



**The author(s) shown below used Federal funding provided by the U.S. Department of Justice to prepare the following resource:**

**Document Title:** Studying the Impact of Video Analytics for Pre, Live and Post Event Analysis on Outcomes of Criminal Justice

**Author(s):** Mubarak Shah, Ph.D.

**Document Number:** 252266

**Date Received:** October 2018

**Award Number:** 2015-R2-CX-K025

**This resource has not been published by the U.S. Department of Justice. This resource is being made publically available through the Office of Justice Programs' National Criminal Justice Reference Service.**

**Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.**

# Project Final Report

**Award Number:** 2015-R2-CX-K025  
**Project Title:** Studying the Impact of Video Analytics for Pre, Live and Post Event Analysis on Outcomes of Criminal Justice  
**Reporting Period:** January 2016 – June 2018

**Contact Information (Organization):**

Mathew Cronan  
Sr. Proposal Manager  
Engineering II, Suite 202,  
University of Central Florida,  
Orlando, FL 32826  
Phone: (407) 823-3031  
Email: [matthew.cronan@ucf.edu](mailto:matthew.cronan@ucf.edu)

**Contact Information (Principal Investigator):**

Dr. Mubarak Shah  
Director, Center for Research in Computer Vision  
Suite 245D  
4328 Scorpius St.  
University of Central Florida  
Orlando, FL 32816-2365  
Phone: (407) 823-5077  
Email: [shah@crcv.ucf.edu](mailto:shah@crcv.ucf.edu)

# Accomplishments

This project aimed to develop and study the effect of computer analytics for public space surveillance camera systems. Videos from surveillance cameras have the ability to not only aid in post-event investigations but also to improve intervention in live criminal incidents by flagging them as they occur. However, when left unmonitored or poorly integrated into police departments, surveillance cameras often become useless. Currently, the number of surveillance cameras in the U.S. is increasing rapidly, with human monitoring capability unable to keep pace. In this project, we developed computer vision analytics for large surveillance camera networks and installed them into a Public Safety Visual Analytics Workstation (PSVAW) operating at the Orlando, Florida police department (OPD).

In this report, we cover the tasks in four areas performed during the two and a half years (January 2016 – June 2018). First, we have developed and tested algorithms at the University of Central Florida's Center for Research in Computer Vision (CRCV) and Columbia University's Digital Video and Multimedia (DVMM) Lab. Algorithms related to (1) action and event detection in videos, (2) video anomaly detection, (3) video summarization, (4) object and attribute retrieval, and (5) hashing for efficient information retrieval. Second, we developed a computer vision system GUI that incorporates a set of computer vision modules for the Public Safety Visual Analytics Workstation. Specifically, four computer vision modules were integrated into the system including anomaly detection, face attribute prediction, body attribute prediction, and action detection. Third, we have transferred equipment necessary purchased to employ the PSVAW to the Orlando Police Department. Fourth, we report the current status of the field placement collaboration in the Orlando Police Department.

The final report is comprised of five sections. Descriptions of the underlying logic for the developed computer vision algorithms and the results of their evaluations against standard computer vision science criteria are reported in section 1. The Public Safety Visual Analytics workstation components and GUI interface is described in section 2. Descriptions of the PSVAW related equipment transferred to the Orlando Police Department are provided in section 3. The field placement is discussed in section 4 and presentations and publications associated with the grant are listed in section 5.

## 1. Development and Implementation of Computer Vision Algorithms

### 1.1 Action and Event Detection in Videos

**Tube Convolutional Neural Network (T-CNN).** Detection of video events has two parts. First is the temporal localization of the event's start/ending time in long videos. Second is the localization of the person and behaviors by setting bounding boxes within video frames. We completed the first task of temporal event localization in video streams during the first half of Phase 1. During the recent six-month span, we developed an approach for the second task of spatial localization and recognition of events within videos utilizing an extension of computer vision deep learning.

Deep learning has been shown to produce excellent results for image classification and object detection. In particular deep learning based approaches have shown superior performance on a standardized computer vision assessment employing the “ImageNet” challenge and recent impressive successes in use of faster Region-based Convolutional Neural Network (R-CNN) [1] for object detection. However, the impact of deep learning on video analysis for tasks such as action detection and recognition has been limited due to complexity of video data and the limited number of annotated training videos.

Previous deep learning based action detection approaches first detect potential frame-level action proposals associated with pre-existing popular proposal algorithms (e.g. selective search [2]) and labels (i.e., running or jumping) or using action labels assigned through a training process. The frame-level action proposals are next associated across frames to generate final action detections. In order to capture both spatial and temporal information of an action, two-stream networks (a spatial CNN and a motion CNN) have been typically used with the spatial and motion information analyzed separately.

Inspired by Faster R-CNN [1], we proposed Tube Convolutional Neural Network (T-CNN) for action detection by leveraging the descriptive power of 3D CNN. To better capture the spatiotemporal information of video, we exploited 3D CNN since it was able to capture motion characteristics in videos and showed promising results on video action recognition. In our approach, an input video was first divided into equal length clips. Then the clips were fed into Tube Proposal Network (TPN) and a set of tube proposals were obtained. Next, tube proposals from each video clip were linked according to their actionness scores and overlap between adjacent proposals to form a complete tube proposal for spatio-temporal action localization in the video. Finally, the Tube-of-Interest (ToI) pooling was applied to the linked action tube proposal to generate a fixed size feature vector for action label prediction.

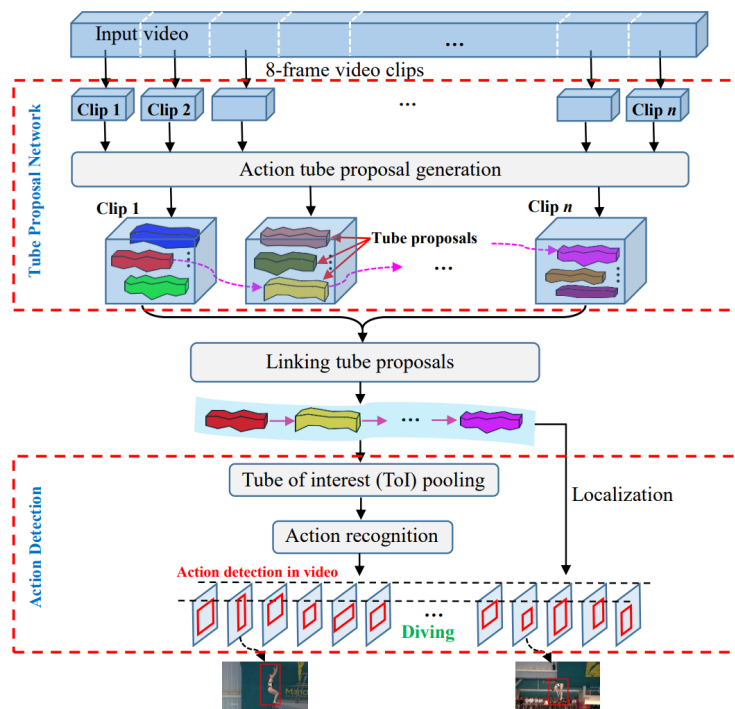


Figure 1. Pipeline and network structure of the T-CNN.

Figure 1 shows an overview of the proposed Tube Convolutional Neural Network (T-CNN) for action detection. First, an input video was divided into equal length clips of 8 frames and fed to Tube Proposal Network to generate tube proposals. Next, these proposals are then linked into larger tubes covering full actions and fed to Action Detection network. Finally, the Action Detection network employed TOI pooling to recognize and localize the action.

For the Tube Proposal Network, we first divided the video into 8-frame clips. For each 8-frame video clip, 3D convolution and 3D pooling were used to extract a spatio-temporal feature cube. In 3D CNN, convolution and pooling were performed spatio-temporally preserving the temporal information of the input video. Our 3D CNN consisted of seven 3D convolution layers and four 3D max-pooling layers. We denoted the kernel shape of 3D convolution/pooling by  $d \times h \times w$ , where  $d$ ,  $h$ ,  $w$  are respectively depth, height and width. In all convolution layers, the kernel sizes were  $3 \times 3 \times 3$ , padding and stride remained as 1. The numbers of filters were 64, 128 and 256 respectively in the first 3 convolution layers and 512 in the remaining convolution layers. The kernel size was set to  $1 \times 2 \times 2$  for the first 3D max-pooling layer, and  $2 \times 2 \times 2$  for the remaining 3D max-pooling layers.

Each bounding box (bbx) is associated with an “actionness” score, which measured the probability that the bbx corresponded to a valid action. We assigned a binary class label (of being an action or not) to each bounding box. Bounding boxes with actionness scores smaller than a selected threshold were discarded. In the training phase, the bbx which had an Intersection-over-Union (IoU) overlap higher than 0.7 with any ground-truth bbx, or had the highest IoU overlap with a ground-truth box (the later condition was considered in case the former condition found no positive cases) was taken as a positive bounding box proposal.

Bounding box proposals generated from conv5 feature tube can be used for frame-level action detection by bounding box regression. However, due to temporal max pooling (8 frames to 1 frame), the temporal order of the original 8 frames is lost. Therefore, we used temporal skip pooling to inject the temporal order for frame-level detection. Specifically, we mapped each positive bounding box generated from conv5 feature cube into conv2 feature cube which had 8 feature frames/slices. Since these 8 feature slices corresponded to the original 8 frames in a video clip, the temporal order was preserved. As a result, if there were 5 bounding boxes in conv5 feature cube for example, 5 scaled bounding boxes were mapped to each conv2 feature slice at the corresponding locations. This created 5 tube proposals as illustrated in Figure 2, which were paired with the corresponding 5 bounding box proposals for frame-level action detection. To form a fixed feature maps, ToI pooling was applied to the variable size tube proposals as well as the bounding box proposals. Since a tube proposal covers 8 frames in Conv2, the ToI pooled bounding box from Conv5 was duplicated 8 times to form a tube. We then applied L2 normalization to the paired two tubes, and vectorized and concatenated them. Since we used the C3D model as the pretrained model, we applied a  $1 \times 1$  convolution to match the input dimension of fc6. Three fully-connected (FC) layers process each descriptor and produce the output: displacement of height, width and 2D center of each bounding box (“bbx”) in each frame. The regression loss measured the differences between ground truth and predicted bounding boxes, represented by a 4D vector ( $\Delta$ center-x,  $\Delta$ center-y,  $\Delta$ width,  $\Delta$ height). The sum of them for all bounding boxes was the regression loss of the whole tube. Finally, a set of refined tube proposals by adding the displacements of height, width and center were generated as an output from the TPN representing potential spatio-temporal action localization of the input video clip. Figure 2 illustrates the proposed Tube Proposal Network.

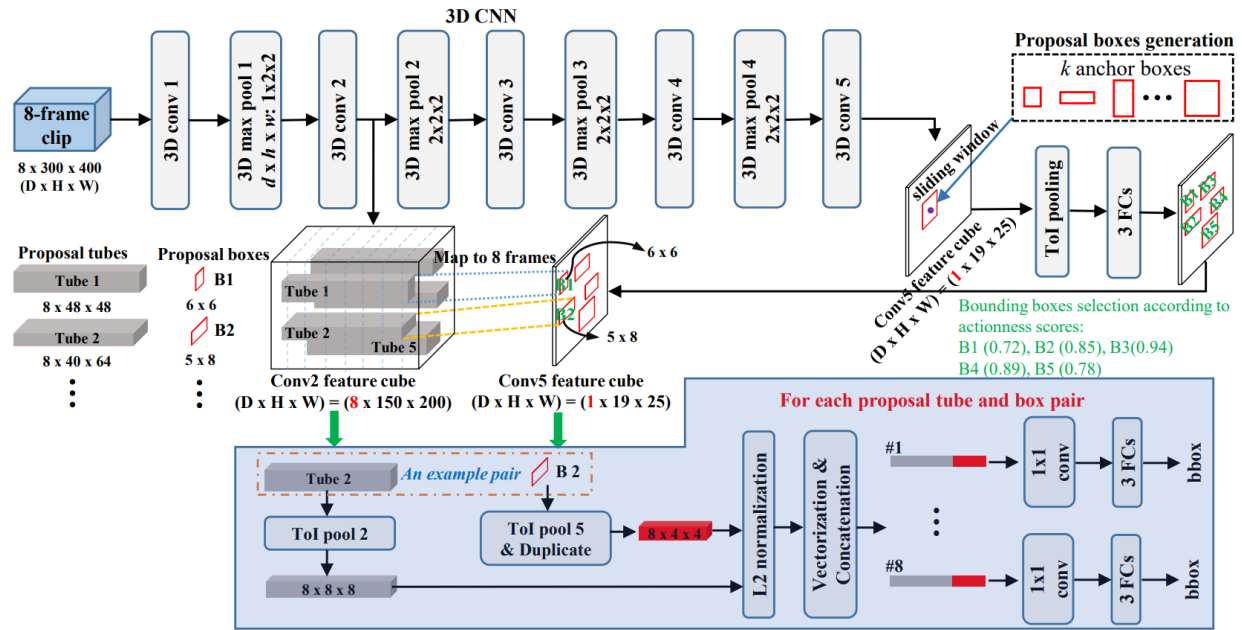


Figure 2. The tube proposal network (TPN) takes a 8-frame clip as input and applies 3D convolution and max-pooling to extract spatio-temporal features. Conv5 feature cube are used to generate bounding box proposals. Those with high actionness scores are mapped to conv2 feature cube (contains 8 frames information) at the corresponding positions to get tube proposals. Each proposal tube and box pair are aggregated after separate ToI pooling, then bounding box regression is performed for each frame.

**Segmentation Tube CNN (ST-CNN).** The proposed T-CNN approach was able to detect actions in videos and place bounding boxes for localization in each frame. As discussed, T-CNN as a top-down approach relies on exhaustive search in the whole frame and appropriate bounding boxes selection. It has been shown that bottom-up approaches which directly operate on group of pixels e.g. through supervoxel or super pixel segmentation are more efficient for action detection. Also, it is obvious that pixel-wise action segmentation maps provide finer human silhouettes than bounding boxes, since bounding may also include background pixels. To achieve this goal, we developed a ST-CNN (Segmentation Tube CNN) approach to automatically localize and segment the silhouette of an actor for action detection. Figure 3 shows the network structure of the proposed ST-CNN. It is an end-to-end 3D CNN, which builds upon an encoder-decoder structure for image semantic segmentation. Its development starts with a video being divided into 8-frame clips as input to the network. On the encoder side, 3D convolution and max pooling were performed. Due to 3D max pooling, the spatial and temporal sizes were reduced. In order to generate the pixel-wise segmentation map for each frame in the original size, 3D up-sampling was used in the decoder to increase the resolution of feature maps. To capture spatial and temporal information at different scales, a concatenation with the corresponding feature maps from the encoder was employed after each 3D up-sampling layer. Finally, a segmentation branch was used for pixel-wise prediction (i.e. background or action foreground) for each frame in a clip. The recognition branch takes the segmentation maps (output of the segmentation branch), where the foreground segmentation maps (action regions) were converted into bounding boxes, and the feature cube of the last concatenation

layer (concat1), to extract the feature tube of the action volume. ToI pooling was applied to the feature tube and followed by three FC layers for action recognition. The Network architecture of ST-CNN is shown in Figure 3.

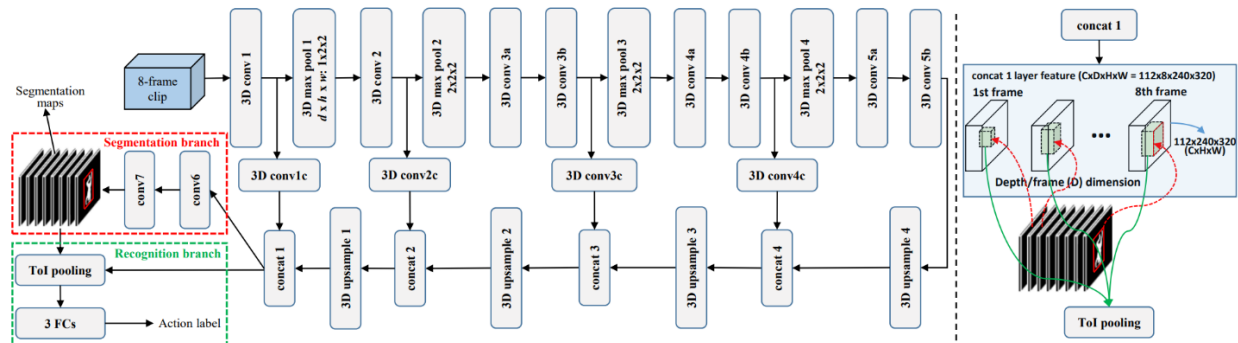


Figure 3. The framework of ST-CNN for action detection. An encoder-decoder 3D CNN architecture is employed. The segmentation branch produces a binary segmentation map (action foreground vs. background) for each frame of the input video clip. The inferred framewise bounding boxes (action localization) are based on the foreground pixels. The feature tube is extracted from concat1 as an input to the recognition branch (see the right part of figure 3 for details).

**Experiment results.** We tested our proposed detection methods (T-CNN and ST-CNN) on computer vision science benchmark datasets.

**UCF-Sports** dataset contains 150 short videos of 10 different sport classes. Videos were trimmed and the action and bounding boxes annotations were provided for all frames. We followed the standard training and test split to carry out the evaluation. We used the usual IoU criterion and generated the ROC curves reported in Figure 4(a) when overlap equals to  $\alpha = 0.2$ . Figure 4(b) illustrates AUC (Area-Under-Curve) measured with different overlap criterion. In direct comparison, our T-CNN clearly outperformed all the competing methods shown. We were unable to directly compare the detection accuracy against Peng et al. in the plot, since they did not provide ROC and AUC curves. As shown in Table 1, the frame level mAP of our approach outperformed Peng et al. in 8 actions out of 10. Moreover, using the same metric, the video mAP of our approach reached 95.2 ( $\alpha = 0.2$  and 0.5), while they reported 94.8 ( $\alpha = 0.2$ ) and 94.7 ( $\alpha = 0.5$ ).

**J-HMDB** consisted of 928 videos with 21 different actions. All the video clips were well trimmed. There were three train-test splits and the evaluation was done on the average results over the three splits. The comparison of the experimental results is shown in Table 2. We report our results using 3 metrics: frame-mAP, the average precision of detection at frame level as in [1=3]; video-mAP, the average precision at video level as in [3] with IoU threshold  $\alpha = 0.2$  and  $\alpha = 0.5$ . Our T-CNN algorithm consistently outperformed the state-of-the-art approaches for all three evaluation metrics.

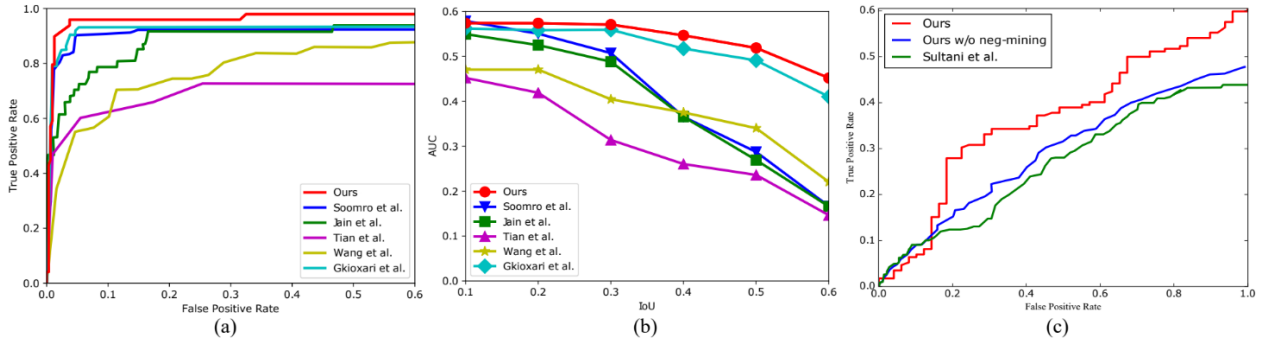


Figure 4. The ROC and AUC curves for UCF-Sports dataset are shown in (a) and (b), respectively. The results are shown for Jain et al. [38] (green), Tian et al. (purple), Soomro et al. (blue), Wang et al. (yellow), Gkioxari et al. (cyan) and Proposed Method (red). (c) shows the mean ROC curves for four actions of THUMOS’14. The results are shown for Sultani et al. (green), the proposed T-CNN (red) and T-CNN without negative mining (blue).

Table 1. mAP for each class of UCF-Sports. The IoU threshold  $\alpha$  for frame m-AP is fixed to 0.5.

	Diving	Golf	Kicking	Lifting	Riding	Run	SkateB.	Swing	SwingB.	Walk	mAP
Gkioxari <i>et al.</i> [1]	75.8	69.3	54.6	99.1	89.6	54.9	29.8	88.7	74.5	44.7	68.1
Weinzaepfel <i>et al.</i> [2]	60.71	77.55	65.26	<b>100.00</b>	99.53	52.60	47.14	88.88	62.86	64.44	71.9
Peng <i>et al.</i> [3]	<b>96.12</b>	80.47	73.78	99.17	97.56	82.37	57.43	83.64	98.54	75.99	84.51
Kalogeiton <i>et al.</i> [43]	–	–	–	–	–	–	–	–	–	–	<b>87.7</b>
Ours (T-CNN)	84.38	<b>90.79</b>	86.48	99.77	<b>100.00</b>	<b>83.65</b>	<b>68.72</b>	65.75	<b>99.62</b>	<b>87.79</b>	86.7
Ours (ST-CNN*)	70.9	89.1	<b>90.7</b>	89.7	99.6	71.1	80.4	<b>89.3</b>	86.7	77.5	84.5

Table 2. Comparison of the state-of-the-art approaches on J-HMDB. The IoU threshold  $\alpha$  for frame m-AP is fixed to 0.5.

	f.-mAP ( $\alpha = 0.5$ )	v.-mAP ( $\alpha = 0.2$ )	v.-mAP ( $\alpha = 0.5$ )
Gkioxari <i>et al.</i> [1]	36.2	–	53.3
Weinzaepfel <i>et al.</i> [2]	45.8	63.1	60.7
Peng <i>et al.</i> [3]	58.5	74.3	73.1
Kalogeiton <i>et al.</i> [43]	<b>65.7</b>	74.2	73.7
Singh <i>et al.</i> [16]	–	73.8	72.0
Ours (T-CNN)	61.3	78.4	76.9
Ours (ST-CNN)	64.9	<b>78.6</b>	<b>78.3</b>
Ours (un-pool)	57.1	71.6	73.9



The **UCF-101** dataset with 101 actions is commonly used for action recognition and a subset (THUMOS’13) of 24 action classes and 3, 207 videos which have spatio-temporal annotations were utilized. Similar to other methods, we performed the experiments on the first train/test split. We report our results in Table 3 with 3 metrics: frame-mAP, video-mAP ( $\alpha = 0.2$ ) and videomAP ( $\alpha = 0.5$ ). Our approach again yielded the best performance. Moreover, we also report the action recognition results of T-CNN on the above three datasets in Table 4.

Table 3. Comparison of the state-of-the-art on UCF-101 (24 actions). The IoU threshold  $\alpha$  for frame m-AP is fixed to 0.5.

IoU th.	f.-mAP	video-mAP			
		0.05	0.1	0.2	0.3
Weinzaepfel <i>et al.</i> [2]	35.84	54.3	51.7	46.8	37.8
Peng <i>et al.</i> [3]	65.73	<b>78.76</b>	77.31	72.86	65.7
Kalogeiton <i>et al.</i> [43]	67.1	–	–	<b>77.2</b>	–
Singh <i>et al.</i> [16]	–	–	–	73.5	–
Ours	<b>67.3</b>	78.2	<b>77.9</b>	73.1	<b>69.4</b>

Table 4. Action recognition results of T-CNN on three datasets.

	Accuracy (%)
UCF-Sports	95.7
J-HMDB	67.2
UCF-101 (24 actions)	94.4

To further validate the effectiveness of our T-CNN approach for action detection, we evaluated it using the untrimmed videos from the **THUMOS’14** dataset. The THUMOS’14 spatio-temporal localization task consisted of 4 classes of actions: baseball pitch, golf swing, tennis swing and discus throw. There were about 20 videos per action and each video contained 500 to 3,000 frames. The videos were divided into a validation set and a test set, but only videos in the test set have spatial annotations provided. Therefore, we used samples corresponding to those 4 actions in UCF-101 with spatial annotations to train our model. The mean ROC curves of different methods for THUMOS’14 action detection are plotted in Figure 4(c). Our method without negative mining performed better than the baseline method used by Sultani et al. Additionally, with negative mining, the performance was further improved. As a demonstration of our results, we show examples of detected action tubes in videos from UCF-Sports, JHMDB, UCF101 (24 actions) and THUMOS’14 datasets in Figure 5. Each block corresponds to a different video that was selected

from the test set. Figure 5 shows the highest scoring action tube for each video.

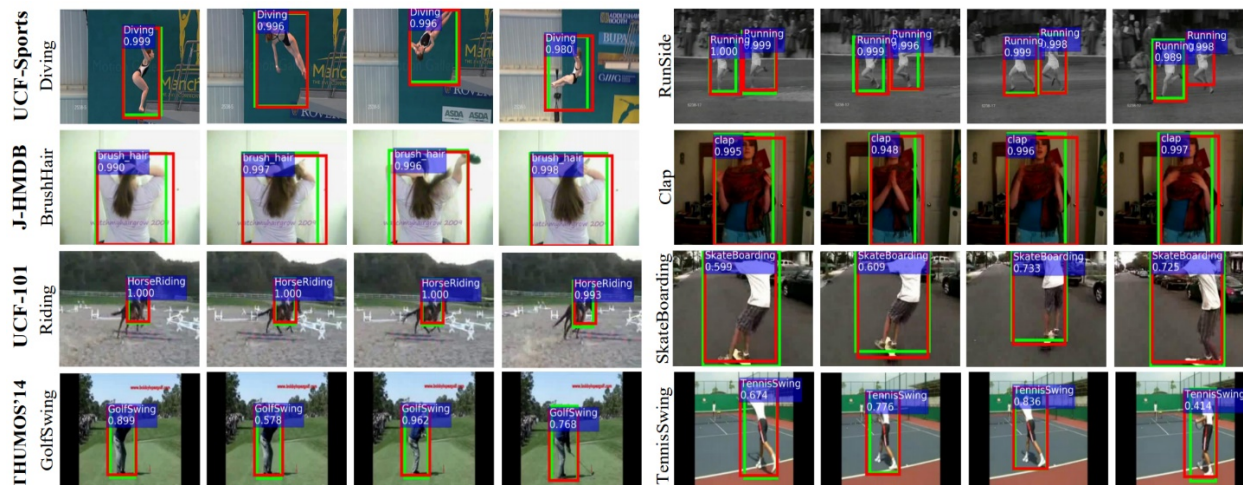


Figure 5. Action detection results obtained by T-CNN on UCF-Sports, JHMDB, UCF-101 and THUMOS'14. Red boxes show the detections in the corresponding frames, and green boxes show ground truth. The predicted labels are overlaid.

**Video object segmentation experiment.** We further evaluated our ST-CNN approach on the video object segmentation task. Densely Annotated Video Segmentation (DAVIS) 2016 dataset was specifically designed for the task of video object segmentation. It consisted of 50 videos with 3455 annotated frames. Consistent with most prior work, we conducted experiments on the 480p videos with a resolution of  $854 \times 480$  pixels. Thirty videos were taken for training and 20 for validation.

We adopted the same evaluation setting as reported in [4]. There were three parts. Region Similarity J, which was obtained by IoU between the prediction and the ground-truth segmentation map. Contour Accuracy F measured the contours accuracy performance. Temporal Stability T tracked the temporal consistency in a video. For the first two evaluation, we reported the mean, recall and decay. For the third one, we reported the average. We compared our results with several unsupervised implementations, since our approach did not require any manual annotation or prior information about the object to be segmented (defined as unsupervised segmentation). This approach was different from the semi-supervised approaches which assumed the ground truth segmentation map of the first frame of a test video as given. Unsupervised segmentation apparently is a much harder task, but is more practical since it does not require human labelling during testing once the segmentation model has been trained. We compared our method with the state-of-the-art unsupervised approaches in Table 5. According to the results, our method achieved the best performance in all performance metrics. Compared to ARP, the previous state-of-the-art unsupervised approach, our method achieved 5% gain in contour accuracy (F) and a 15% gain in temporal stability (T), demonstrating that 3D CNN can effectively take advantage of the temporal information in video frames to achieve temporal segmentation consistency. Figure 6 shows the quantitative results per video sequence of our approach and the next three top performing methods on the DAVIS dataset: ARP, LVO and FSEG. Our approach performed the best on low contrast

videos including black-swan, car-roundabout and scooter-black and achieved competitive results on other videos. Figure 7 presents the qualitative results on four video sequences. In the first row, our results were the most accurate of the four. Our method was the only one which detected the wheel rims of the bike. In the second row, ARP performed the best in suppressing background. However, only our approach detected both legs of the break dancer. The third row shows that only our method was able to accurately segment the tail of the camel. The last row was a challenging video because of the smoke and small initial size of the car. ARP missed part of the car, while LVO and FSEG mis-classified part of the background as a moving object. However, our method segmented out the car completely and accurately from the background smoke in the scene.

Table 5. Overall results of region similarity ( $\mathcal{J}$ ), contour accuracy ( $\mathcal{F}$ ) and temporal stability ( $\mathcal{T}$ ) for different approaches.  $\uparrow$  means higher values better performance, and  $\downarrow$  means lower values equals better performance on each method.

Measure		ARP [61]	FSEG [45]	LMP [46]	FST [62]	CUT [63]	NLC [64]	MSG [65]	KEY [66]	CVOS [67]	TRC [68]	SAL [69]	Ours
$\mathcal{J}$	Mean $\uparrow$	76.2	70.7	70.0	55.8	55.2	55.1	53.3	49.8	48.2	47.3	39.3	<b>77.6</b>
	Recall $\uparrow$	91.1	83.5	85.0	64.9	57.5	55.8	61.6	59.1	54.0	49.3	30.0	<b>95.2</b>
	Decay $\downarrow$	7.0	1.5	1.3	<b>0.0</b>	2.2	12.6	2.4	14.1	10.5	8.3	6.9	2.3
$\mathcal{F}$	Mean $\uparrow$	70.6	65.3	65.9	51.1	55.2	52.3	50.8	42.7	44.7	44.1	34.4	<b>75.5</b>
	Recall $\uparrow$	83.5	73.8	79.2	51.6	61.0	51.9	60.0	37.5	52.6	43.6	15.4	<b>94.7</b>
	Decay $\downarrow$	7.9	<b>1.8</b>	2.5	2.9	3.4	11.4	5.1	10.6	11.7	12.9	4.3	4.9
$\mathcal{T}$	Mean $\downarrow$	39.3	32.8	57.2	36.6	27.7	42.5	30.1	26.9	25.0	39.1	66.1	<b>22.0</b>

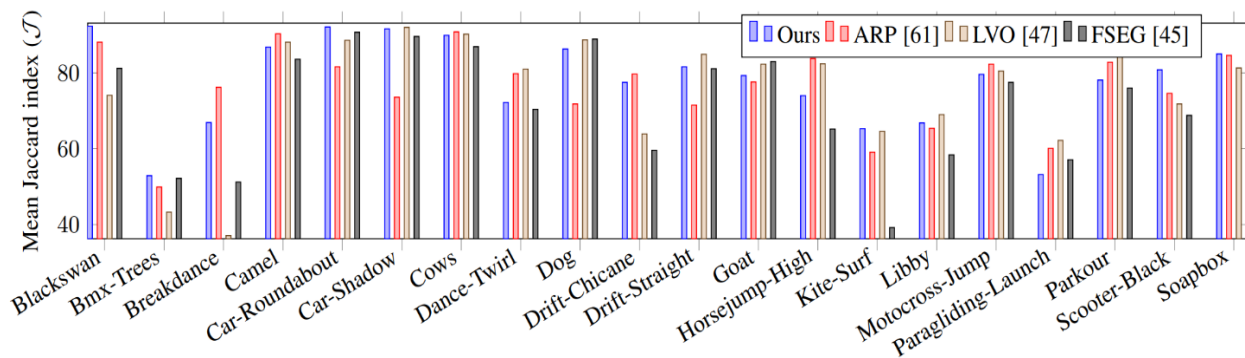


Figure 6. Comparison of Mean Jaccard index ( $\mathcal{J}$ ) of different approaches on each of the sequences independently.



Figure 7. Qualitative results of the proposed approach (red), ARP (yellow), LVO (cyan) and FSEG (magenta) on selected frames from DAVIS dataset.

## 1.2 Temporal Action Localization with Convolutional-Deconvolutional (CDC) Networks

An action / event usually consists of a sequence of sub-actions / sub-events in a specific order. Localizing actions in long untrimmed videos has recently drawn considerable interest. The goal is to detect an action class, and its starting and end time, for an untrimmed input video. It is particularly important in video surveillance analytics. In an online surveillance monitoring setting, temporal action localization can detect the start of an action of interest, for example a crime such as robbery, and stamp the time when the action/event is complete. In an offline setting, temporal action localization can be applied to recorded long untrimmed surveillance videos in a database. It allows a law enforcement agent to automatically pinpoint the time duration when an event of interest such as a crime happened in a surveillance video.

Recent temporal action localization approaches were based on segment proposal, i.e., proposing a set of intervals and then classifying the content of each interval. However, determining a good interval before full understanding of its content was to some degree difficult. We proposed a method to determine the action start time and end time after fine-grained analysis of the content and determination of which action each frame contained. Per-frame labeling of actions in videos can be conventionally done by feeding individual frames into Convolutional Neural Networks (CNN) for classification. A recurrent layer can be added on top of such an architecture to utilize temporal dependency. However, none of these methods can model motion explicitly. It has been shown in C3D [5] that 3D CNNs can successfully model spatiotemporal patterns such as motion in videos.

**CDC Model Architecture.** In the following, we represented a convolutional feature map

as a 4-dimensional tensor with the shape  $(a, b, c, d)$ , where  $a$ ,  $b$ ,  $c$ , and  $d$  respectively represented the number of channels, temporal length, spatial height and spatial width. C3D takes a video clip of dimensions  $(3, 16, 112, 112)$  as input (RGB channels, 16 frames of  $112 \times 112$  pixels), encoded it gradually to  $(512, 2, 4, 4)$  at the end of the pool5 layer, and then by means of fully connected layers, encoded it to  $(C, 1, 1, 1)$ , which was a vector of classification scores per class. In both sections of the model (convolutional and fully connected parts), down sampling was done in time and space. However, to enable per-frame labeling, we needed to up sample in time to reach  $(C, 16, 1, 1)$ , that is, the classification score vectors for each frame. To this end, we replaced the fully connected portion of the network with a new set of layers, to which we referred to as Convolutional-De-Convolutional layers.

Our proposed network encoded the  $(3, 16, 112, 112)$  input into  $(512, 2, 4, 4)$  by a sequence of layers identical to C3D, but then gradually up sampled in time while keeping down sampling in space until it reached  $(C, 16, 1, 1)$ . The up sampling was done in literature both by linear up sampling (non-trainable), and deconvolutional layers (trainable). In our case however, we needed to down sample in space (from  $4 \times 4$  to  $1 \times 1$ ) while up sampling in time. To do this, we proposed Convolutional-De-Convolutional layers, which behaved like a convolution in the spatial dimensions, and a deconvolution in time.

Since the CDC layer was not fully connected, the network was categorized as a fully convolutional network. In such network architectures, the input tensor shape is not fixed, and the output shape was determined based on the input shape. This means the window size did not necessarily have to be 16 frames and the model could operate on long untrimmed videos. Hence, we applied the network on any input of shape  $(3, L, 112, 112)$ , where  $L$  is the number of frames, encode it through convolutional/pooling layers to  $(512, L/8, 4, 4)$ , and then transformed it using CDC layers to  $(C, L, 1, 1)$ . Figure 8 illustrates how a CDC layer works.

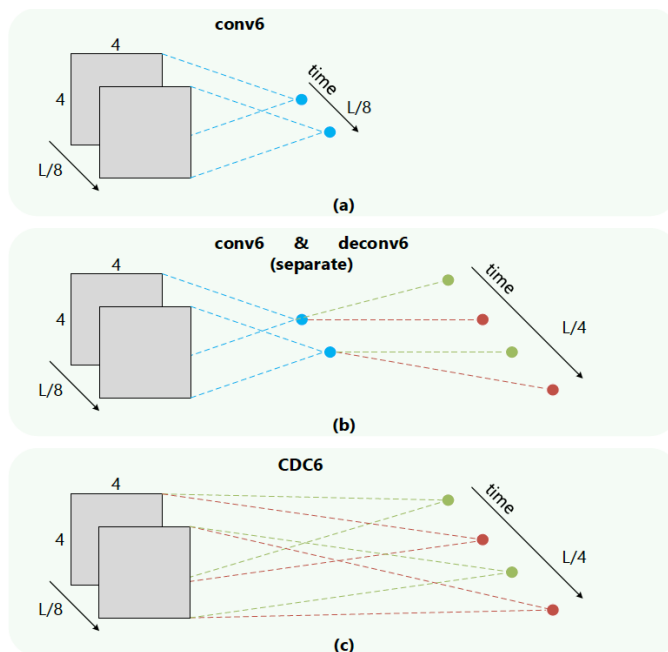


Figure 8. Illustration of receptive fields for (a) regular convolutional layer (b) convolution followed for deconvolution (c) the Convolutional-De-Convolutional layer.

In the proposed network, we used three CDC layers in order to perform the mentioned transformation. The detailed illustration of the proposed architecture is depicted in Figure 9. The specification of CDC layers was chosen so that pre-trained FC6 and FC7 layers of the original C3D could be reused.

We fine-tuned this network on the validation set of THUMOS'14 dataset. This dataset consisted of untrimmed videos annotated by intervals with specific start and end time and action class. We converted these annotations to per-frame labels, by considering segment-level action classes for all frames inside the segment and defining a background class for all other frames. After fine-tuning, the model learned to classify each frame. To detect action intervals, we started by applying action proposals with high recall, then applied per-frame labeling inside each proposed window and used that to modify the proposed boundaries. This still relied on a good action proposal method, which we intended to circumvent. However, this was what currently showed the best results. Thus, an algorithm that does not rely on action proposal will entail further research. Figure 10 explains our detection procedure in more detail.

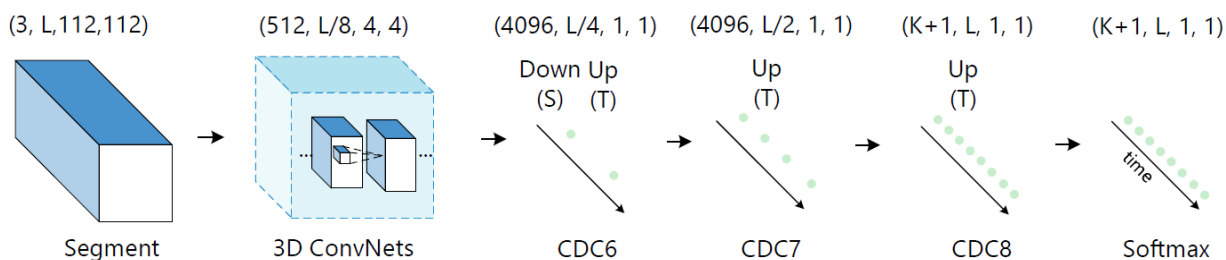


Figure 9. Proposed network architecture with three CDC layers.

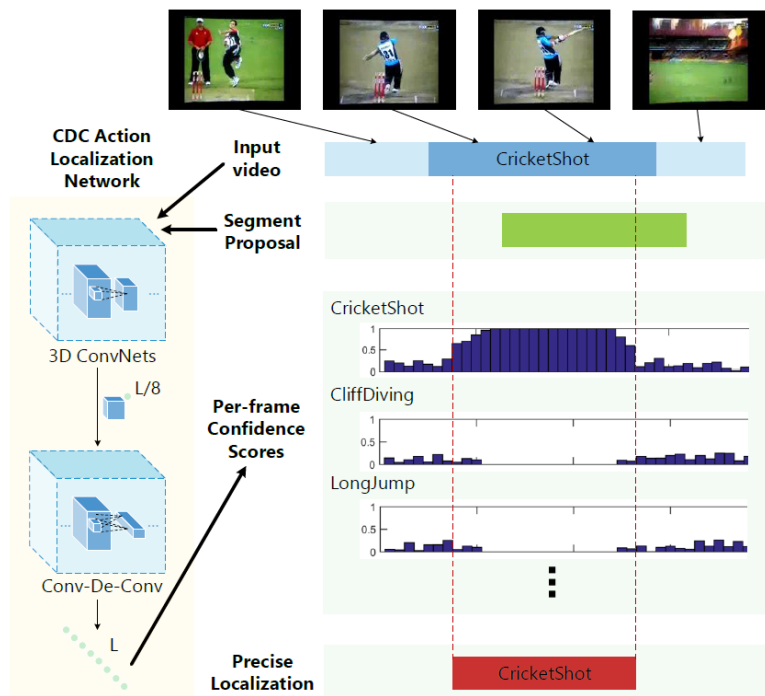


Figure 10. Proposed method of action interval detection.

**Evaluation results.** We evaluated the proposed method on the THUMOS’14 action detection challenge. The algorithm had to process untrimmed videos and provide a list of tuples of start time, end time, and action class. Each detected interval was considered correct if it reasonably overlapped with a ground truth interval and matched its action class. The overlap was considered enough, if the Intersection over Union (IOU) of the detected interval and true interval was more than a selected threshold. To prevent scoring of redundant detections, two detected intervals could not be matched to the same ground truth interval.

We compared our results on the THUMOS’14 challenge with some baselines as well as recent state-of-the-art methods. The results are summarized in Table 6, where CDC shows significant improvement over a variety of state-of-the-art works. An example of the model output is depicted in Figure 11. Our implementation of the CDC network on an average machine with an NVIDIA TITAN X GPU performs at 500 frames per second. This means it could concurrently process video streams captured by 20 cameras at a frame rate of 25 FPS in real-time.

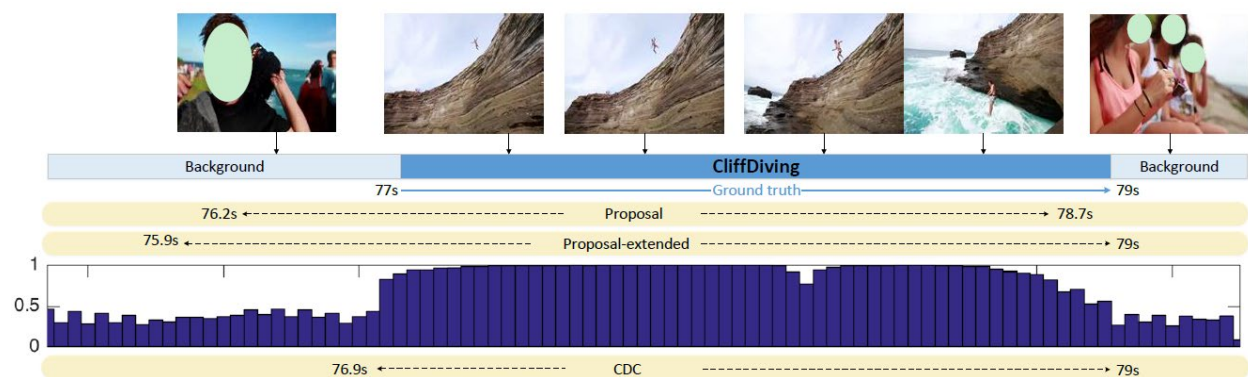


Figure 11. Visualization of an example input and output and the process of refining the proposed interval.

IoU threshold	0.3	0.4	0.5	0.6	0.7
Karaman <i>et al.</i> [26]	0.5	0.3	0.2	0.2	0.1
Wang <i>et al.</i> [66]	14.6	12.1	8.5	4.7	1.5
Heilbron <i>et al.</i> [18]	-	-	13.5	-	-
Escorcía <i>et al.</i> [9]	-	-	13.9	-	-
Oneata <i>et al.</i> [39]	28.8	21.8	15.0	8.5	3.2
Richard and Gall [43]	30.0	23.2	15.2	-	-
Yeung <i>et al.</i> [74]	36.0	26.4	17.1	-	-
Yuan <i>et al.</i> [77]	33.6	26.1	18.8	-	-
S-CNN [47]	36.3	28.7	19.0	10.3	5.3
C3D + LinearInterp	36.0	26.4	19.6	11.1	6.6
Conv & De-conv	38.6	28.2	22.4	12.0	7.5
CDC (fix 3D ConvNets)	36.9	26.2	20.4	11.3	6.8
<b>CDC</b>	<b>40.1</b>	<b>29.4</b>	<b>23.3</b>	<b>13.1</b>	<b>7.9</b>

Table 6. Mean Average Precision for different methods at different IoUs.

## 1.3 Weakly Supervised Anomaly Detection in Videos

Surveillance cameras are increasingly used in public places including streets, intersections, banks, shopping malls, and other locations to increase public safety. One critical task in video surveillance is detecting anomalous events such as traffic accidents, crimes or illegal activities. By definition, anomalous events rarely occur compared to normal activities and it is tedious and expensive for human operators to view entire videos in order to determine interesting but anomalous activities. Therefore, a key component of the PSVAW system is the automatic video anomaly detection.

As real-world anomalous events are rare, complicated, and diverse, it is impossible to list all possible anomalous events. Therefore, it is desirable that the anomaly detection algorithm not rely on prior information about events. Typically, sparse coding based approaches represent the current state-of-the-art anomaly detection methods. These methods assume that only the small initial portion of a video contains normal events, and therefore the initial video portion is used to build a “normal event” dictionary. The main assumption underlying anomaly detection is that anomalous events cannot be re-constructed from the normal event dictionary. Their uniqueness in comparison to the normal events identify them as anomalies.

Although such unsupervised approaches are appealing, they are based on the assumption that any pattern, which deviates from the learned normal patterns would be considered an anomaly. However, this assumption may not hold true because it is very difficult or impossible to accurately define the normal region of a video which takes all possible normal behavior patterns into account. More importantly, the boundary between normal and anomalous behavior is often ambiguous. Under realistic conditions such as time of day, the same behavior could be a normal under some conditions or anomalous behavior under different conditions. Therefore, employing training data of normal and anomalous events can make an anomaly detection system more accurate.

Along these lines, we developed an anomaly detection algorithm using weakly labeled training videos. That is, we only know the video-level labels, i.e., a video was normal with no anomaly or it contained an imbedded anomaly at an unknown temporal location within the video file. This was an effective approach because we could easily annotate a large number of videos by only assigning video-level labels. To formulate a weakly supervised learning approach, we resorted to multiple instance learning (MIL). Specifically, we learned to identify and locate anomalies through a deep MIL framework by treating normal and anomalous surveillance videos as conceptual bags and short segments of each video as instances in a bag. Based on training videos, our approach automatically learned an anomaly-ranking model. During testing, a long untrimmed video was divided into segments and fed to our deep network, which assigned an anomaly score for each video segment and was generalizable with the capacity to detect any anomalous event.

In our approach, we quantified anomaly detection as a regression problem. We wanted anomalous video segments to have higher anomaly scores than the normal segments. The most direct approach would be to use a ranking loss, which encourages high score for anomalous video segments as compared to normal segments. However, in the absence of video segment level annotations (videos were only weakly labeled as containing or not containing anomalies), this was not possible. Instead, the following multiple instance ranking objective function was employed:



$$l(B^+, B^-) = \max\left(0, 1 - \max_{i \in B^+} f(V_s^{i+}) + \max_{i \in B^-} f(V_s^{i-})\right)$$

where  $B^+$  and  $B^-$  represent positive and negative bags,  $V_s^+$  and  $V_s^-$  represent anomalous and normal video segment and  $f(V_s^+)$  and  $f(V_s^-)$  represent their predicted scores, respectively. In this MIL ranking loss, the error was back propagated from the maximum scored video segments in both positive and negative bags. By training on a large number of positive and negative bags, the computer vision software learned a generalized model to predict high scores for anomalous segments in positive bags. The framework of our approach is illustrated in Figure 12.

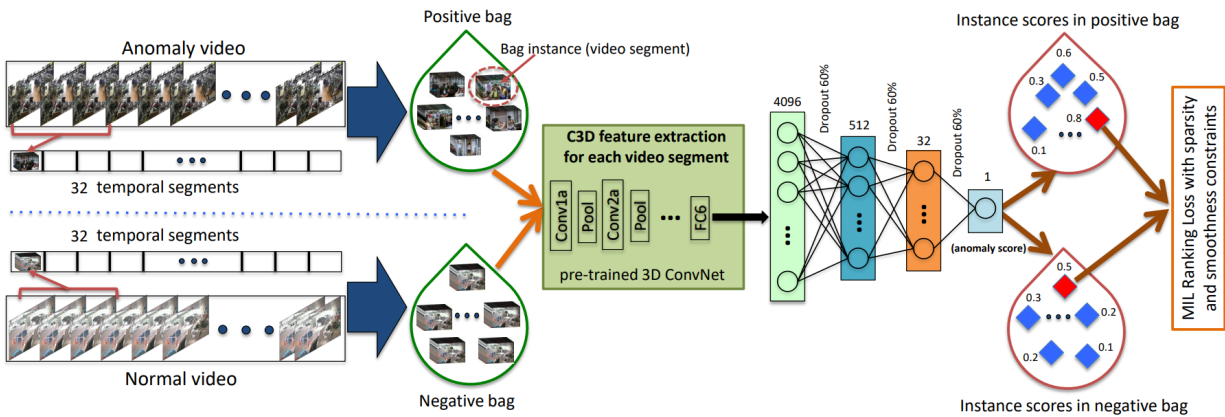


Figure 12. The flow diagram of the proposed anomaly detection approach. Given the positive (containing anomaly somewhere) and negative (containing no anomaly) videos, we divide each of them into multiple temporal video segments. Then, each video is represented as a bag and each temporal segment represents an instance in the bag. After extracting C3D features [36] for video segments, we train a fully connected neural network by utilizing a novel ranking loss function which computes the ranking loss between the highest scored instances (shown in red) in the positive bag and the negative bag.

Due to the limitations of previous datasets, we constructed a new large-scale dataset to evaluate our method. It consisted of long untrimmed surveillance videos which covered 13 real-world anomalies including Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies were selected because they have a significant impact on public safety. To ensure the quality of our dataset, we trained ten annotators (having different levels of computer vision expertise) to collect the dataset. We searched videos on YouTube and LiveLeak using text search queries (with slight variations e.g. “car crash”, “road accident”) of each anomaly. In order to retrieve as many videos as possible, we also used text queries in different languages (e.g. French, Russian, Chinese, etc.) for each anomaly. We removed videos which fell into any of the following conditions: manually edited, prank videos, not captured by CCTV cameras, taken from newscasts, captured using a hand-held camera, and containing compilations. We also discarded videos in which the anomaly was not

clear. With the above video pruning constraints, 950 unedited real-world surveillance videos with clear anomalies and 950 normal videos were collected, for a total of 1900 videos. In Figure 13, we show four frames of example video from each anomaly.

**Annotation.** For our anomaly detection method, only video-level labels were required for training. However, in order to evaluate its performance on testing videos, we needed to know the temporal annotations, i.e. the start and ending frames of the anomalous event in each testing anomalous video. To this end, we assigned the same videos to multiple annotators to label the temporal extent of each anomaly. The final temporal annotations were obtained by averaging annotations of different annotators. The complete dataset was finalized after intense efforts covering several months. Regarding the creation of training and testing sets, we divided our dataset into two parts: the training set consisted of 800 normal and 810 anomalous videos (details shown in Table 7) and the testing set included the remaining 150 normal and 140 anomalous videos. Both training and testing sets contained all 13 anomalies at various temporal locations in the videos with some of the videos having multiple anomalies.

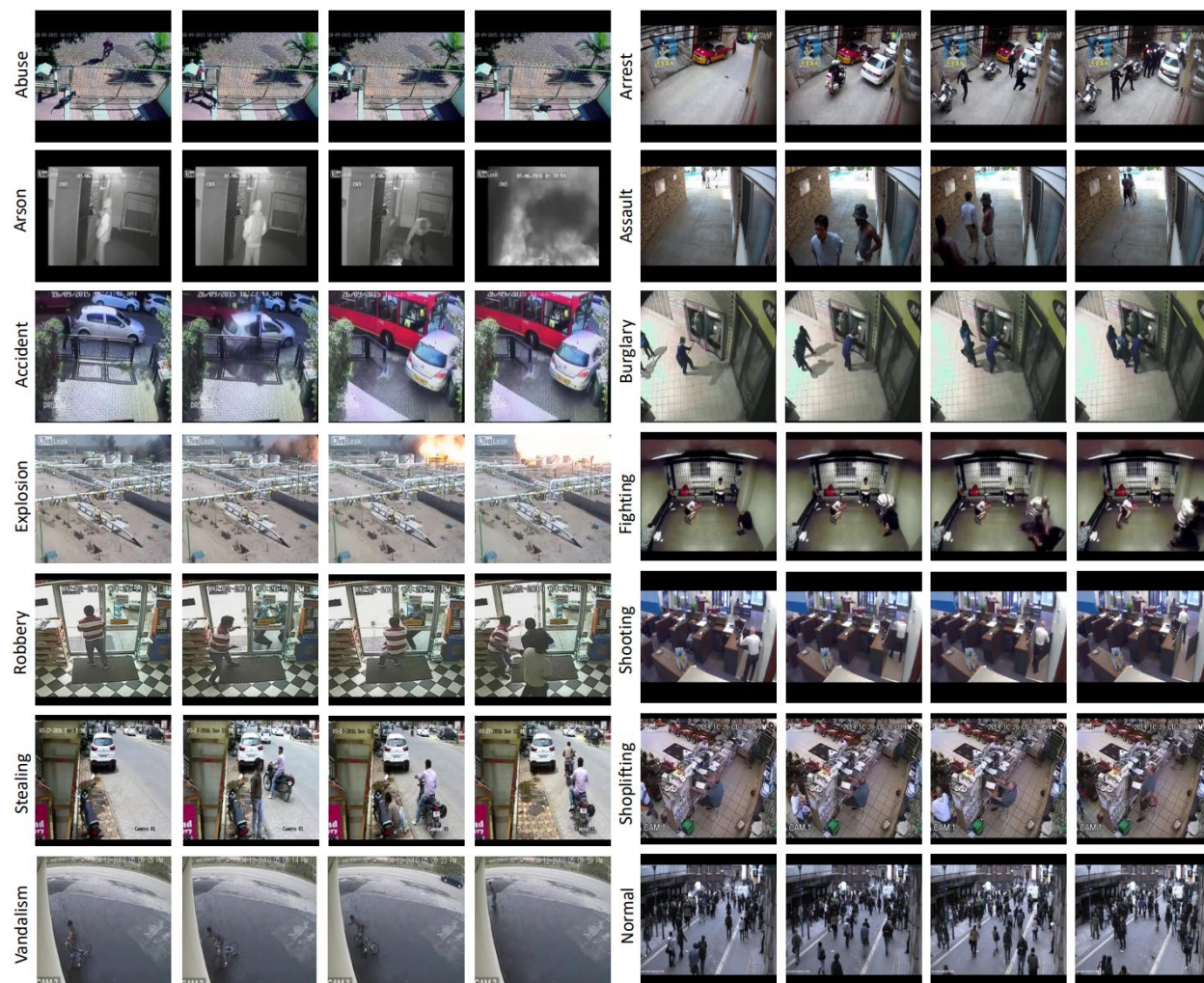


Figure 13. Examples of different anomalies from the training and testing videos in our dataset.

Table 7. Number of videos of each anomaly in our dataset. Numbers in brackets represent the number of videos in the training set.

# of videos	Anomaly
50 (48)	Abuse
50 (45)	Arrest
50 (41)	Arson
50 (47)	Assault
100 (87)	Burglary
50 (29)	Explosion
50 (45)	Fighting
150 (127)	Road Accidents
150 (145)	Robbery
50 (27)	Shooting
50 (29)	Shoplifting
100 (95)	Stealing
50 (45)	Vandalism
950 (800)	<b>Normal events</b>

**Experiment results.** We compared our method with two state-of-the-art approaches for anomaly detection. Lu et al. proposed dictionary-based approach to learn the normal behaviors and reconstruction errors to detect anomalies. Following their code, we extracted 7000 cuboids from each of the normal training video and compute gradient based features in each volume. After reducing the feature dimension using PCA, we generated the dictionary using sparse representation. Hasan et al. proposed a fully convolutional feedforward deep auto-encoder based approach to learn local features and classifier. Using their implementation, we trained the network on normal videos using a temporal window of 40 frames. Reconstruction error was used to measure anomaly. We also used a binary SVM classifier as a baseline method. Specifically, we treated all anomalous videos as one class and normal videos as another class. C3D features were computed for each video, and a binary classifier was trained with linear kernel. For testing, this classifier provided the probability of each video clip being anomalous. The quantitative comparisons in terms of ROC and AUC are shown in Figure 14 and Table 8.

We also compared the results of our approach with and without smoothness and sparsity constraints. The results showed that our approach significantly outperformed the existing methods. Particularly, our method achieved much higher true positive rates than other methods under low false positive rates e.g. 0.1-0.3. The binary classifier results demonstrated that traditional action

recognition approaches cannot be used for anomaly detection in real-world surveillance videos. This was because our dataset contained long untrimmed videos where an anomaly mostly occurs for a short period of time. Therefore, the features extracted from these untrimmed training videos were not discriminative enough for the anomalous events. In the experiments, binary classifier produced very low anomaly scores for almost all testing videos. Dictionary learning based method was not robust enough to discriminate between normal and anomalous pattern. In addition to producing the low reconstruction error for normal portion of the videos, it also produced low reconstruction error for anomalous part. Hasan et al. method learned normal patterns quite well. However, it tended to produce high anomaly scores for new normal patterns. Our method performed significantly better than Hasan et al. and demonstrated its effectiveness while it emphasizing that training using both anomalous and normal videos is indispensable for a robust anomaly detection system.

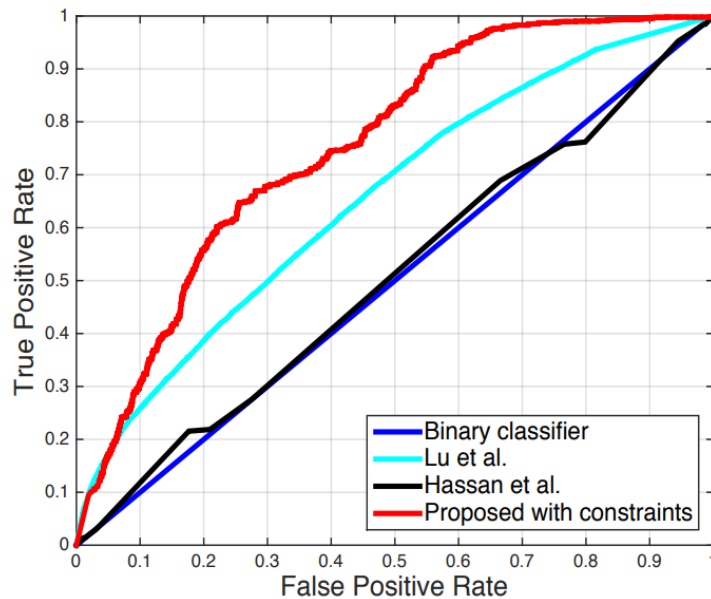


Figure 14. ROC comparison of binary classifier (blue), Lu et al. (cyan), Hasan et al. (black), proposed method without constraints (magenta) and with constraints (red).

Method	AUC
Binary classifier	50.0
Hasan <i>et al.</i> [18]	50.6
Lu <i>et al.</i> [28]	65.51
Proposed w/o constraints	74.44
<b>Proposed w constraints</b>	<b>75.41</b>

Table 8. AUC comparison of various approaches on our dataset.

In Figure 15, we present qualitative results of our approach on eight videos. (a)-(d) show four videos with anomalous events. Our method provides successful and timely detection of those anomalies by generating high anomaly scores for the anomalous frames. (e) and (f) are two normal videos. Our method produced low anomaly scores (close to 0) throughout the entire video, yielding zero false alarm for the two normal videos. We also illustrate two failure cases in (g) and (h). Specifically, (g) is an anomalous video containing a burglary event (person entering an office through a window). Our method failed to detect the anomalous part because of the darkness of the scene (a night video). Also, it generated false alarms mainly due to occlusions by flying insects in front of camera. In (h), our method produced false alarms due to people suddenly gathering to watch a street relay race, failing to identify the normal group activity.

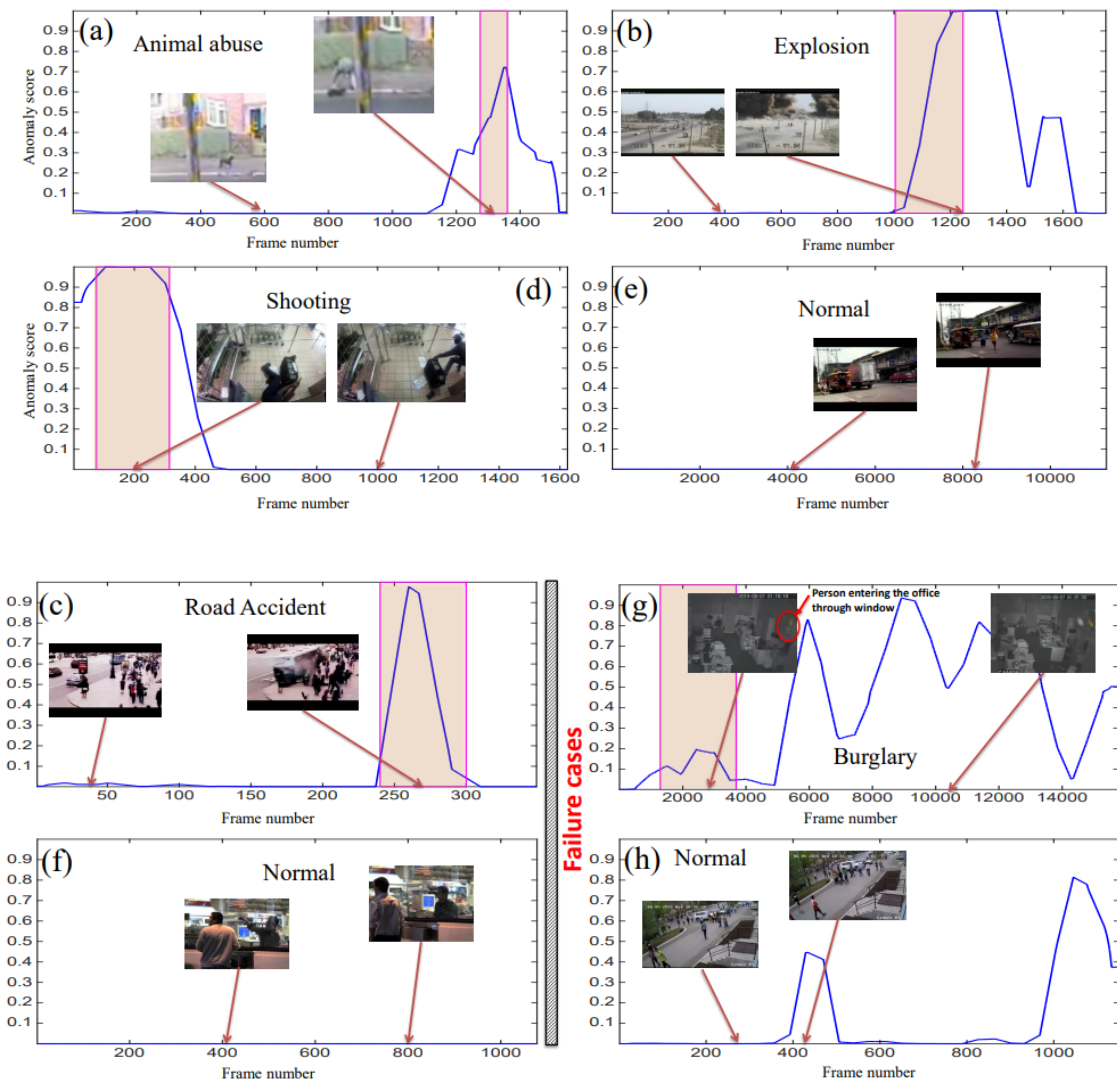


Figure 15. Qualitative results of our method on testing videos. Colored window shows ground truth anomalous region. (a), (b), (c) and (d) show videos containing animal abuse (beating a dog), explosion, road accident, and shooting, respectively. (e) and (f) show normal videos with no anomaly. (g) and (h) present two failure cases of our anomaly detection method.

Our dataset can be used as an anomalous activity recognition benchmark since we have event labels for the anomalous videos during data collection, but which were not used for our anomaly detection method discussed above. For activity recognition, we used 50 videos from each event and divided them into 75/25 ratio for training and testing. We provided two baseline results for activity recognition on our dataset based on a 4-fold cross validation. For the first baseline, we constructed a 4096-D feature vector by averaging C3D [36] features from each 16-frames clip followed by an L2-normalization. The feature vector was used as input to a nearest neighbor classifier. The second baseline was the Tube Convolutional Neural Network (TCNN) which introduced the tube of interest (ToI) pooling layer to replace the 5-th 3d-max-pooling layer in C3D pipeline. The ToI pooling layer aggregated features from all clips and output one feature vector for a whole video. Therefore, it was an end-to-end deep learning based video recognition approach. The quantitative results are given in Table 9. These state-of-the-art action recognition methods performed poorly on this dataset. It was because the videos were long untrimmed surveillance videos with low resolution. In addition, there were large intra-class variations due to changes in camera viewpoint and illumination, and background noise. Therefore, our dataset was a unique and challenging dataset for anomalous activity recognition.

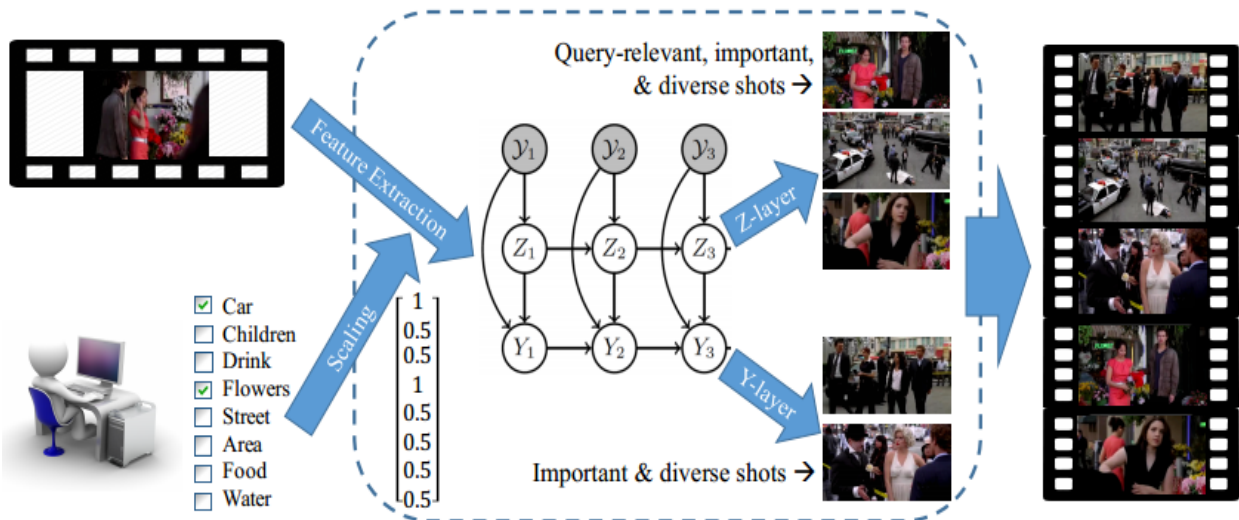
Table 9. Activity recognition results of C3D and TCNN.

Method	C3D [36]	TCNN [21]
Accuracy	23.0	28.4

## 1.4 Query-focused Extractive Video Summarization

We also developed a user-oriented video summarization approach which incorporated high-level supervised information. Rich Web images and videos provided (weak) priors for defining user-oriented importance of the visual content in a video. For instance, the car images on the Web revealed the canonical views of the cars, which should therefore be given special attention in video summarization. The advantages of leveraging high-level supervised information in video summarization over merely low-level cues is that the system developers were able to better infer the system users’ needs. It was more desirable to design a system based on user’s input such that the system’s sensitivities approached the users’.

In our query-focused video summarization, a query referred to one or more concepts (car, weapon) that were both user-nameable and machine-detectable. The decision to add a video shot to the output summary final depended on both the relevance between the shot and the query and the importance of the shot in the context of the video. To tackle this problem, we developed a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), and efficient learning and inference supportive algorithms. Our SH-DPP summarizer conveniently handled extremely long videos and online streaming videos. The logic of our approach is illustrated in Figure 16.



(a) Input: Video & Query (b) Algorithm: Sequential & Hierarchical Determinantal Point Process (SH-DPP) (c) Output: Summary

Figure 16. Query-focused video summarization and our approach to this problem.

We tested our query-focused video summarization approach on the UT Egocentric (UTE) dataset [6] and TV episodes [7]. The UTE dataset included four daily life ego centric videos, each 3–5 hours long, and the TV episodes contained four videos, each roughly 45 minutes long. These two datasets were very different in nature. The videos in UTE were long and recorded in an uncontrolled environment from the first-person view. As a result, many of the visual scenes were repetitive and unnecessary in a user summary. In contrast, the TV videos were episodes of TV series and from a third person’s viewpoint. The scenes were hence controlled and concise. A good summarizer should be able to function well in both scenarios. We evaluated our system generated video summary by contrasting it against the “ground truth” summary. The video summaries were mapped to text paragraphs and then compared by the ROUGE-SU metric.

Table 10 shows the results of different summarizers for the query focused video summarization when the patient and impatient users supply bi-concept queries. An immediate observation was that our SH-DPP was able to generate better overall summaries as our average F-scores were higher than the others’. Furthermore, our method was able to adapt itself to two essentially different datasets, the UTE daily life egocentric videos and TV episodes.

Table 10. Results of query-focused video summarization with bi-concept queries.

Patient Users	UTE (%)					TV episodes (%)				
	F	Prec.	Recall	HR	HR <sub>Z</sub>	F	Prec.	Recall	HR	HR <sub>Z</sub>
Sampling	<b>22.12</b>	<b>35.07</b>	17.11	23.61	n/a	27.99	34.75	24.36	16.00	n/a
Ranking	20.66	24.35	18.38	22.05	n/a	32.19	39.96	32.19	16.61	n/a
SubMod [28]	20.98	31.40	26.99	30.10	n/a	32.19	<b>41.59</b>	27.01	21.69	n/a
Quasi [46]	12.45	19.47	13.14	14.95	n/a	31.88	27.49	41.69	19.67	n/a
DPP [38]	15.7	19.22	32.08	30.94	n/a	29.62	35.26	34.00	21.29	n/a
seqDPP [27]	18.85	20.59	35.83	31.91	n/a	27.96	23.80	35.62	14.08	n/a
SH-DPP (ours)	21.27	17.87	<b>41.65</b>	<b>38.26</b>	<b>36.92</b>	<b>37.02</b>	38.41	<b>36.82</b>	<b>23.76</b>	20.35

Impatient Users	UTE (%)					TV episodes (%)				
	F	P	R	HR	HR <sub>Z</sub>	F	P	R	HR	HR <sub>Z</sub>
Sampling	25.44	44.16	18	6.48	n/a	33.74	<b>41.03</b>	28.8	13.03	n/a
Ranking	17.92	21.86	15.46	4.4	n/a	29.67	37.56	24.72	15.43	n/a
SubMod [28]	<b>27.10</b>	<b>51.79</b>	18.85	8.05	n/a	29.41	38.51	23.85	8.65	n/a
Quasi [46]	11.52	42.32	7.06	1.63	n/a	25.09	27.25	23.71	17.06	n/a
DPP [38]	14.36	30.9	16.18	12.54	n/a	26.01	28.85	39.15	<b>18.86</b>	n/a
seqDPP [27]	12.93	7.89	43.39	12.68	n/a	23.35	16.60	39.69	12.56	n/a
SH-DPP (ours)	25.56	18.51	<b>45.21</b>	<b>22.91</b>	11.57	<b>35.36</b>	30.94	<b>42.02</b>	17.07	17.07

## 1.5 Unsupervised Action Discovery and Localization in Videos

The problem of action recognition is to classify a video by assigning a label from a given set of annotated action classes (in comparison, action localization involves the detection of the spatio-temporal extent of a recognized action). Existing action recognition and localization approaches heavily rely on strong supervision in the form of training videos that have been manually collected, labeled and annotated. These approaches learn to detect an action using manually annotated bounding boxes and recognize action class labels from training data. Since supervised methods have the spatio-temporally annotated ground truth at their disposal, they can take advantage of learning detectors and classifiers by fine-tuning over the training data.

However, due to the difficulty of video annotation supervised algorithms have some disadvantages compared to unsupervised approaches. First, a video may consist of several actions in a complex cluttered background. Second, video level annotation in a supervised setting involves manually labeling the location (bounding box), the class of each action, and the temporal boundaries of each action, a time consuming set of tasks. Third, actions vary spatio-temporally (i.e. in height, width, spatial location and temporal length) resulting in various tubelet deformations. Fourth, different people may have a different understanding of the temporal extent



of an action, generating bias errors. Given the abundance of unlabeled videos available on the Internet, unsupervised learning approaches provided a promising alternative.

In this project, we developed an algorithm to automatically discover action classes by discriminatively clustering a group of unlabeled training videos. Our approach began by selecting a strongly coherent subset called a dominant set within each cluster, and trained a classifier for each action cluster to iteratively assign an action class to all videos. Next, using these action classes, we used a Knapsack approach to annotate actions in training videos. In this approach, we segmented the video into supervoxels and used a combinatorial optimization framework to select the supervoxels that belonged to the actor performing the action. Hence, we automatically obtained the ground truth, the action class labels and the actor bounding box annotations, for the training videos and generated an action classifier to perform Unsupervised Action Localization (see Figure 17).

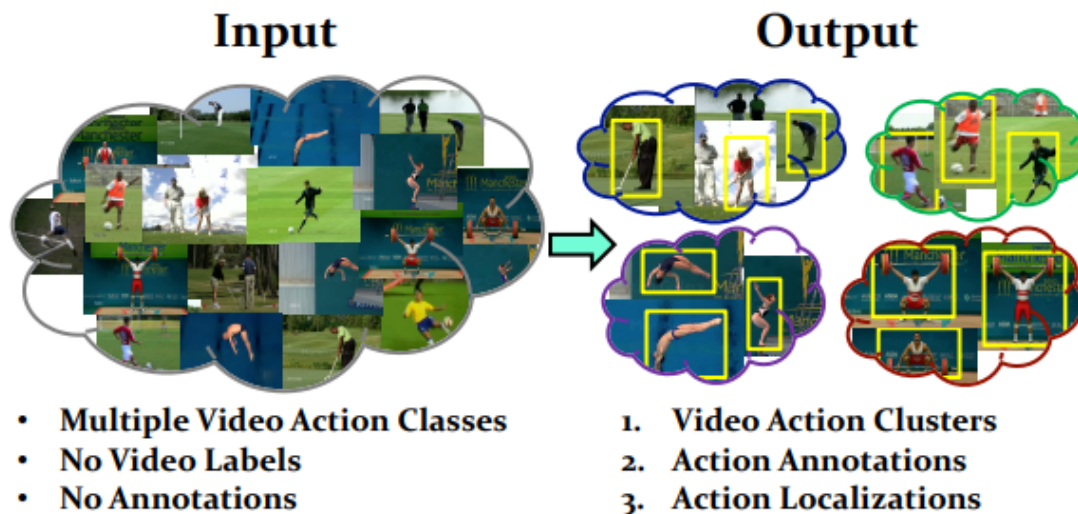


Figure 17. We tackled the problem of Unsupervised Action Localization without any action class labels or bounding box annotations, where a given collection of unlabeled videos contain multiple action classes. First, the proposed method discovers action classes by discriminative clustering using dominant sets (e.g. green and purple contours show clusters for kicking and diving actions, respectively) and then applies a variant of knapsack problem to determine spatio-temporal annotations of discovered actions (yellow bounding boxes). These annotations and action classes are used together to train an action classifier and perform Unsupervised Action Localization.

In our proposed approach, we first aimed to discover action classes from a set of unlabeled videos. We started by computing local feature similarity between videos to apply spectral clustering. Then, within each cluster, we constructed an undirected graph to extract a dominant set. This subset was used to train a Support Vector Machine (SVM) classifier within each cluster and discriminatively select videos from the non-dominant set to assign to one of the clusters in an iterative manner.

Given discovered action classes from our discriminative clustering approach, our aim was to annotate the action within each training video in every cluster. We began by over-segmenting a video into supervoxels, where every supervoxel either belonged to the foreground action or the background. Our goal was to select a group of supervoxels that collectively represented an action. We achieved this goal by solving the 0-1 Knapsack problem: Given a set of items (supervoxels), each with a weight (volume of a supervoxel) and a value (score of a supervoxel belonging to an action), determine the subset of items to include in a collection, so that the total weight was less than a given limit and total value was as high as possible. This combinatorial optimization problem would select supervoxels in a video based on their individual scores, resulting in a degenerate solution, where selected supervoxels were not spatio-temporally connected throughout the video. Therefore, we proposed a variant of the knapsack problem with temporal constraints that enforced the annotated action to be well-connected and the weight limit ensured the detected volume was the size of an actor in the video. Since, the solution to the knapsack problem resulted in a single action annotation, we solved this problem iteratively to generate multiple annotations that satisfied the given constraints (see Figure 18).

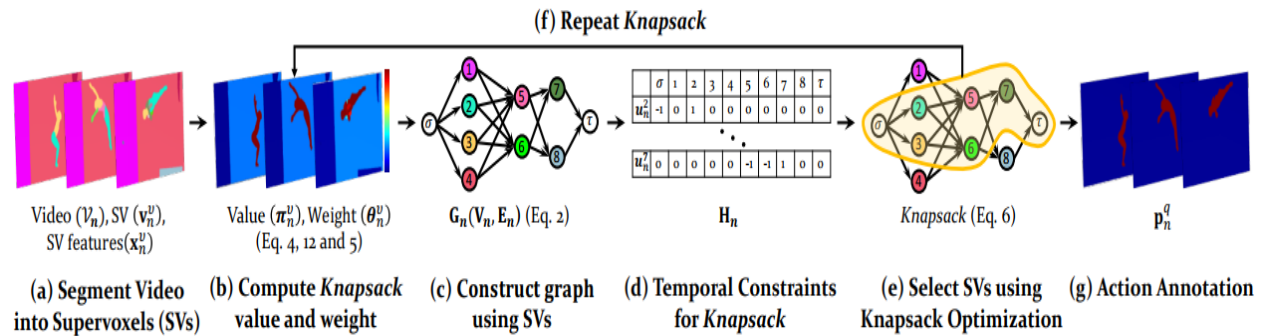


Figure 18. This figure shows the proposed knapsack approach: (a) Given an input video we extract supervoxel (SV) segmentation. (b) Each supervoxel is assigned a weight (spatio-temporal volume) and a value (score of belonging to the foreground action). (c) A graph  $G_n$  is constructed using supervoxels as nodes. (d) Temporal constraints are defined for the graph to ensure contiguous selection of supervoxels from start ( $\sigma$ ) to end ( $\tau$ ) of an action. (e) Knapsack optimization is applied to select a subset of supervoxels having maximum value, constrained by total weight (volume of the action) and temporal connectedness. (f) The knapsack process is repeated for more action annotations. (g) Annotations represented by action contours.

**Experiment results.** We evaluated our Unsupervised Action Discovery and Localization approach on five datasets: 1) UCF Sports 2) JHMDB, 3) Sub-JHMDB 4) THUMOS13, and 5) UCF101. The experimental setup, evaluation metrics, and an analysis of quantitative and qualitative results are provided. We report localization results with Area Under Curve (AUC) of ROC (Receiver Operator Characteristic) at varying overlap threshold with the ground truth.

**Unsupervised Action Discovery:** The proposed approach discovered the action labels in training videos of five datasets. We compared the performance of our approach with: K-Means, K-Medoids, Shi and Malik (S&M), Dominant Sets (DS), Spectral Clustering (SC) and the state-of-the-art DAKM clustering methods. We followed DAKM’s experimental setup and evaluation, by setting the number of clusters to be the number of action classes in each dataset. The clustering results are reported in Table 11. The numbers in the table indicates the action clustering accuracy (%). Clustering on all datasets was performed using C3D features, except for UCF Sports where we also report results using iDTF features for comparison. Table 11 shows that our approach resulted in superior performance across all five datasets. Overall, the results indicate that unsupervised clustering of human actions is a challenging problem and that known techniques such as K-Means, K-Medoids and NCuts do not perform well. Significant improvement over Dominant Sets and Spectral Clustering reflects the strength of our iterative approach. We attributed this improvement to our use of dominant sets to select a subset of coherent videos to train a SVM and to discriminatively learn to cluster actions. We observed the highest performance on UCF Sports, which contained distinct scenes and motion in the dataset, as compared to JHMDB and UCF101, that had complex human motion, independent of scene, and large intra-class variability.

Table 11. This table shows action discovery results using C3D on training videos of: 1) UCF Sports 2) Sub-JHMDB, 3) JHMDB, 4) THUMOS13, and 5) UCF101. We also report a comparison of C3D and iDTF features on UCF Sports.

	<i>UCF Sports</i>		<i>Sub JHMDB</i>	<i>JHMDB</i>	<i>THUMOS 13</i>	<i>UCF101</i>
	<i>iDTF</i>	<i>C3D</i>				
<i>K-Means</i>	34.9	64.4	41.1	40.4	62.1	45.4
<i>K-Medoids</i>	26.4	59.6	36.6	34.3	67.3	33.0
<i>S&amp;M [35]</i>	44.2	63.2	45.9	37.9	54.4	7.8
<i>DS [28]</i>	53.3	66.1	37.3	31.1	33.9	19.2
<i>SC [23]</i>	59.4	76.5	48.7	49.5	80.2	51.6
<i>DAKM [15]</i>	60.9	78.5	52.2	50.2	82.5	37.1
<b><i>Proposed</i></b>	<b>69.9</b>	<b>90.1</b>	<b>57.4</b>	<b>53.7</b>	<b>88.3</b>	<b>61.2</b>

**Unsupervised Action Annotation:** We independently evaluated the quality of annotations to localize actions by assuming perfect action class labels compared to a weakly-supervised approach. We show the strength of our Knapsack annotation approach by performing significantly better (~ 7%) than the published state-of-the-art weakly-supervised method of Ma et al. in Table 12.

Table 12. A comparison of localization performance with weakly-supervised approach on UCF Sports.

Actions	Dive	Golf	Kick	Lift	Ride	
<i>Ma et al. [21]</i>	44.3%	50.5%	<b>48.3%</b>	51.4%	<b>30.6%</b>	
<i>Proposed (Weakly)</i>	<b>59.4%</b>	<b>59.9%</b>	37.7%	<b>59.5%</b>	14.1%	
Actions	Run	Skate	Swing-B	Swing-S	Walk	Average
<i>Ma et al. [21]</i>	33.1%	38.5%	<b>54.3%</b>	20.6%	39.0%	41.0%
<i>Proposed (Weakly)</i>	<b>50.0%</b>	<b>57.9%</b>	50.0%	<b>44.6%</b>	<b>43.4%</b>	<b>47.7%</b>

**Unsupervised Action Localization:** We show localization performance using AUC curves for (a) UCF Sports (b) JHMDB, (c) Sub-JHMDB, and (d) THUMOS13 in Figure 19. The difference in performance was attributed to the supervised versus unsupervised nature of the methods. The results highlight that the proposed method performed competitively to the state-of-the-art supervised methods that use video level class labels as well as ground truth bounding box annotations. In comparison we did not use such information. With our action discovery approach and knapsack for localization, we were able to perform better than some of the supervised methods on UCF Sports. Supervised baseline results have been reported by Wang et al. on Sub-JHMDB and Soomro et al. on UCF Sports, JHMDB and THUMOS13. These baselines were computed by generating bounding boxes and connecting them spatio-temporally. A classifier trained on ground truth annotations and iDTF features was applied for recognition. Our approach outperformed these baselines on all datasets in an unsupervised manner and at higher overlap thresholds.

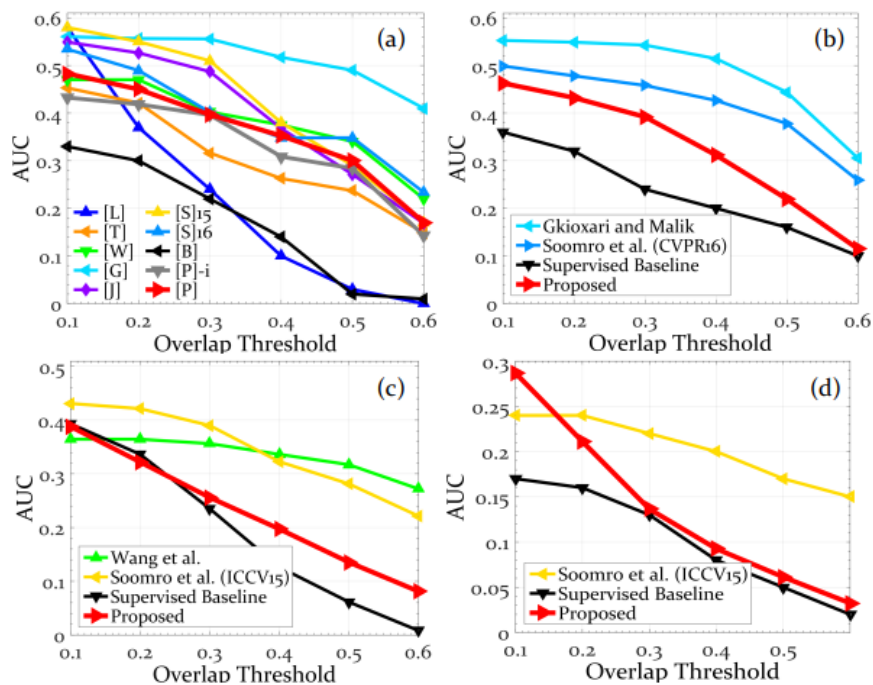


Figure 19. AUC of the proposed Unsupervised Action Localization approach, along with existing supervised methods on (a) UCF Sports, (b) JHMDB, (c) SubJHMDB and (d) THUMOS13.

Our qualitative results are shown in Figure 20 with action localization (yellow) and ground truth (green bounding box). In the case of low contrast and slow-motion the underlying supervoxel approach merged the actor with the background, therefore, when knapsack limited the localization to a specific actor volume, our approach failed to localize (Figure 20).

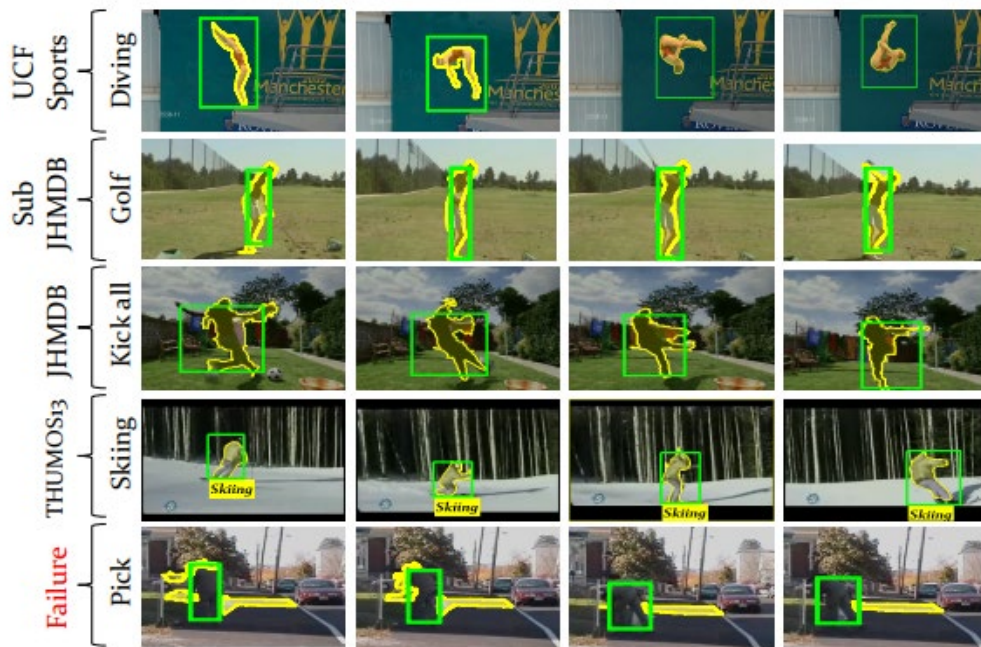


Figure 20 This figure shows qualitative results for the proposed approach on UCF Sports, Sub-JHMDB, JHMDB, and THUMOS13 datasets (rows one thru four). The action localization is shown by yellow contour and ground truth bounding box in green. Row five shows a failure case from the JHMDB dataset.

## 1.6 Human Semantic Parsing for Person-related Attribute Prediction

An important and frequent law enforcement task is the identification of a specific individual in a database. Often, an investigator is looking for a person based upon some background knowledge of ‘face attributes’ such as “wearing hat”, “goatee”, “mustache”, etc. Upon a query submitted by the investigator, such an attribute-based search will significantly reduce the number of irrelevant images presented to the investigator, thereby speeding up the investigation. These visual attributes human describable and machine detectable. Attributes are semantically meaningful tools to describe objects, scenes, actions, and events. We developed an algorithm for predicting both face attributes and body attributes from images.

To perform attribute prediction, we fed an image to a fully convolutional neural network which generated feature maps that were ready to be aggregated and passed to a classifier. However, global pooling was agnostic to where, in the spatial domain, the attribute-discriminative activations

occurred. Hence, instead of propagating the attribute signal to the entire spatial domain, we funneled them into the semantic regions. By doing so, our model learned where to attend and how to aggregate the feature map activations. We refer to this approach as Semantic Segmentation-based Pooling (SSP) where activations at the end of the attribute prediction pipeline are pooled within different semantic regions. Alternatively, we incorporated the semantic region proposals to earlier layers of the attribute prediction network with a gating mechanism. Specifically, we augmented max pooling operations such that they did not mix activations that resided in different semantic regions. We gated the activation output of the last convolution layer prior to the max pooling by element-wise multiplying with the semantic region proposals. This generated multiple versions of the activation maps that were masked differently and presumably discriminatively for various attributes. We refer to this approach as Semantic Segmentation-based Gating (SSG) (see Figure 21).

Since the semantic region proposals were not available for the attribute benchmarks, we estimated them using a deep semantic segmentation network. Once trained, the network was able to provide localization cues in the form of semantic region proposals (decoder output) that decomposed the spatial domain of an image into mutually exclusive semantic regions.

**Network Architectures.** We used Inception-V3 as the convolutional backbone for both our semantic segmentation and attribute prediction models. Its architecture was 48 layers deep and used global average pooling instead of fully-connected layers which allowed operating on arbitrary input image sizes. InceptionV3 had a total output stride of 32. However, to maintain low computation cost and memory utilization, the size of activation maps quickly reduced by a factor of 8 in the first seven layers. This was done by two convolution and one max pooling layer that operate with the stride of 2. The network followed by three blocks of Inception layers separated by two grid reduction modules. Spatial resolution of the activations remained intact within the Inception blocks, while grid reduction modules halved the activation size and increased the number of channels.

**Evaluation.** We evaluated our attribute prediction models on multiple benchmarks. Specifically, we used CelebA and LFWA for facial attributes while benchmarking on WIDER Attribute and Berkeley Attributes of People for person attribute prediction.

We compared our method with existing state-of-the-art attribute prediction techniques on the CelebA data. To prevent confusion and to have a fair comparison, Table 13 reports the performances in two separate columns distinguishing the experiments that were conducted on the original image set from those where the pre-cropped image set was used. We see that our base model with global average pooling and a more modern architecture not only outperformed our earlier average pooling model but also the semantic segmentation based ones.

Experimental results indicated that under different settings and evaluation protocols, our semantic segmentation-based pooling and gating mechanisms can be effectively used to boost the facial attribute prediction performance. That is particularly important given that our global average pooling baselines already beat the existing state-of-the-art methods. To see if SSP and SSG are complementary to each other, we also report their combination where the corresponding predictions were simply averaged. We observed that combination further boosts performance.

To investigate the importance of aggregating features within the semantic regions, we replaced the global average pooling in our basic model with the spatial pyramid pooling layer [10]. We used a pyramid of two levels and refer to this baseline as SPPNet. While aggregating the output activations in different locations, SPPNet did not align its pooling regions according to the semantic context that appears in the image. This was in direct contrast with the intuition behind our methods. Experimental results shown in Table 13 confirm that simply pooling the output activations at multiple locations was not sufficient. In fact, it resulted in a lower performance than global average pooling. This verified that the improvement obtained by our proposed models was due to their content aware pooling/ gating mechanisms.

**Balanced Classification Accuracy.** Given significant imbalance in the attribute classes, we used average precision instead of classification accuracy/error to evaluate attribute prediction. Instead, Huang et al. adopted a balanced accuracy measure. To see if our approach was superior to Huang and colleagues' under balanced accuracy measure, we fine-tuned our models with the weighted (imbalance level) binary cross entropy loss. From Table 13, we observe from the under balanced accuracy measure that all the variations of our model outperformed by large margins Huang et al.

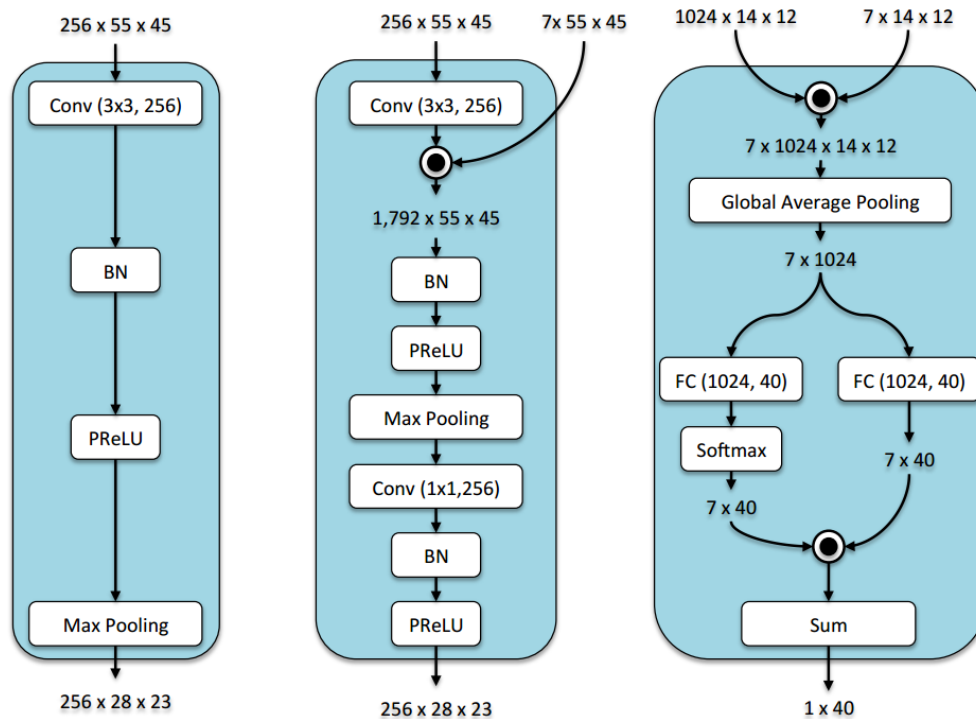


Figure 21. Left: Standard convolution layer followed by max pooling. Middle: Semantic Segmentation-based Gating architecture. Right: Semantic Segmentation-based Pooling architecture.

<b>Classification Error(%)</b>		
Method	Original	Pre-cropped
FaceTracer [12]	18.88	–
PANDA [26]	15.00	–
Liu <i>et al.</i> [13]	12.70	–
Wang <i>et al.</i> [67]	12.00	–
Zhong <i>et al.</i> [68]	10.20	–
Rudd <i>et al.</i> [64]: Separate	–	9.78
Rudd <i>et al.</i> [64]: MOON	–	9.06
Kalayeh <i>et al.</i> [1]:		
SPPNet*	–	9.49
Naive Approach	9.62	9.13
BBox	–	8.76
Avg. Pooling	9.83	9.14
SSG	9.13	8.38
SSP	8.98	8.33
SSP + SSG	8.84	8.20
Ours: Avg. Pooling	8.64	8.22
<b>Average Precision(%)</b>		
Method	Original	Pre-cropped
Kalayeh <i>et al.</i> [1]:		
SPPNet*	–	77.69
Naive Approach	76.29	79.74
BBox	–	79.95
Avg. Pooling	77.16	79.74
SSG	77.46	80.55
SSP	78.01	81.02
SSP + SSG	78.74	81.45
Ours: Avg. Pooling	79.58	81.32
<b>Balanced Accuracy(%) [3]</b>		
Method	Original	Pre-cropped
Huang <i>et al.</i> [3]	–	84.00
Kalayeh <i>et al.</i> [1]:		
Avg. Pooling	–	86.73
SSG	–	87.82
SSP	–	88.24

Table 13. Attribute prediction performance evaluated by the classification error, average precision and balanced classification accuracy on the CelebA original and pre-cropped image sets.



To better understand the effectiveness of our approach, we report experimental results on the LFWA dataset in Table 14. We observed that all the models which exploit localization cues improve our basic model. Specifically, SSP + SSG achieves considerably performed better than the average pooling basic model with 1.86% in classification error and 2.59% in the average precision. Our best model also outperformed all other state of- the-art methods.

Method	Classification Error(%)	AP(%)
FaceTracer [12]	26.00	–
PANDA [26]	19.00	–
Liu <i>et al.</i> [13]	16.00	–
Zhong <i>et al.</i> [68]	14.10	–
Wang <i>et al.</i> [67]	13.00	–
Kalayeh <i>et. al</i> [1]:		
Avg. Pooling	14.73	82.69
SSG	13.87	83.49
SSP	13.20	84.53
SSP + SSG	12.87	85.28
Ours: Avg. Pooling	14.26	81.99

Table 14. Attribute prediction performance evaluated by the classification error and the average precision (AP) on LFWA dataset.

## 1.7 Semi-supervised Action Recognition/Retrieval by Hashing

We developed a semi-supervised learning method for action recognition/retrieval using a limited number of instances. The method only required a small number of training samples, similar to the scenario in which the law enforcement analyst interactively chooses and labels a few videos of interest and the system learns a model based on both the few labeled videos as well as the large number of unlabeled videos available in a database. This approach effectively reduces the time burden on human monitors.

Our method consisted of training a deep neural network using stochastic gradient descent with supervised and unsupervised graph-based losses simultaneously. We defined two unsupervised losses based on relational (graph-based) information. We also successfully incorporated compact hashing methods to significantly reduce the complexity involved in building large graphs that were required in the graph-based semi-supervised learning process. The details about our graph-based learning for deep neural networks approach follow.

The input data was a graph (see Figure 22), the nodes of which were video segments which needed to be classified and edges (connecting lines) which represented some level of similarity between nodes. Accordingly, the existence of an edge between two nodes reflected that they should be similarly classified. Given a query which consisted of semantic labels for a set of nodes, we

trained a model which classified the labeled nodes the most accurately while also being smooth with respect to the graph. These two constraints transferred to the standard supervised classification loss and graph smoothness loss in our model training.

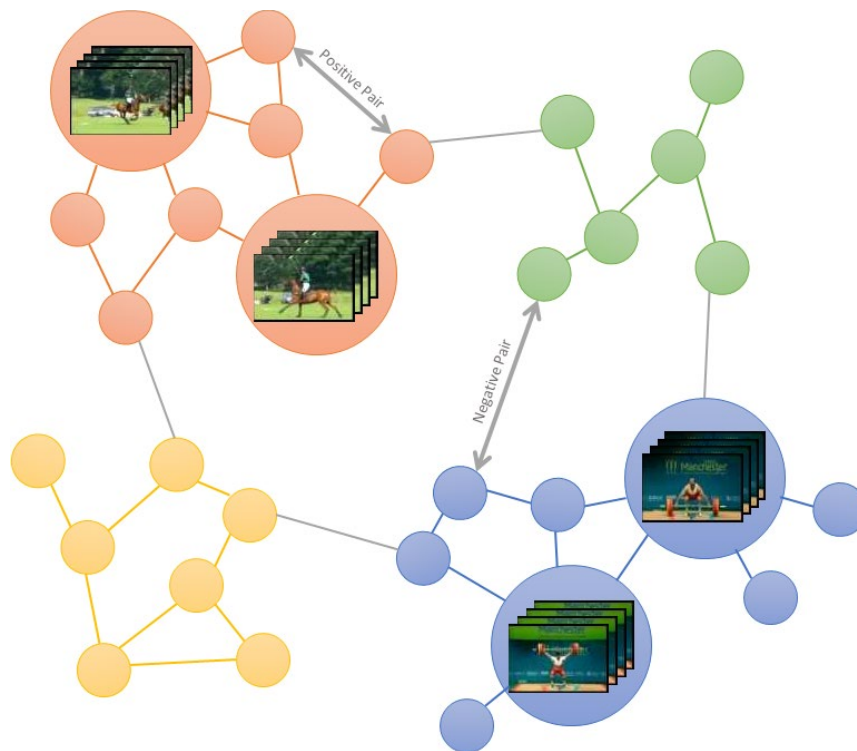


Figure 22. Semi-supervised Action Recognition/Retrieval by Hashing Example Graph. Each node represents a segment of video. The large nodes are labeled and the rest are unlabeled. Different colors show different action categories. An example of positive and negative pairs is provided.

To ensure the smoothness of the model with respect to the graph, we defined two unsupervised losses based on relational (graph-based) information given that relational labels usually do not require human supervision and can be computed automatically. As illustrated in Figure 22, a positive pair were two nodes that should be classified similarly and a negative pair were two nodes that should not be paired. The first loss, named “pairwise smoothness loss”, encouraged positive pairs of nodes to be relatively close in the feature space, while negative pairs were set further apart. In a large graph, the number of possible pairs was impractical for exhaustive optimization. Instead, we randomly sampled batches of pairs using random walks [11] and updated the parameters based on each batch. The second loss was called the  $K$  nearest neighbors (KNN) smoothness loss. Specifically, we used locally linear embedding (LLE) [12] to learn an embedding where each node can be linearly reconstructed by its  $K$  nearest neighbors in the graph. Figure 23 and Figure 24 show the proposed models with these two smoothness losses, respectively.

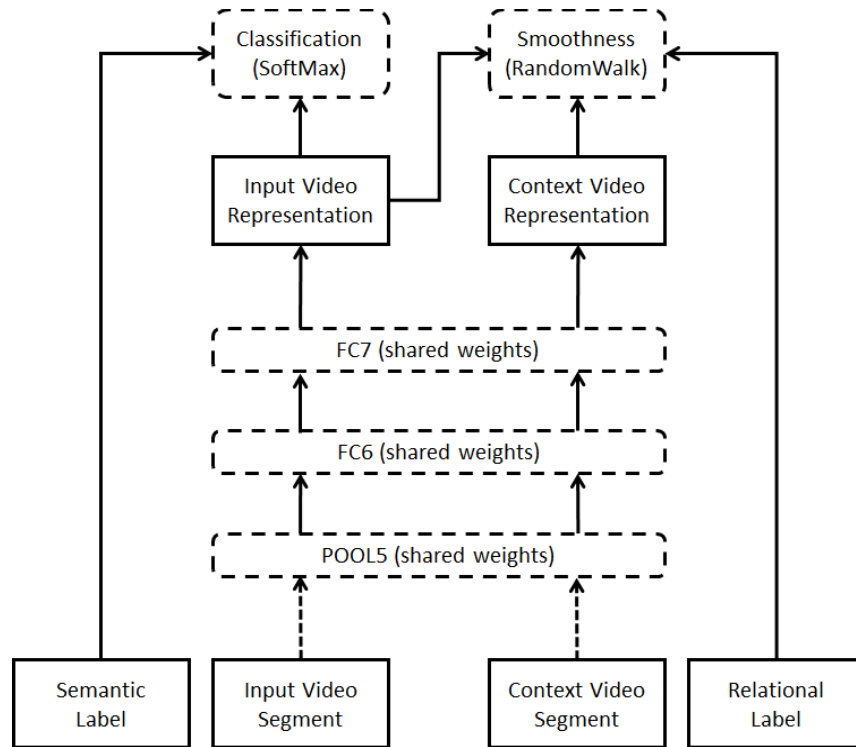


Figure 23. The Siamese network of the proposed model with pairwise smoothness loss.

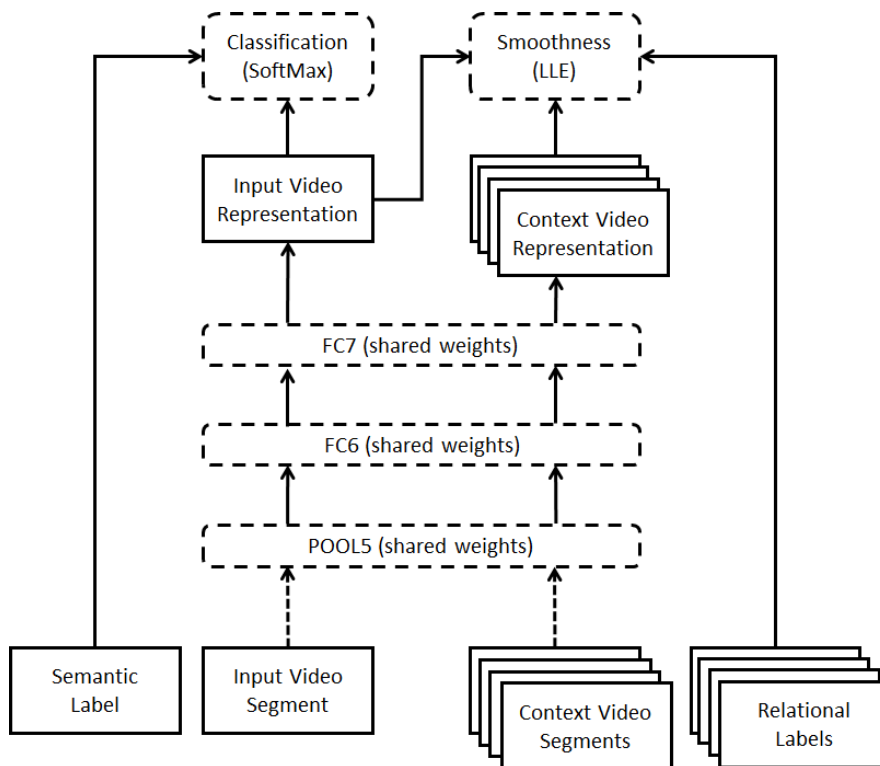


Figure 24. The Siamese network of the proposed model with KNN smoothness loss.

The performance of the model relied on the information within the relational data, i.e., the graph. To this end, we proposed two new graph construction approaches. We first built a KNN graph with edges representing similarities in the feature space (i.e. C3D [5]). However, computing a KNN graph was computationally expensive, as it involved computing pairwise distances between all edges. To make this problem tractable, we utilized hashing to convert features from the continuous space to hamming space. More specifically, we used Iterative Quantization (ITQ [14]) to convert 4096-dimensional data points into 1024-bit codes. Then we computed pairwise hamming distances using simple XOR operations which significantly sped up the analysis. We also developed an object re-identification graph, in which two videos were connected if they shared the same object. For example, if a video containing a black van was of interest, it was more likely for other videos with a black van to be of interest as well. To make a graph using object re-identification, we first generated object tube proposals from each video segment and extracted CNN features from them. Then we used ITQ to hash the features into 1024-bit codes and connected videos containing objects closer than a threshold in the hamming distance. Figure 25 depicts two examples of matched objects. Videos containing those objects were semantically related to each other and probably shared the same activity.



Figure 25. Example pairs of objects that have been matched in the object graph, one pair in the top and another in the bottom row.

For experimental purposes, we also used a hypothetical graph which was made by connecting nodes with the same action category. This graph is unrealistic because it required labels for all nodes. However, we used this graph to show that our model works best if the graph was exactly consistent with the semantics.

To evaluate the performance of our semi-supervised learning approach for action recognition/retrieval, the UCF101 dataset [15] was employed which contained 13320 trimmed videos from 101 action categories. We used the first training split of this dataset which contains 9537 videos to build the graph. Since the input to the C3D model was a 16-frame tube, we randomly sampled a 16-frame clip from each video and discarded the rest. The graph therefore had 9537 nodes, each of which was a 16-frame clip from one of the UCF101 training videos. For the edges we used the KNN graph ( $K=7$ ) as previously described. We compared three methods. The baseline was trained using only the supervised loss using the small labeled set that was provided, denoted by Supervised Only. The other two methods were semi-supervised methods

which respectively used the pairwise and the KNN smoothness losses in addition to the supervised loss, denoted by SemiSup: RandWalk and SemiSup: LLE, respectively. We also constructed the ground truth graph for comparison, denoted by SemiSup: RandWalk (GTruth Graph).

In the experiment, we randomly sampled  $m$  nodes from each class, and considered them as labeled nodes. The rest were treated as unlabeled in the training phase. For evaluation, we performed both transductive and inductive tests. In the transductive test, we reported the prediction accuracy over the unlabeled nodes of the graph, which were used in training. In inductive test, we applied the learnt model to predict the action classes in test videos that had not been seen in the training process, thus making it a more challenging task. We report in Figure 26 and figure 27 the classification accuracy for each test, which is the number of correct classifications divided by the total.

In the surveillance application, the detection of the activity of interest both in the historical data and in real-time streams was a goal. In the former, the graph could be computed offline for all existing video clips. Thus, the task was to classify nodes that were already part of the graph. In real-time applications however, there might not be time to add a new video clip to the graph as it might involve re-computation of distances. Therefore, the task was to classify video clips that were not already part of the graph. In the transductive case, we expected higher performance because the model was fitted to data it had already processed during training.

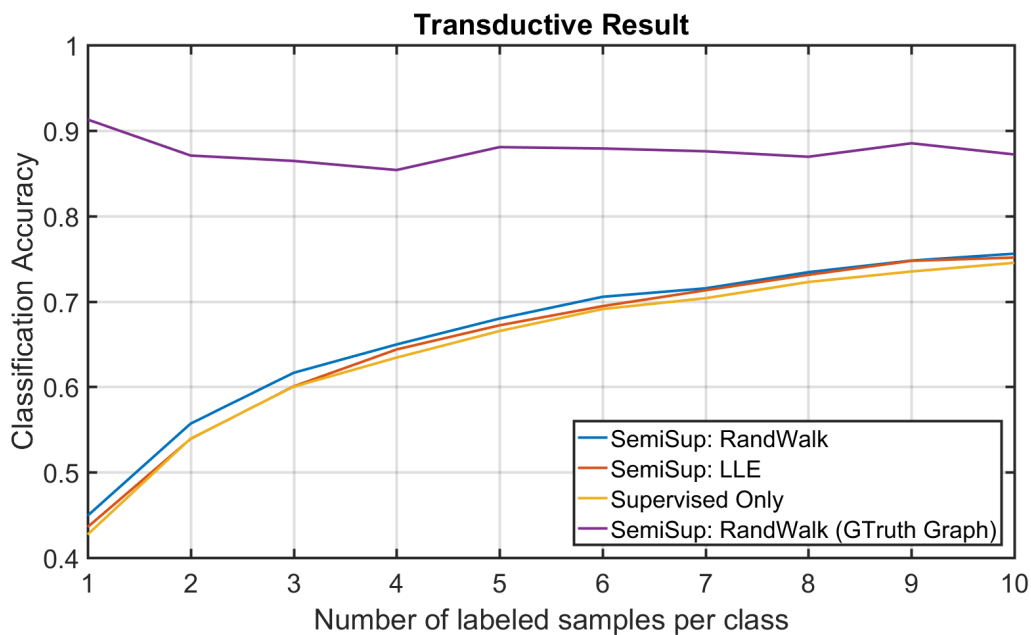


Figure 26. Transductive test results.

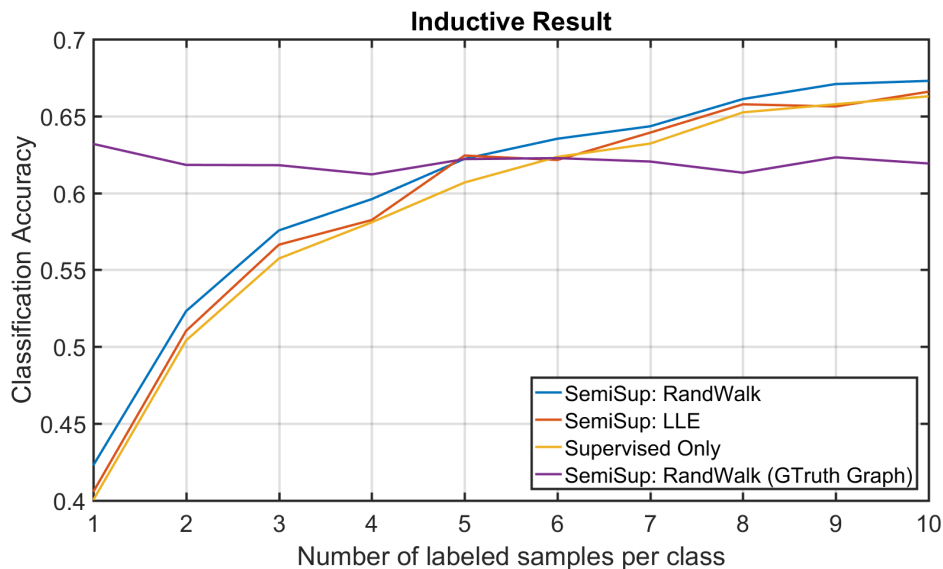


Figure 27. Inductive test results.

Figs. 26 and 27 illustrate the results for transductive and inductive tests, respectively. As can be seen, both semi-supervised methods outperformed the supervised baseline in all queries. The pairwise smoothness was slightly better than KNN, which could be due to the fact that in the pairwise smoothness we performed random walks which could explore the manifold structure of the data better. Another fact is that the learned semi-supervised model using the ground truth graph significantly outperformed other methods. Confirming the potential of our combined graph-based learning and deep neural network method, if the graph conveys more semantically consistent information, the model was capable of learning and performing classification tasks better.



Figure 28. Example results for retrieving action class “walking with dog”. Each row shows top 5 results, respectively using 1, 3, and 5 labeled instances per class. The green bounding boxes

indicate the corrected results and red boxes indicate the incorrect results.

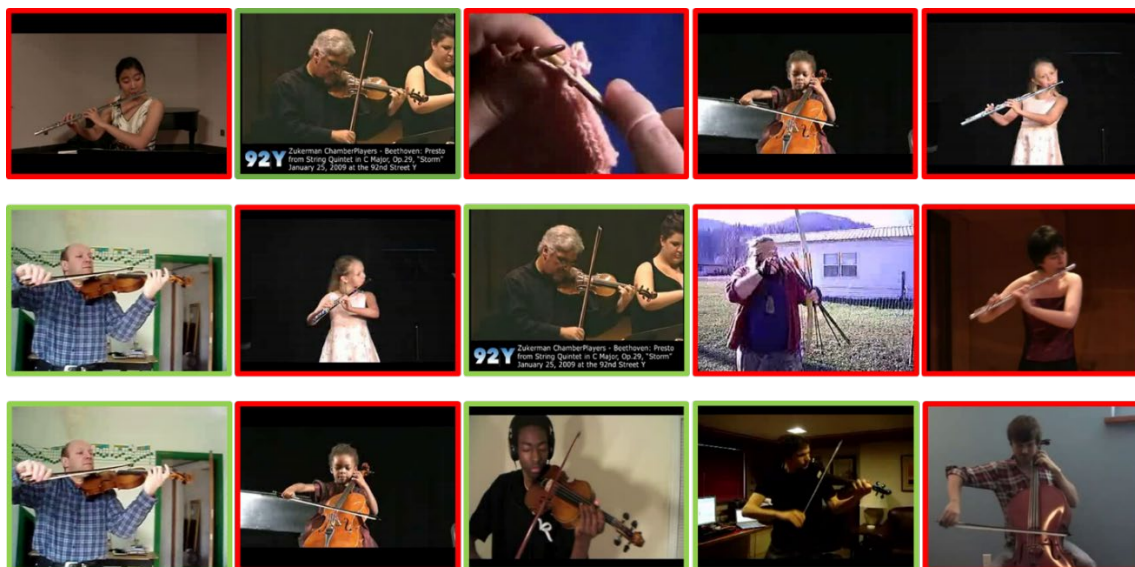


Figure 29 Example results for retrieving action class “playing violin”. Each row shows top 5 results, respectively using 1, 3, and 5 labeled instances per class. The green bounding boxes indicate the corrected results and red boxes indicate the incorrect results.

For the semi-supervised method with pairwise loss on KNN graph, we additionally illustrate example action retrieving results in Figs. 28 and 29. Consider the three rows of images in Figure 28. In the first row, we trained a model using one labeled sample per class. Then we applied the model on the test videos and retrieved the top 5 videos having the highest confidence being from the same class of actions, in this example “walking with dog”. From each retrieved video, we displayed a key frame in the figure. The next two rows in Figure 20 present the result for models trained using respectively 3 and 5 labels per class. Figure 29 illustrates similar results for the action class “playing violin”. As the results indicate, the more labeled samples used during training, the more accurate the inductive results were. Direct practical law enforcement application from our methods will be faster and more accurate identification and location of actions of interest in long video streams.

## 1.8 Summary

The developed algorithms for various vision tasks have been extensively evaluated using the standard benchmark datasets and their efficacy have been validated. As compared to the existing methods, the experimental results show our developed methods achieved superior performance. To further test our methods for real-world scenarios, we integrated four computer vision modules, namely anomaly detection, face attribute prediction, body attribute prediction, and action detection, into our public safety visual analytics workstation. The details are given in the

next section.

## 2. Public Safety Visual Analytics Workstation

We developed a computer vision system GUI that incorporates different computer vision modules for Public Safety Visual Analytics which has been installed in a dedicated workstation in the Orlando Police Department surveillance camera monitoring room. Specifically, four computer vision modules were integrated into the system: anomaly detection, face attribute prediction, body attribute prediction, and action detection. The main GUI is shown in Figure 30. The GUI is implemented using the Python programming language. In the following discussion, we show the functionality of each individual module.

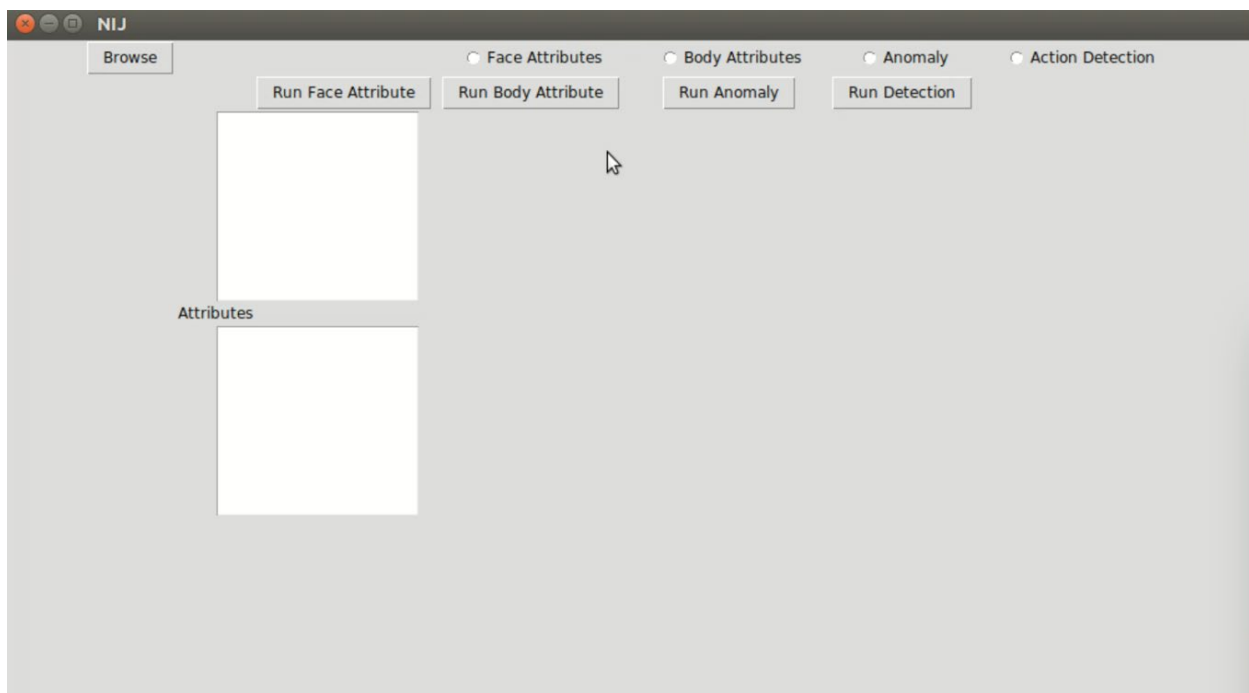


Figure 30. The main interface GUI of the computer vision work system.

### Face attribute prediction module

The face attribute module predicts attributes of a face image. Currently, we support the following face attributes:

'5 O'Clock Shadow', 'Arched Eyebrows', 'Attractive', 'Bags Under Eyes', 'Bald', 'Bangs', 'Big Lips', 'Big Nose', 'Black Hair', 'Blond Hair', 'Blurry', 'Brown Hair', 'Bushy Eyebrows', 'Chubby', 'Double Chin', 'Eyeglasses', 'Goatee', 'Gray Hair', 'Heavy Makeup', 'High Cheekbones', 'Male', 'Mouth Slightly Open', 'Mustache', 'Narrow Eyes', 'No Beard', 'Oval Face', 'Pale Skin', 'Pointy Nose', 'Receding Hairline', 'Rosy Cheeks', 'Sideburns', 'Smiling', 'Straight Hair', 'Wavy Hair', 'Wearing Earrings', 'Wearing Hat', 'Wearing Lipstick', 'Wearing Necklace', 'Wearing Necktie', and 'Young'



Below we provide an example of a face attribute prediction result in Figure 31. Given an image, the algorithm predicts the probability (between 0 and 1) of each face attribute. As seen in this example, the algorithm was able to predict the probabilities of different face attribute accurately. For example, the algorithm predicts the man had “no beard” with a probability of 0.99, and a 0.99 probability for “mouth slightly open”.

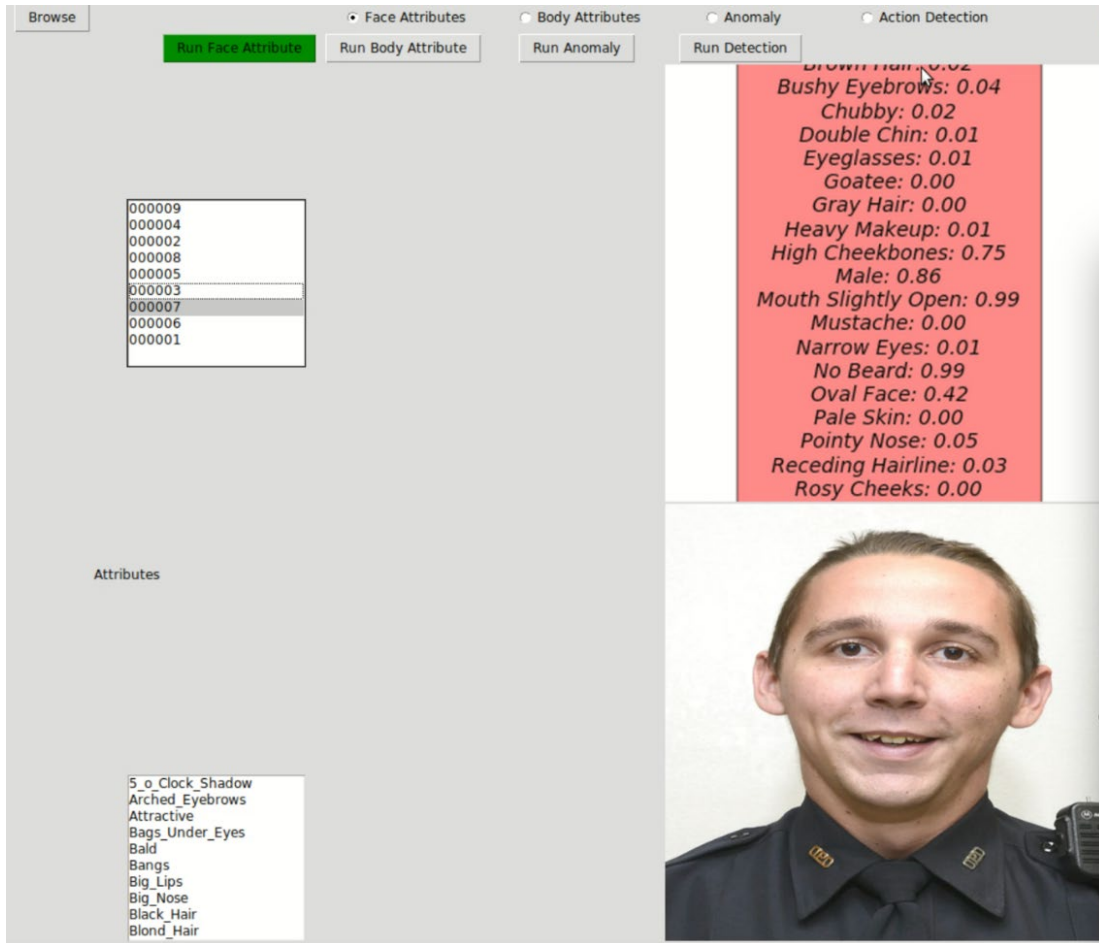


Figure 31. Face attribute prediction.

### Body attribute prediction module

Similar to the face attribute prediction, the body attribute module predicts attributes of a full body image. Currently, the algorithm predicts the following face attributes: 'male', 'longhair', 'sunglasses', 'hat', 't-shirt', 'long sleeve', 'formal', 'shorts', 'jeans', 'long pants', 'skirt', 'facemask', 'logo', and 'stripe'.

Figure 32 shows an example of the body attribute prediction results based on a surveillance image. Although the resolution of the surveillance image was low, our algorithm was able to

accurately predict key body attributes such as “male”, “t-shirt”, and “shorts”, demonstrating that the algorithm can successfully be deployed for real-world camera surveillance applications.

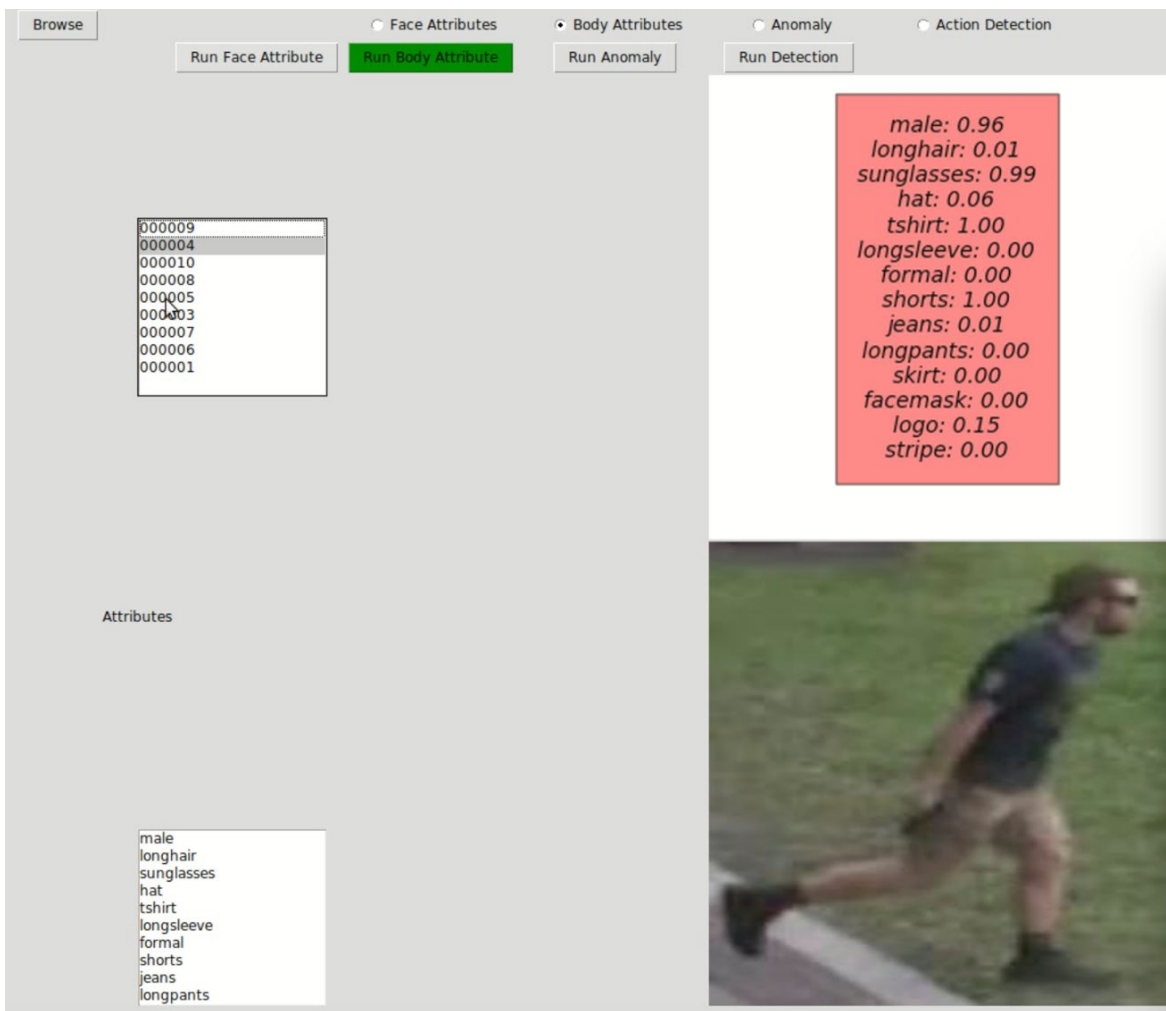


Figure 32. Body attribute prediction.

### Anomaly detection module

The anomaly detection module aims to automatically detect anomalous events such as traffic accidents, crimes, or illegal activities in surveillance videos. It addresses the law enforcement need to rapidly identify events and correctly respond to them. In addition it significantly reduces the time and effort of human monitors. A practical anomaly detection system will signal activity in a timely fashion that deviates from normal patterns and provide the time window of the anomaly. Our system processes video continuously, for example in 2 second time windows, and predicts an anomaly score for that 2-second video segment between 0 (low probability of anomaly) and 1 (high probability of anomaly). A probability threshold can be set for

the anomaly score such as 0.5 where only events that score higher than 0.5 will generate an alert flag indicating an anomaly.

The developed anomaly detection model was trained using our collected UCF Criminal Activities Dataset. The left side of Figure 33 displays the anomaly score of each video frame (on the right of the figure), which will be a value between 0 and 1. The higher anomaly score of a frame, the higher probability of an anomalous event within that frame. As depicted in Figure 33, the anomaly scores were very low (close to 0) when normal activities were present, but approached 1 when a road accident (an anomalous event) occurred in the video. Our anomaly detection system processes video data and generates scores at 35 frames per second, enabling real-time throughput. Our system can also report the temporal localization of anomalous events (e.g. time stamps of the beginning and end of an event) in a post-event video analysis stage..

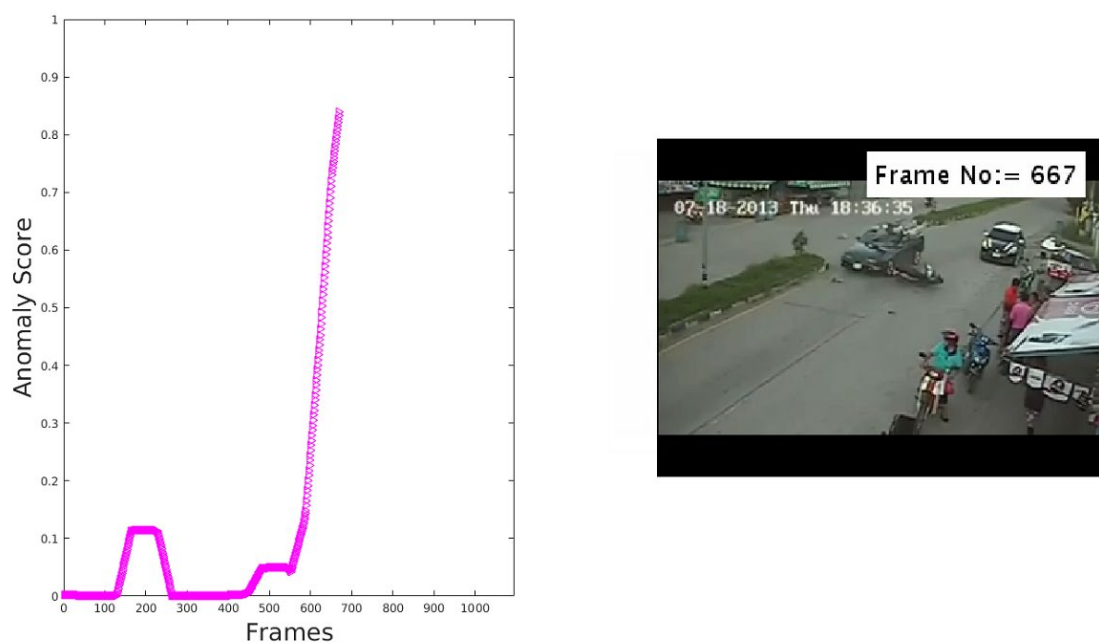


Figure 33. Anomaly detection example.

### Action detection module

The goal of action detection is to detect every occurrence of a given action within a long video, and to localize each detection in space and time. The action localization was achieved by setting bounding boxes within video frames. To simultaneously analyze the spatio-temporal information from video, in our approach we have employed an end-to-end deep network architecture for action detection in video called “Tube Convolutional Neural Network” (T-CNN). The architecture is a unified network that directly applies to an input video and is able to recognize and localize action tubes based on 3D convolution features. The interface of the action detection module is presented in Figure 34.

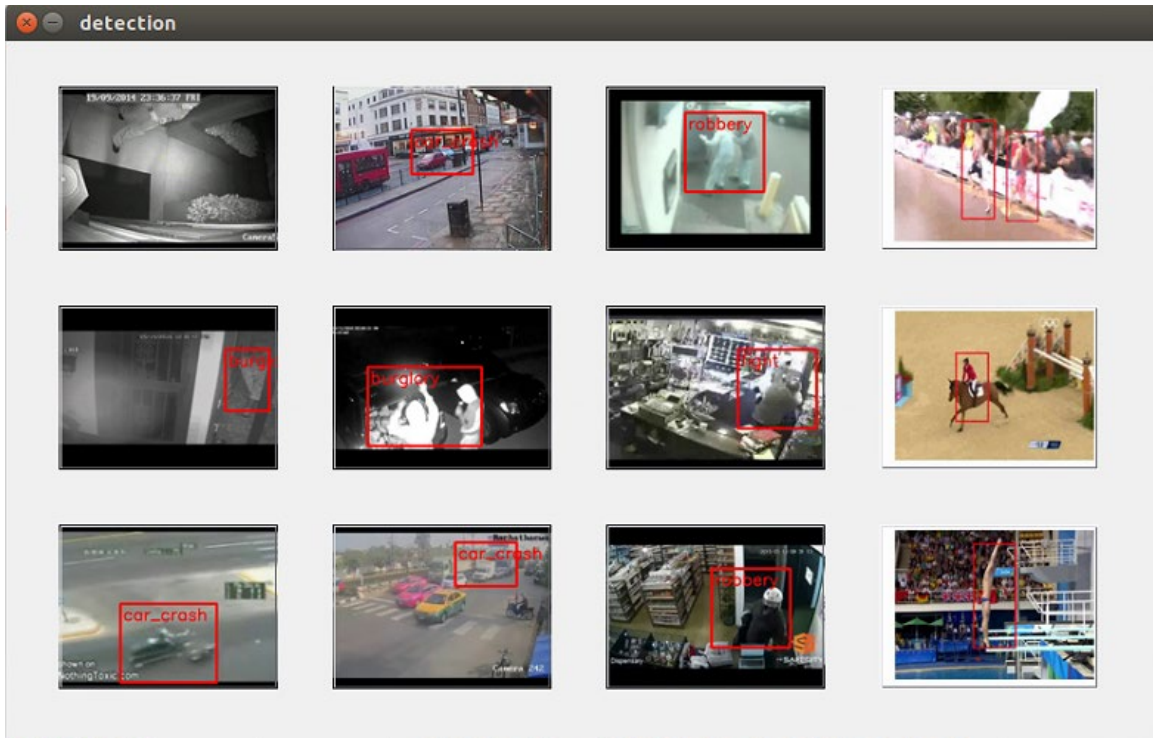


Figure 34. The interface of the action detection module. Each thumbnail represents an example of action detection (e.g. burglary, fight, etc.) in one video.

Our action detection algorithm produces action detection results when a video file is clicked on. Below, a few outputs from our action detection system results are exemplified in Figure 35. As can be seen, our method correctly located the action within the demonstrative frames.



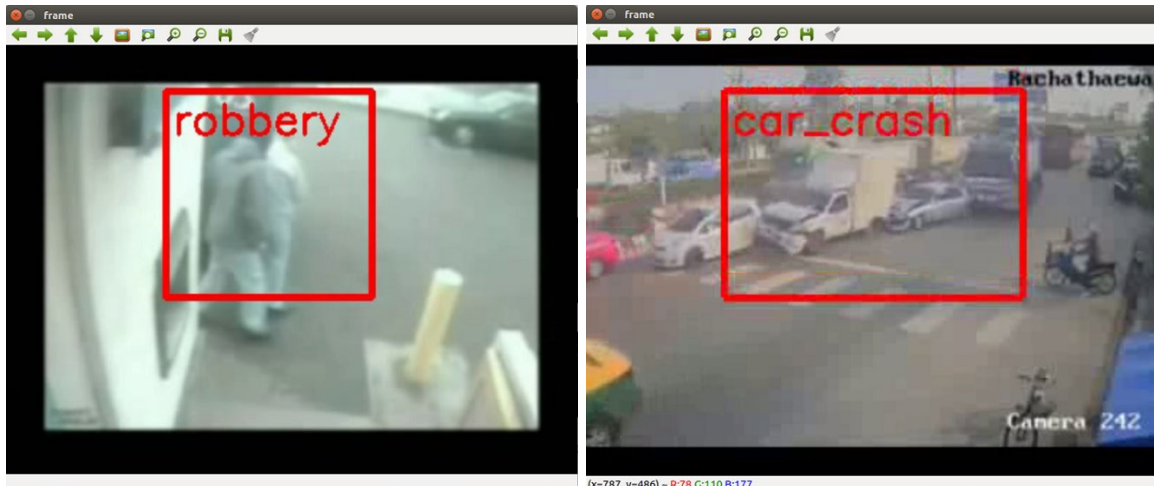


Figure 35. Action detection results.

We demonstrated the four function computer vision workstation (PSVAW) at the Orlando Police Department’s IRIS camera monitoring room on June 12, 2018. Feedback following the presentation indicated that OPD personnel planned to incorporate the computer vision capability into their operations. Discussion focused on staffing and policies and how to best proceed post grant with fully utilizing the workstation in daily police operations.

### 3. Equipment Purchased

We purchased 10 cameras and related items needed to establish a community computer vision enhanced camera network in Orlando, Florida (see Table 15). The cameras are set to be installed in the Orlando Rosemont neighborhood. As of January 1, 2017 the City of Orlando Technology Management Division (TM) acquired oversight over all camera and camera network technology in the city. While resulting in needed streamlining and updating of multiple Orlando camera networks including the IRIS camera network operated by the Orlando Police Department, the centralization of camera technologies resulted in project delays as new policies and camera network technologies were implemented. New updated “memo of understanding” (MOU), “non-disclosure agreement” (NDA), and “scope of project” documents had to be negotiated and prepared prior to installation of the OPD computer vision workstation. Establishment of a new community camera network was also delayed due to a move to and re-establishment of the Orlando Police Department’s IRIS camera monitoring room in the new department headquarters. The move to the new building in combination with local construction projects resulted in the significant disruption and downtime of the existing police IRIS camera network. The purchase of cameras and support technology was additionally delayed by a lag in camera equipment specifications from the City of Orlando Technology Management Division. Orlando TM focused on re-establishing and updating the pre-existing camera network links while simultaneously selecting and testing new camera models associated with significant equipment order delays and computer vision

workstation installation.

Item	Part#	Quantity	Description
TCD IRIS Cabinet	N/A	10	Base Weather proof Cabinet with filter, mounting plate/rail, and lock and Key
Surge Protector	DT-LAN-CAT6	10	
AXIS - Wall/Pole Mount	T91L61	10	
AXIS - PTZ Camera	Q6128-E	10	
Cisco IE1000 Switch	IE-1000-4P2S-LM	10	
Cisco IE Power Supply	PWR-IE170W-PC-AC	10	
Fiber SFP Module (Optics)	GLC-LG-SMD	20	
Ocularis Enterprise Camera License	Product Code: OC-ENT-1C	10	

Table 15. A list of camera items purchased.

**Due to the above unanticipated delays, all** equipment purchased for this project was transferred to the Orlando Police Department **prior to June 26, 2018**. The equipment include: 1. 10 Axis cameras and 10 camera cabinets; 2) Dell PowerVault MD3060e storage; 3. Digital Storm customized workstation; 4. Dell PowerEdge R730 Server.

The 10 Axis cameras have been installed in the following locations in Orlando area:

- Kirkman/Vineland (Northeast corner)
- Vineland/Turkey Lake (Northwest corner)
- Turkey Lake/Hollywood (Southwest corner)
- Hollywood/Universal (Northwest corner)
- Universal/Major (Southwest corner)
- Major/Kirkman (Northwest corner)
- Universal/Turkey Lake (Southeast corner)
- Vineland/Conroy (Southeast corner)
- Vineland/LB McLeod (Southwest corner)

The Digital Storm customized workstation was delivered to the OPD IRIS monitoring room on June 26, 2018 and on-site testing commenced. The server and storage equipment associated with the workstation were installed in the OPD server room. Figure 36 shows the software interface to display the real-time video streams from 9 IRIS cameras. The video streams are constantly recorded to dedicated local hard drivers and available for computer vision analysis.

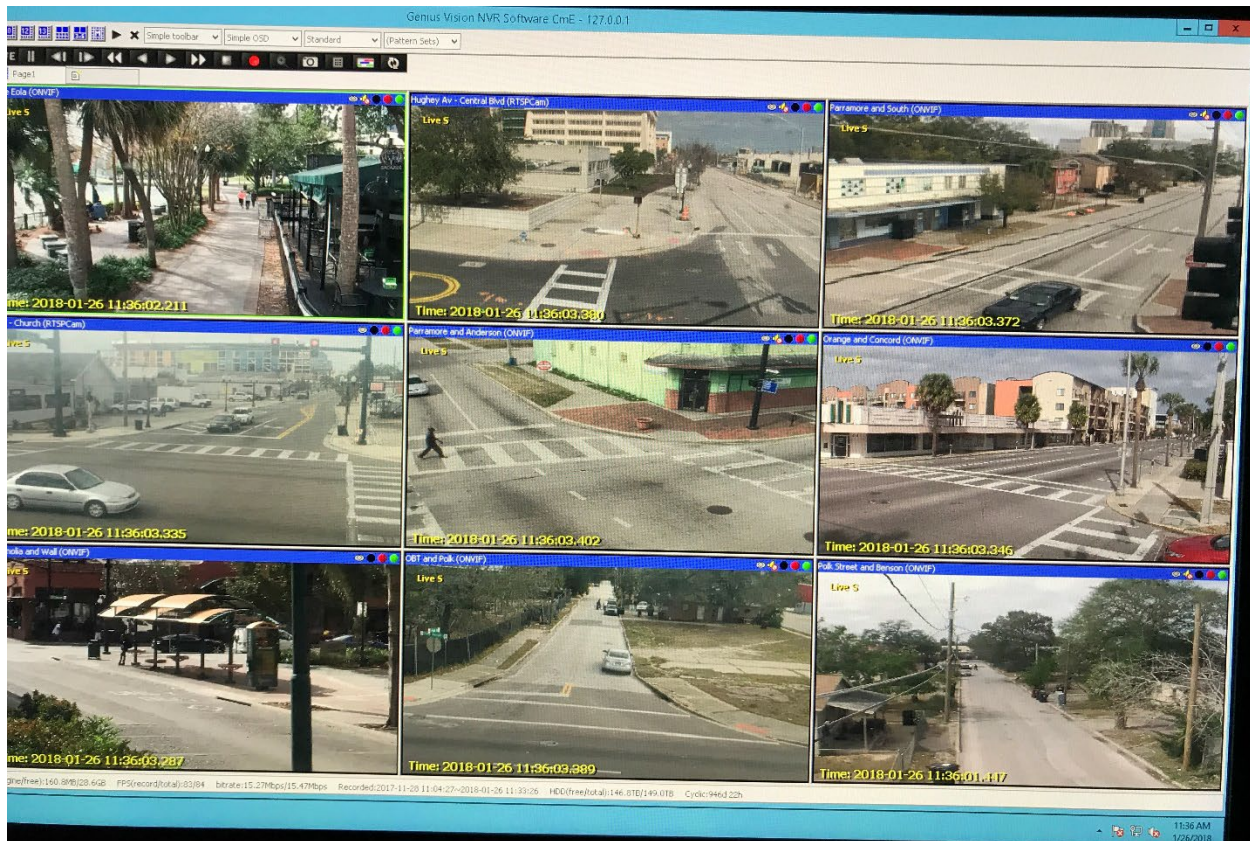


Figure 36. Software interface to display the real-time video streams of 9 OPD surveillance cameras.

## 4. Field Study

Parallel to the development of computer vision algorithms appropriate for law enforcement needs, the foundation for the field test and evaluation has been set. Three elements of the field evaluation were established. First, a pre-computer vision survey of Orlando Police Department personnel has been conducted. Second, a set of interviews with OPD IRIS camera monitors and other agency stakeholders involved in the operation of the existing departmental camera network have been administered. Third, a computerized data set composed of information derived from the “video work requests” forms maintained by the Orlando Police Department has been prepared and a strategy to follow investigations into the local criminal justice system to examine for potential impacts from the addition of computer vision capabilities to the agency’s camera network explored. Brief descriptions of the pre-computer vision personnel surveys and interviews and video work requests are provided.

**OPD Personnel Survey and Interviews.** The pre-computer vision cameras personnel survey and interviews measured attitudes and expectations regarding computer vision enabled cameras and their implementation. A ten minute online survey was conducted through an OPD dedicated email system and sent to all sworn members of the department. It was available for six weeks during September and October, 2016. In addition to officer demographics, respondents were asked about the usefulness of computer vision for policing, the impact they expected computer vision to have on their own, suspect, and citizen behaviors, and their expected impact on investigations, deterrence, and case adjudications.

Respondents. At survey administration, the Orlando Police Department had 744 sworn personnel with 409 officers responding. Regarding gender, the entire department was 83% male and was 64% white, 16% African American, and 17% Hispanic. Although not a rigorous scientific sample, the sample’s respondent characteristics closely matched the department’s distributions with the sample comprised of 83.9% males, 60.0% white, 15.8% Hispanic, and 13.9% African American providing an acceptable level of agency representativeness. The sample ranged in age from 23 to 66 years old with a medium age of 41. The sample further reflected a well-educated department with just over half of the sample (52%) holding a bachelor’s degree and 16% having graduate degrees. The average number of years of service in the Orlando Police Department for respondents was 13.5 years. Across the agency a majority (64%) of the sample respondents were assigned to patrol and an additional 33% assigned to specialized units totally 97% of sworn personnel. Three percent (3%) held administrative positions. These distributions followed comparable departmental figures of 6% administration and 94% sworn officers.

Overall, respondents reported that their behavior would not change when they were in front of a computer vision enabled camera. As shown in Table 16, a large majority of the respondents (84.8%) reported that they would not be more likely to ignore minor offenses when in the presence of a computer vision enabled camera. Almost three quarters (74.6%) of the officers believed that computer vision enabled cameras would not alter their use of force against subjects compared with just 5.1% who stated they would reduce their level of force. Asking subjects to move into the view of a camera was the only officer behavior that the officers implied may change. Forty three percent of officers reported they would ask someone to move into the camera’s field of view; 42.5% were unsure if they would ask a subject to move into the camera’s view. Regarding the respondent’s beliefs about citizens’ behavior and computer vision, the large majority similarly saw behavior



changes as unlikely. Only a quarter (25.7%) of officers believed that the computer vision enhanced cameras would improve the behavior of citizens they encountered.

Table 16. Selected OPD Personnel Expectations regarding Computer Vision Capable Cameras on Officer Behaviors.

	<u>More</u>	<u>Less</u>	<u>Unchanged</u>
Likelihood of making an arrest	4.6%	4.2%	91.2%
Likelihood to be by the book	16.9%	.5%	82.5%
Likelihood to ignore minor offense	5.1%	10.1%	84.8%
<u>Move someone to camera's view</u>	<u>42.5%</u>	<u>24.7%</u>	<u>32.8%</u>

Officers' responses were mixed when examining questions regarding the impact of computer vision on police officer discretion (see Table 17). A little less than a third believed that computer vision cameras would result in less officer discretion with another third believing that the cameras would not result in reduced officer discretion. However, OPD personnel were more in agreement with the expectation that computer vision enabled cameras would result in higher levels of administrative supervision of patrol officers. About 1 in 5 disagreed that computer vision would result in higher levels of supervision while about twice that amount agreed that it would increase supervision. Along similar lines, about one/fifth reported that computer vision would be more useful for supervisors and administration than for line officers.

Table 17. Selected OPD Personnel Expectations regarding Computer Vision Capable Cameras on Officer Discretion.

	<u>Agree</u>	<u>Disagree</u>
More useful for administrators	23.7%	22/0%
Greater supervision of patrol	36.9%	19.4%
<u>Less officer discretion</u>	<u>28.1%</u>	<u>27.3%</u>

To a large degree as shown in Table 18, OPD officers hold positive expectations about computer vision's impact on handling patrol decisions and situations. Respondents indicated that they expected computer vision enabled cameras to be effective for identifying suspects and solving crimes. Officers were more mixed in their belief that computer vision enabled cameras would be effective in detecting sick or injured people; a bit under one-sixth reported it would not be effective while 4 of 10 indicated it would be effective. Lastly, one of five respondents believed that computer vision cameras would scare off potential offenders. Regarding officer safety, computer vision cameras were perceived as likely to have little impact on police officer safety with 4 of 10 respondents disagreeing that the cameras would make them feel safer.

Table 18. Selected OPD Personnel Expectations regarding the Effectiveness of Computer Vision Capable Cameras.

	<u>Not at all</u>	<u>A little/somewhat</u>	<u>Very</u>
For identifying suspects	3.3%	45.1%	23.1%
For detecting injured people	16.0%	37.6%	6.2%
For solving crimes	3.4%	52.8%	16.6%
<u>For reducing citizen complaints</u>	<u>8.8%</u>	<u>49.0%</u>	<u>23.7%</u>

Respondents were unsure about how the computer vision enabled cameras would impact case processing (see Table 19). Officers were equally divided regarding whether computer vision would reduce the number of court appearances for officers and only about a third believed that computer vision cameras would result in more guilty pleas. However, respondents strongly felt that computer vision enabled cameras would be a valuable source of case evidence. However, the majority disagreed or were unsure about whether computer vision would speed the processing of criminal cases.

Table 19. Selected OPD Personnel Expectations regarding the Impact of Computer Vision Capable Cameras on Case Processing.

	<u>Agree</u>	<u>Disagree</u>
Speeding criminal cases	29.1%	17.8%
Valuable source of evidence	71.8%	4.3%
Fewer court appearances	27.2%	27.2%
<u>More guilty pleas</u>	<u>34.6%</u>	<u>18.5%</u>

Overall, OPD personnel were welcoming and comfortable with the idea of computer vision enhanced cameras. Six of 10 respondents welcomed computer vision cameras to the Orlando Police Department while only 1 of 10 did not. Similarly, more officers (4 of 10) were comfortable working with computer vision cameras compared with a minority (1 of 10) who were not. This initial personnel survey reflects a general ‘wait-and-see’ attitude toward computer vision’s utility among respondents.

Supplementing the department wide survey, a number of stake-holder interviews were conducted which explored in depth pre-existing attitudes and perceptions regarding computer vision and its utility for law enforcement. The interview protocol for each interview covered eight substantive areas: the interviewee’s prior knowledge of computer vision, their perceived utility of computer vision, their expectations regarding computer vision impact on officer safety and on officer field behaviors, computer vision impact on police use of force, computer vision impact on case processing and investigations, and lastly concerns with computer vision use and the level of support for adoption of computer vision personally, among OPD sworn officers, and across the OPD Department.

In general, the interviewees mirrored the larger OPD personnel survey results. For both interviewees and surveyed OPD personnel there existed limited prior knowledge of computer vision with meeting with UCF Computer Vision Center persons cited as the main and sole source of knowledge. There was however a strong belief that computer vision had tremendous potential

as a technical enhancement for law enforcement, particularly for security and investigations. There existed substantive support for the addition of a computer vision capability to the department’s existing camera network. There was little expectation that computer vision would impact officer field behaviors or use of force beyond any effects already generated by the pre-existing OPD surveillance camera network. This perception existed despite the fact that interviewees generally acknowledged that the OPD IRIS camera system was under-monitored and not well maintained. Concerning the resolution of use-of-force complaints there was an expectation that computer vision would help more than hinder officers by recording incidents more completely and reducing false reports of inappropriate use of force against officers. There were no explicitly stated concerns regarding computer vision. The expected benefits concentrated in improved monitoring and capturing of useful video and an increase in speed and accuracy in processing video. In general, the dominate view was that computer vision would be a positive upgrade to the department’s existing ‘dumb’ camera system. It was not expected to have significant across the board impacts on behaviors but was expected to have substantial administrative and investigative benefits.

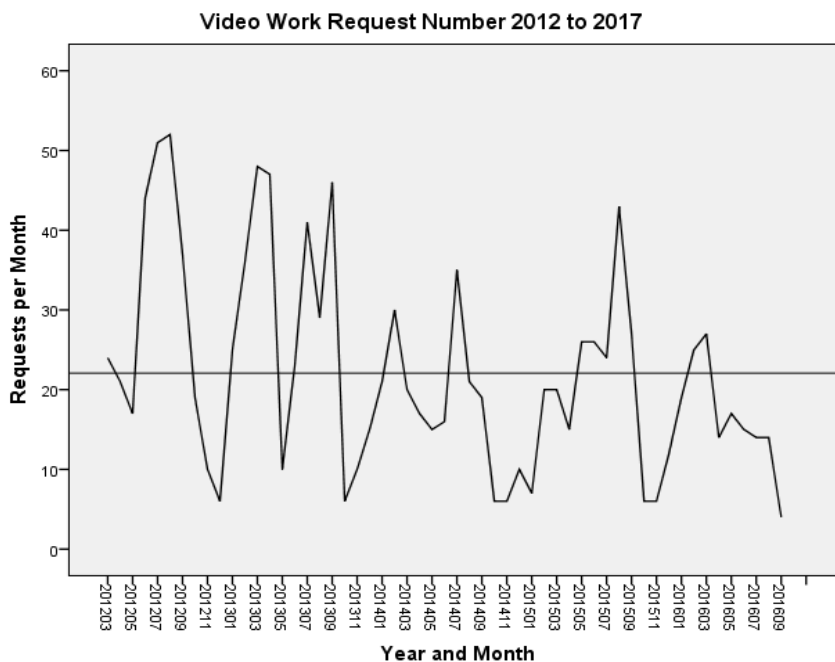


Figure 37. OPD Video Work Requests.

**Video Work Requests.** The second baseline data set collected for a field evaluation was the OPD “video work requests” (see Figure 37). A separate video work request was generated and filed each time video was requested by OPD personnel, the State Attorney’s or Public Defender’s office, or private attorneys or citizens. Information from the paper records of all video work requests since 2012 through October 2017 were entered into a computerized data file to be updated following substantial field operation. A total of 1,863 video work requests that pre-date the installation of computer vision at OPD were compiled and provided the empirical history of the utilization of the IRIS camera system before computer vision. The data provided a trend line for the number of requests over time and a measure of time in minutes spent on each individual

request. Pre-computer vision time spent on video work requests averaged a bit more than half an hour with the medium amount of time being 20 minutes. The most common amount of time spend on a request was 15 minutes, but the time spent processing video work requests ranged from a low of 1 minute to a high of 11 hours. Following a sufficient operational time period, the assessment of whether CV substantially improved processing time will be addressed.

As seen in Figure 37, requests per month show a long-term decline linked back to the under-utilization of the OPD IRIS system due to chronic monitor shortages and camera maintenance issues. Requests per month averaged 22 with a significant downward slope ( $B = -.223$ ,  $p = .046$ ) over the course of the data. Since late 2016, monthly video work requests averaged less than 20 per month. The downward trend reflects camera downtimes as the system aged and uneven staffing of the IRIS monitoring room, but also to some extent the actual use and perceived usefulness of the IRIS camera system by local criminal justice professionals. The video work request temporal flow clearly reflected the IRIS camera network suffered a decline in use over time due to issues that were felt to have prompted the department's interest in acquiring a computer vision capability.

In sum, the general takeaway from the pre-computer vision camera assessment was that there existed little organizational resistance to the adoption and trial of computer vision software that co-existed with a long term declining use of the police department's public space camera network. Department personnel were largely supportive and anticipated more benefits than concerns regarding this technology and there existed an expectation that computer vision enhanced cameras would reverse the declining utilization of the IRIS camera network.

## 5. Workshop Presentations

### 5.1 Academy of Criminal Justice Sciences convention, Kansas City, Mo., March 22 to March 25, 2017.


Two contributions at the Academy of Criminal Justice Sciences (ACJS) convention were presented. March 24, Professor Surette presented a paper at a panel on Crime Prevention Approaches: Security and Technology titled "Computer Vision Enhanced Surveillance Cameras: A Field Test and Demonstration." In addition, University of Central Florida Criminal Justice Ph.D. student, Matthew Stephenson, contributed to the conference poster session a poster presentation (see below) titled "Man Versus Machine" which provided an overview of the project.

## Man Versus Machine


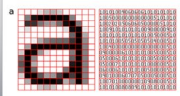
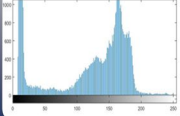
Matthew Stephenson  
University of Central Florida

### Introduction

- Cameras are watched by cameras every day; rarely are these events noticed in real time. With a grant from the National Institute of Justice, researchers from the University of Central Florida's computer analytics center will teach computers how to recognize crime and other events of interest in real time.
- Humans make poor camera monitors. "Unauthorized Hauling" is the most common and most costly violation of Florida's computer analytics center will teach computers how to recognize crime and other events of interest in real time.



### Computer Vision Basics


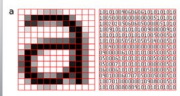
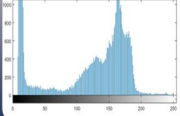




### Proposed Solution

Teach the computer to watch the video feeds for us! Researchers at the University of Central Florida's Center for Research in Computer Vision (CRCV) and Columbia University's Digital Video and Multimedia Lab (DVML) have worked to develop algorithms to automatically recognize law enforcement events of interest.

**Example Events and Objects of Interest:** Emergency vehicles, Motor Vehicle Accidents, Criminal activity, abandoned property, Criminal incidents, Robbery • "Shoplift", Drug transactions, Bathroom stalls, shootings, snuggles, "Golfing", "Suspicious person", "Officer down", "Drain cover", "Crime scene, Encounter", "Firecracker", "Weapons, handguns & rifles".

### Computer Vision Basics

### Product

**Law enforcement tasks that computer vision can address:**

- Tracking objects and people
- Video summarization
- Anomaly Detection

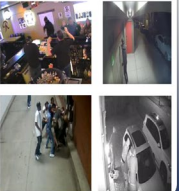
**Tracking:** Ability to track an object from one camera to another.

**Video Summarization:** Search archival or live feed, and pull specific events of interest. Reduces the need for humans to watch hours of video to find a specific event of interest.

**Anomaly Detection:** The algorithms will flag events that are defined as abnormal compared to what has been previously "seen" in the feed.

**Problems with finding abnormal events**

- "Normal" changes with time and place
- It is difficult to define all abnormal activity
- It is difficult to define all examples of "normal"



### Conclusions

**Unanswered Research questions**

Q1: Do the newly fashioned computer vision algorithms work in the lab?  
Q2: Does computer vision work in the field?  
Q3: What are computer vision impacts on a criminal justice system?

**Field Assessment**

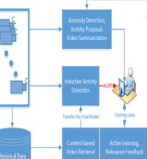
Crime data include: "all calls for service" and "all reported crime, from OPD"

Surveys and interviews with OPD officers and managers

Information from Video Work Request Form

6 cameras to be installed, pre-post data analyzed to determine if the software can reduce crime

**Public Safety Video Analytics Workstation (PSVAW)**



**Bibliography**

Bredemeyer, K & Stevens, D. (2012). Working memory and attentional blinkout. *Psychological Bulletin*, 138, 239-244.

Sera, A. (2010). "Not seeing the crime for the camera?" *Communications of the ACM*, 53, 22-25.

Stephenson, M. (2014). *Computer Vision*. New York: Routledge & Taylor & Francis.

## 5.2 International Association of Chiefs of Police (IACP) Technology Convention, St Louis, MO., May 22-24, workshop "Adding Computer Vision to Pre-Existing Police Surveillance Camera Networks.

The workshop addressed the growth of camera surveillance of public spaces having out-paced the ability of human monitors to effectively monitor the camera networks. Three issues were addressed. First, computer vision algorithms for real-world law enforcement were demonstrated. Second, the Orlando Police Department's expectations for implementing a computer vision capability into a pre-existing camera network were presented. Third, plans for a field evaluation of the computer vision enhanced camera network were outlined.

## 5.3 National Institute of Justice/RAND Workshop on Video Analytics and Sensor Fusion, Washington, D.C., July 12-13, 2017.

Principle Investigator, Professor Shah, and OPD grant liaison, Sgt. Paul Sanderlin, attended the NIJ workshop on "video analytics and sensor fusion" on July 12-13 in Washington, DC as part of an expert panel assembled to develop a roadmap for innovation in video analytics and sensor fusion technologies aimed at public safety needs. The roadmap will help guide research and development on these technologies over the next 5 years for NIJ and other organizations. This panel was sponsored by the US Department of Justice (DOJ) National Institute of Justice.

## Products

We listed all the published papers that are supported by this grant.

A. Sharghi, B. Gong, and M. Shah, *Query-Focused Extractive Video Summarization, Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, Oct. 2016.*

Hou, Rui, Chen Chen, and Mubarak Shah. "Tube convolutional neural network (T-CNN) for action detection in videos." *IEEE international conference on computer vision*. 2017.

Khurram Soomro, Mubarak Shah, *Unsupervised Action Discovery and Localization in Videos, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 2017.*

Shou, Z., Chan, J., Zareian, A., Miyazawa, K., & Chang, S. F. (2017, July). *Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 1417-1426). IEEE.*

Mahdi M.Kalayeh, Boqing Gong and Mubarak Shah, *Improving Facial Attribute Prediction using Semantic Segmentation, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, July 22-25, 2017.*

Haroon Idrees, Raymond Surette, Mubarak Shah, *Enhancing Camera Surveillance using Computer Vision: A Research Note, Policing: an International Journal of Police Strategies & Management, (Policing), Vol. 41 No. 2, 2018.*

Khurram Soomro, Haroon Idrees, Mubarak Shah, *Online Localization and Prediction of Actions and Interactions, IEEE Transactions on Pattern Analysis and Machine Intelligence, (TPAMI), 2018.*

R. Hou, C. Chen, and M. Shah, *An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos, Cornell University Library, arXiv:1712.01111 [cs.CV], 2018.*

We also have a manuscript under review entitled "A Developing Point of View: Computer Vision Applications for Police" to *The Police Journal: Theory, Practice and Principles*.

## Conclusion

In summary, this report discusses the multiple tasks that have been performed across the entire project period from January 2016 to June 2018. This includes the development of computer vision algorithms for different tasks of interest; integration of computer vision analytics into a Public Safety Visual Analytics Workstation (PSVAW); purchase and transfer of equipment consisting of 10 cameras, one computer server, and one computer workstation; and multiple professional presentations of our research. The project cumulated with a well-received demonstration of the developed computer vision capabilities and workstation to the Orlando Police Department staff at project's conclusion. Currently, OPD is considering means to continue to staff the workstation in partnership with UCF and negotiations with OPD regarding on-going utilization of the PSVAW and the department's computer vision capability are in progress.

## References

- [1] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." In NIPS, 2015.
- [2] Uijlings, Jasper RR, et al. "Selective search for object recognition." IJCV 104.2 (2013): 154-171.
- [3] G. Gkioxari and J. Malik, "Finding action tubes," in CVPR, 2015.
- [4] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in CVPR, 2016
- [5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.
- [6] Ghosh, J., Lee, Y.J., Grauman, K.: "Discovering important people and objects for egocentric video summarization." In CVPR, 2012.
- [7] Yeung, S., Fathi, A., Fei-Fei, L.: "Videset: Video summary evaluation through text." arXiv preprint arXiv:1406.5824 (2014).
- [8] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002.
- [9] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In CVPR, 2003.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in European Conference on Computer Vision. Springer, 2014, pp. 346–361.
- [11] Wu, Xiao-Ming, et al. "Learning with partially absorbing random walks." In NIPS, 2012.
- [12] Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290.5500 (2000): 2323-2326.
- [13] Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric

discriminatively, with application to face verification." In CVPR 2005.

[14] Gong, Yunchao, et al. "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (2013): 2916-2929.

[15] Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." *arXiv preprint arXiv:1212.0402* (2012).