVM).

The video data used in previous works is collected from stable and stationary cameras (*e.g.* Hollywood datasets as per [15]).The Hollywood dataset, in particular, contains frames that have clearly defined and well-lit scenes, followed by carefully directed camera and actor movements. There are few occlusions to the camera view, if any. These properties set the Hollywood video dataset at a considerable advantage over our dataset. The methods used in [15] do not appear to work as well with videos that are as grainy and distorted as videos collected from moving officers.

The videos in our dataset undergo severe geometric and photometric distortions, and contain many poor-resolution and grainy scenes. Many videos are recorded in low light, which makes analysis of events occurring within the video difficult. Furthermore, the videos are frequently obstructed by multiple objects such as hands, jacket lapels, car bodies, multiple people and other close-focus moving objects. Thus, the BWV dataset requires tailored algorithms.

The authors also acknowledge that while several deep-learning methods have shown successes (such as [13]), they require an intensive investment of computational resources that aren't easily available to our users. Furthermore, as with most deep learning methods, the unsupervised algorithm's choice of features is brittle and often requires significant time to train correctly.

3. Body-Worn Video Data. The BWV dataset provided by LAPD is composed of 691 videos (500GB) collected from Central SCI officers in the field from a span of two weeks, December 28, 2015 to January 3, 2014 and May 24, 2015 to May 31, 2015. The videos were collected from Skid Row, Los Angeles. Skid Row is a community commonly affected by problems of homelessness, drug-abuse and assault [21, 7].

The video filename contains the time at which the clip was recorded. Some files indicate that the video was recorded at the same situation but from the perspective of another officer. As expected, the poor lighting, contrast changes, and sudden scene distortions in the videos do not make for ideal compositions, unlike Hollywood videos (See 2 and [15]).

The video is recorded at a resolution of 640 × 480 pixels at 30 frames per seconds and compressed into an MPEG-4 format. For confidentiality purposes, the videos do not contain audio. The statistics regarding the length of the video are summarized in Table 3.1.

Median	9 min
Maximum	30 min
Minimum	12 sec
Total length	130 hr

Table 3.1
Statistics for the length of the BWV videos.

We perform preprocessing procedures to reduce the size of the files in order to prevent overloading memory resources. We reduce the resolution of the video from 640 × 480 to 320 × 240. Subsequently, the data is partitioned into training and testing sets. We further splice the training videos into 30 second clips containing specific actions. A '-1' indicates that the video contains a non-foot-chase event, while '1' indicates that the video contains a foot-chase event. Similarly, testing videos are spliced into 30 sec videos and assigned a label. The testing labels are used solely for accuracy measures. Two videos in our given dataset contained foot-chases.

To train our algorithm, we needed a sufficient number of existing foot-chase videos in order to identify future ones. Due to the sparsity of existing foot-chases, we recorded additional videos to simulate a diverse range of running, walking, and, other movements that were observed from the given dataset. We recorded a total of 67 videos out of which 18 simulated a foot-chase.

For training, we combined our simulated data with BWV data for a total of 100 training videos. We tested on 60 LAPD BWV videos. In this subset, four videos were manually labeled as foot-chase. This proportion was selected to provide a consistent basis for measuring training and testing error on our algorithm.

4. Our Algorithm. Our approach is derived from a text-based classification method on documents known as Bag-of-Words (from [23]). Bag-of-Words is a sparse vector that contains occurrence counts of categories, usually words present in the document to be classified. We develop our own Bag-of-Intrinsic-Words algorithm to classify our dataset as foot-chase videos or non-foot-chase-videos. We consider a set of videos,

$$U_j(x,t) \in \mathbb{R}^{N \times T_j} \tag{4.1}$$

where N is the number of pixels per frame, T_j is the number of frames in a video, and j is the video index. The frame rate for our BWV dataset is 30 frames/sec.

First, we use built-in MATLAB SURF (Speeded Up Robust Features) keypoint detection algorithm to generate the SURF feature vectors. We initially split the data into three sets: input for Bag-of-Intrinsic-Words, training and testing. We implement a clustering algorithm on the first set to generate a set of descriptors, or "intrinsic words". Then, we use our Bag-of-Intrinsic-Words to test for similarities among videos and create a feature histogram for each video. Lastly, we run a feature-based classifier on the feature histograms, which returns the predicted label of the data set. A visual representation of the feature extraction methodology and pseudocode our algorithm are presented in Figure 4.2 and Algorithm 1, respectively.

4.1. SURF (Speeded Up Robust Features). SURF is one of many frameworks that detect keypoints within images. SURF provides not only the location of the keypoints detected in the image but also a radius of the scale at which they are detected as well as an orientation vector for every keypoint detected. Figure 4.1 depicts the SURF keypoints and their radii.

We extract a feature vector, also known as descriptor, from the regions identified by SURF. The descriptor is calculated by dividing the neighborhood of a keypoint into sub-regions until the orientation is computed at the smallest scale. These orientations are then pooled towards the higher regions until a resultant orientation of the keypoint is determined. The descriptor saves each sub-region calculation as an entry. To obtain same number of SURF feature vectors from each input, we partition each video to create 30 second clips by collecting every 900 consecutive frames from $U_j(x,t)$. To further reduce the data size and run time, we down sample the video clips by taking every p-th frame from the clip. The downsampled video is represented by $u_i(x,t) \in \mathbb{U}_j(x,t)$.

$$u_i(x,t) \in \mathbb{R}^{N \times P} \tag{4.2}$$

where $P = 900/p$ and i is the video clip index.

We define F as function that maps the clips to the corresponding SURF feature descriptors: $F\not\psi\!: \mathbb{R}^{N \times P\psi} \rightarrow \mathbb{R}^{S \times Q\psi}$where in practice $S\not\psi = 64$ since the SURF feature vectors lie in 64 dimensional space and $Q\not\psi = 5P\not\psi$if we choose five strongest SURF feature points.

Now, let $A_{S\times Q}$ be the matrix which contains all SURF feature descriptors,

$$A_{S\times Q} = [F(u_1), ...F(u_n)]. \tag{4.3}$$

Then we partition the feature matrix, $A_{S\times Q}$, into three sets: input for Bag-of-Intrinsic-Words, training data, and, test data. These sets are $A^1_{S\times Q_1}$, $A^2_{S\times Q_2}$, $A^3_{S\times Q_3}$, respectively. where $S \times Q_i$ is the dimension of matrices after partition.

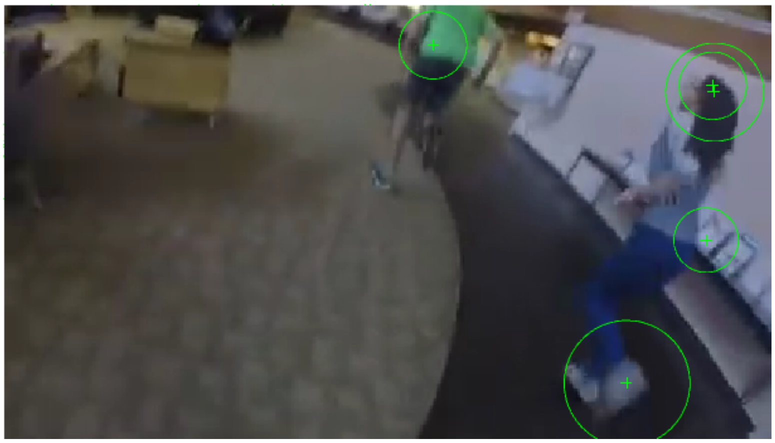

Fig. 4.1. *The graph presents the surf features extracted from a test video simulating real data conditions. The SURF feature descriptors are extracted from the radial regions identified by the SURF algorithm.*

4.2. Our Bag-of-Intrinsic-Words. Our Bag-of-Intrinsic-Words describes the local patterns of the videos with smaller dimensions. This method allows us to compare all videos with an unvaried Bag-of-Intrinsic-Words and test for the similarities among videos. Ultimately, this helps us to distinguish whether a clip contains a foot-chase or not.

After partitioning the SURF feature vectors into three sets, we cluster the feature vectors from the first set, $A^1_{S\times Q_1}$ (input for Bag-of-Intrinsic-Words) using the k-means clustering algorithm. We apply the k-means algorithm to $A^1_{S\times Q_1}$ to create our intrinsic words. k-means clustering partitions a dataset into k distinct clusters [10], thereby generating our features. To perform k-means clustering, we first determine the specific number of desired clusters. In this study, we varied k from 100 to 1500. k-means assigns each observation to exactly one of the k clusters based on a distance metric. The k-means algorithm used here measures distance using the standard Euclidean metric. To find the optimal number of k clusters, we minimized the *within-cluster variation*. Each centroid created by k-means corresponds to an intrinsic word. We represent the visual words in a matrix $W_{S\times M}$ where each column corresponds to a word. By testing on different M, we find that the optimal number of intrinsic-words is 500.

4.3. Feature Histogram. After we obtain our Bag-of-Intrinsic-Words, we use the idea of Approximate Nearest Neighbor to assign each feature vector from the training and test dataset to the nearest intrinsic word measured by a distance function. Each feature vector belongs to one and only one cluster.

Mathematically, for each new video $U_j(x,t)$, we partition it into small video clips $u_i(x,t)$ where j corresponds to the video index and i corresponds to the video clip index. We then calculate $A_{S\times Q}$ and $D_j(Y)$, the Euclidean distance between each SURF feature descriptor and each intrinsic word from $W_{S\times M}$.

For simplicity we use the Euclidean norm. We assign the SURF feature descriptor to the nearest intrinsic-word, denoted by J.

$$D_j(Y) = \quad \mathrm{Dist}(Y, W_j) \quad := \left(\sqrt{\sum_{i=1}^{P}(Y(i) - W(i,j))^2}\right) \tag{4.4}$$

$$J = \quad \mathrm{argmin}_j D_j(Y) \tag{4.5}$$

The frequency of the intrinsic words for each video creates a feature histogram. If another video has a similar histogram, we expect them to have similar video content. In other words, we expect videos that have similar histograms to share comparable features. The feature histograms are then classified using Matlab's Support Vector Machine (SVM).

We used Matlab's in-built fitcsvm function trained using a linear kernel. We attempted to save the cost of sorting through the data by penalizing the false-negatives twice, rather than penalize once, sort and then filter the results. We believe this was a good decision given the size of our dataset and the fact that actual foot-chase videos were a very small fraction of the whole dataset. Thus, we weigh the cost of making the error heavily. For more explanation on why the cost of errors was heavily penlized, please refer to Section 5. Lastly, within each penalizing operation, we applied a ten-fold cross-validation using Matlab's inbuilt tools.

```
Algorithm 1 Our Bag-of-intrinsic-Words Algorithm
    Step 1: Partition the video data into 30 second clips.
    Step 2: Select top five SURF points from each frame and extract the feature vector from those SURF points to form a matrix.
    Step 3: Partition the feature matrix into three sets: $A^1$, input for Bag-of-Intrinsic-Words; $A^2$, training data; and $A^3$, test data.
    Step 4: Perform $k$-means on $A^1$ to create our Bag-of-Intrinsic-Words. The optimal size for our dataset is $k = 500$.
    Step 5: Assign each SURF point from $A^2$ and $A^3$ to a intrinsic word. Count the frequency of intrinsic words for all SURF points from the same video.
    Step 6: Create a histogram of feature occurrence.
```

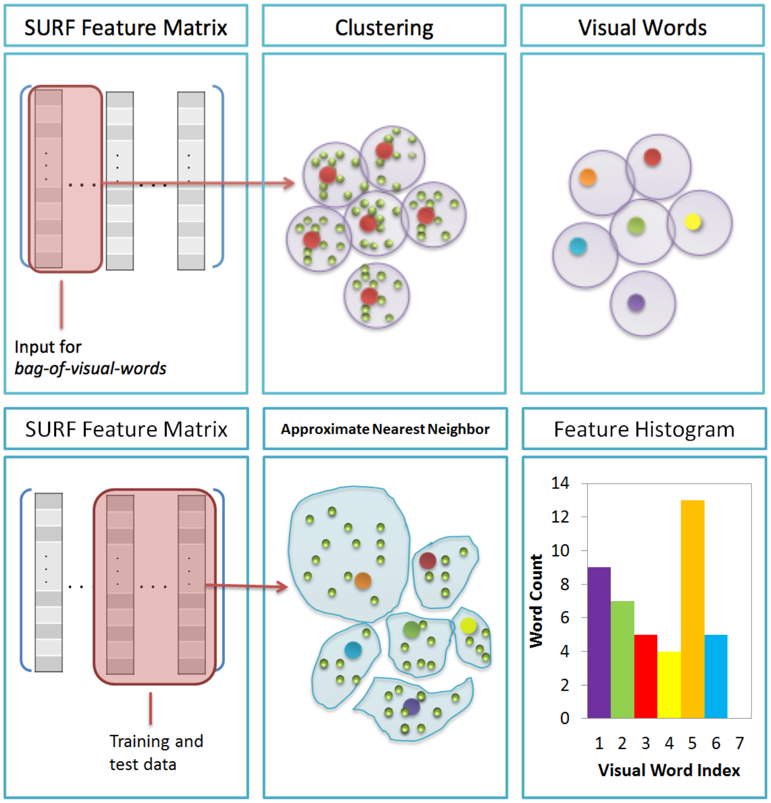

Fig. 4.2. *(From Top-Left to Bottom-Right) This figure contains a visualization of our feature extraction pipeline.*

5. Results and Analysis. For our algorithm, we partition the BWV dataset into three sets: $A^1_{S\times Q_1}$, input for Bag-of-Intrinsic-Words; $A^2_{S\times Q_2}$, training data; and $A^3_{S\times Q_3}$, test data. After obtaining the intrinsic words from $A^1_{S\times Q_1}$, we apply our algorithm to $A^2_{S\times Q_2}$ and test on $A^3_{S\times Q_3}$ to get the accuracy rate. Each video is also assigned an action label, foot-chase or non-foot-chase.

In Figure 5.1, we see that our testing accuracy starts to increase around 20 videos. As we increase the number of videos, we are better able to classify foot-chase from non-foot-chase videos until our accuracy plateaus around 30 videos. Due to time constraints and computational costs, we retain our testing set at 60 videos. In Figure 5.2, we see that the BWV data is well represented after identifying 500 intrinsic words.

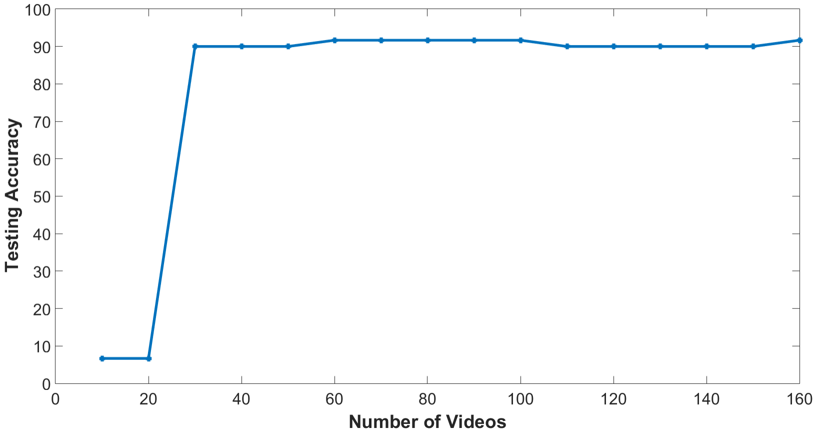

Fig. 5.1. *This plot shows the variability in accuracy when using different training sets.*

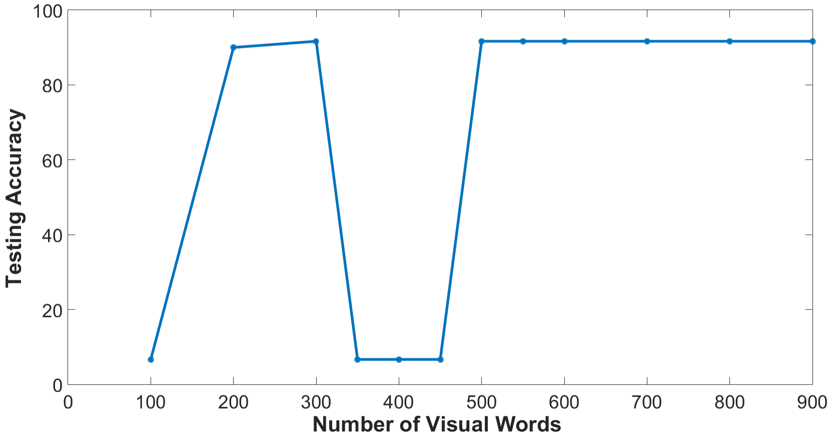

Fig. 5.2. *This plot shows the variability in accuracy when using different number of intrinsic words, or k in k-means clustering algorithm. The optimal number of intrinsic words is 500 since the accuracy plateaus.*

The primary metric we considered for measuring the results of SVM are false negatives and false positives. A false negative prediction means that there exists a video in which a foot-chase instance occurred, but was mislabeled as a non-foot-chase. A false positive prediction means that there exists a video in which a non-foot-chase event occurred, but was mislabeled as a foot-chase. It is important that our algorithm capture at least all of the correct foot-chases, even at the risk of identifying a few non-foot-chases. We seek to to minimize the number of false negatives so that we can detect all of the running videos even at the cost of mislabeling a few of the walking videos. We think this is a more efficient method to represent the strength of our algorithms as a mere accuracy score does not reflect the inherent biases in our data set, namely that we have significantly lesser foot-chase videos as compared to non-foot-chase videos. We reason that the cost of having a user re-verify the running videos over our results is negligible compared to the cost of having a user search through the entire dataset for a running video that was potentially undetected. Our algorithm tagged 8 running videos, of which we know that only 3 were present in our dataset. Therefore, our algorithm generates an accuracy score of 91.6% with 5 false positives and 0 false negatives.

Method	Accuracy	False Negative	False Positive
Bag of Intrinsic Words	91.6%	0	5

TABLE 5.1
This table contains the results for our Bag-of-Intrinsic-Words algorithm.

6. Conclusion. This paper covers a machine learning algorithm for identifying police foot-chase videos. Given the nature of the data and from our exploration of previous research, we posit that our work is the first of its kind to have been attempted in the field. Through our exploratory analysis we also noticed some trends that we will discuss. In Figure 5.1 a possible reason for our accuracy plateauing is due to the extra videos adding no additional value to our model. In Figure 5.2, the reason for the accuracy drop in the 300-400 intrinsic word region is due to the SVM failing to converge which will most likely lead to a lack in classification. Going back to our dataset in Table 5.1, we see that we had a very small quantity of true positives. We penalized false negatives (running videos mislabeled as walking) more heavily than false positives (walking videos mislabeled as running). By penalizing false negatives strongly, we ensured that the algorithm does not miss any running videos even at the cost of mislabeling a few walking videos as running ones. Our Bag-of-Intrinsic-Words algorithm returns results of 91.6% accuracy with 5 false positives and 0 false negatives.

7. Acknowledgements. We would like to thank the Los Angeles Police Department (LAPD) and the Los Angeles Police Foundation (LASF) for sponsoring this project, C.C Sean Malinowski, Sgt. Javier Macias, Sgt. Dan Gomez, and Ofc. Benjamin Hong. Furthermore, we extend our gratitude to the LAPD Air Support Division for the helicopter fly-along. We would like to thank our industrial mentor, Dr. P. Jeffrey Brantingham for his support, and Dr. Hayden Schaeffer for his mentoring and guidance during this project. Lastly, we would also like to thank Dr. Mike Raugh, Dr. Russel Caflisch, Dimi Mavalski, and the IPAM staff. The BWV data was analyzed under UCLA IRB protocol 15-000867. The use of the algorithm and analytics was limited only to the samples and has not be deployed by LAPD.

References.

[1] H. Bay, T. Tuytelaars, and L. Van Gool. “Speeded-up robust features (SURF)”. In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[2] M. Chen and A. Hauptmann. “Mosift: Recognizing human actions in surveillance videos”. In: (2009).

[3] S. Chikkerur et al. “Objective video quality assessment methods: A classification, review, and performance comparison”. In: *Broadcasting, IEEE Transactions on* 57.2 (2011), pp. 165–182.

[4] C. Eith and M. R. Durose. “Contacts between police and the public, 2008”. In: *Washington, DC* (2011).

[5] Clement Farabet et al. “Learning hierarchical features for scene labeling”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013), pp. 1915–1929.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories”. In: *Computer Vision and Image Understanding* 106.1 (2007), pp. 59–70.

[7] Bernard E Harcourt. “Policing LA’s Skid Row: Crime and Real Estate Redevelopment in Downtown Los Angeles (An Experiment in Real Time)”. In: *U. Chi. Legal F.* (2005), p. 325.

[8] Minh Hoai et al. “Learning discriminative localization from weakly labeled data”. In: *Pattern Recognition* 47.3 (2014), pp. 1523–1534.

[9] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.* Vol. 39. Springer Science & Business Media, 2009.

[10] Gareth James et al. *An introduction to statistical learning.* Springer, 2013.

[11] Yan Ke and Rahul Sukthankar. “PCA-SIFT: A more distinctive representation for local image descriptors”. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on.* Vol. 2. IEEE. 2004, pp. II–506.

[12] Ivan Laptev et al. “Learning realistic human actions from movies”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE. 2008, pp. 1–8.

[13] Quoc V Le et al. “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE. 2011, pp. 3361–3368.

[14] Jingen Liu, Jiebo Luo, and Mubarak Shah. “Recognizing realistic actions from videos in the wild”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE. 2009, pp. 1996–2003.

[15] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. “Actions in Context”. In: *IEEE Conference on Computer Vision & Pattern Recognition.* 2009.

[16] Michael Marszalek, Ivan Laptev, and Cordelia Schmid. “Actions in context”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE. 2009, pp. 2929–2936.

[17] Akitsugu Noguchi and Keiji Yanai. “A SURF-Based Spatio-Temporal Feature for Feature-Fusion-Based Action Recognition”. English. In: *Trends and Topics in Computer Vision.* Ed. by KiriakosN. Kutulakos. Vol. 6553. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 153–167.

[18] Rajat Raina et al. "Self-taught learning: transfer learning from unlabeled data". In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 759–766.

[19] Paul Scovanner, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition". In: *Proceedings of the 15th international conference on Multimedia*. ACM. 2007, pp. 357–360.

[20] Muhammad Muneeb Ullah, Sobhan Naderi Parizi, and Ivan Laptev. "Improving bag-of-features action recognition with non-local cues." In: *BMVC*. Vol. 10. Citeseer. 2010, pp. 95–1.

[21] Karla D Wagner et al. "Evaluation of an overdose prevention and response training programme for injection drug users in the Skid Row area of Los Angeles, CA". In: *International Journal of Drug Policy* 21.3 (2010), pp. 186–193.

[22] Heng Wang et al. "Evaluation of local spatio-temporal features for action recognition". In: *BMVC 2009-British Machine Vision Conference*. BMVA Press. 2009, pp. 124–1.

[23] Jun Yang et al. "Evaluating Bag-of-visual-words Representations in Scene Classification". In: *Proceedings of the International Workshop on Multimedia Information Retrieval*. MIR '07. Augsburg, Bavaria, Germany: ACM, 2007, pp. 197–206. ISBN: 978-1-59593-778-0. DOI: 10.1145/1290082.1290111. URL: http://doi.acm.org/10.1145/1290082.1290111.

[24] Liu Yang et al. "Unifying discriminative visual codebook generation with classifier training for object category recognition". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.