EVAL: Explainable Video Anomaly Localization

Anonymous CVPR submission

Paper ID 6684

In this supplemental material, we will give more details on the high-level appearance and motion deep networks that we train including the training examples used and the networks' accuracies. We show more visualizations of the explainability of our method. We discuss the speed of our method both for model-building and anomaly detection. We also discuss the limitations of our approach. Finally, we include a number of videos showing the anomaly detections of our method on videos from the Street Scene, Avenue, Ped1 and Ped2 datasets.

1. Data generation

1.1. Webcam dataset

Our key motivation is to learn features that can efficiently represent generic knowledge about outdoor environments. To this end we collected surveillance videos from publicly available webcams. We collected 33 videos in total of length 3 minutes each on average.

1.2. Appearance Model

As discussed in the main paper, we created our training dataset from multiple sources (CIFAR-10 [4], CIFAR-100 [4], and MIO-TCD [7] and webcam videos as discussed above).

Many of our training examples, especially for the person, car and cyclist classes come from the webcam videos. We manually annotated the videos with bounding boxes around the people, cars and cyclists. In addition we added a subset of the car, pedestrian and cyclist examples from the MIO-TCD dataset [7]. Finally, we used the car and dog examples from CIFAR-10 [4] and the tree, house, skyscraper and bridge classes from CIFAR-100 [4]. In total, we collected 116,799 images for training and 9,240 images held out for validation spread across the 8 classes, all resized to 64x64 pixels.

After initial training of a ResNext-50 network [8], we scanned the resulting classifier across a set of 28 large images of scenes not containing any of the 8 object classes. Any patches classified as one of the objects were collected to form a new set of hard negative examples. This yielded an additional set of 62,336 background images which was added to the training set and a new object recognizer was trained from scratch. Hard negative mining was done a second time to yield one more set of 8,658 background patches. The total set of 187,793 images was used for a final training from scratch to yield the final classifier.

Figure 2 shows example 64x64 pixel training images for each class as well as the basic network architecture used for the appearance network.

1.3. Motion Model

As discussed in the main paper, we use RGB video volumes as input to our motion attribute networks and compute ground-truth attribute labels using optical flow. Every video volume in our dataset can be categorized as either 'motion' or 'background'. To create video volumes, we sequentially sample N continuous frames from a video and the corresponding $N-1$ optical flow frames. We chose $N = 10$ frames to follow the same settings as our video anomaly detection pipeline. Using the flow frames and two fixed thresholds, we define 'regions with significant motion' and 'background regions'. The first threshold (th_{mot}) is the maximum magnitude of a flow vector for it to be counted as a moving pixel. The other threshold (th_{bkg}) is the percentage of moving pixels required to say a video volume contains motion. For our experiments we select $th_{mot} = 1.0$ and $th_{mot} = 99\%$. We sample motion and background video volumes from their respective regions. We do this to improve the efficiency of selecting video volumes, as for most surveillance videos, only a small set of regions have some form of activity.

We sample $2,551,376$ 'motion' video volumes and $283,486$ 'background' video volumes. (Background samples are 10% of the total samples.) After sampling, we resize all the video volumes to spatial dimension $[64 \times 64]$. Thus each video volume is of dimension $[64 \times 64 \times 3 \times N]$ ($[h \times w \times c \times t]$) . We use 90% of these for training our models and the remainder for validation.

Figure 1. Selected frames from the webcam dataset of surveillance videos

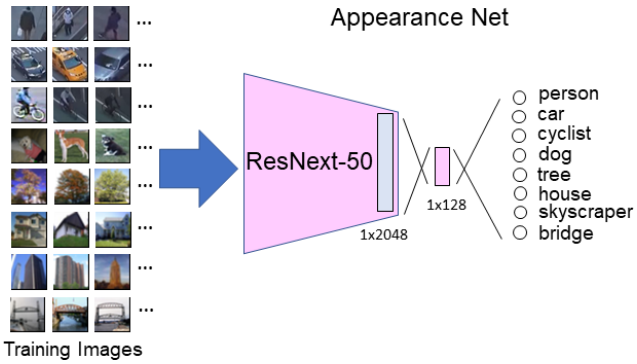

Figure 2. Sample images from each category used for training the appearance model as well as the basic architecture used for our appearance model. The input to the network is a single 64x64 pixel color RGB image and the output is an independent probability (sigmoid function) for each of the eight output classes (not softmax). Thus, there can be more than one object class recognized for a single input image.

2. Motion network Architecture

Our backbone convolutional neural network model for motion attribute learning is composed of three 3D convolution (conv) layers and three 3D max-pooling layers, followed by a fully-connected layer. Each conv layer is followed by a batch normalization layer and a ReLU activation.

The first 3D conv layers use filters of dimensions $5 \times 5 \times 5$, while the remaining two 3D conv layers use filters of dimensions $3 \times 3 \times 3$ each. The three 3D conv layers have 32, 64 and 128 filters respectively. We set the padding to "same" and stride to 1. We perform only spatial pooling for all three 3D max-pooling layers. The pooling size and the stride are both set to 2. We add a fully connected layer at the end to obtain a 128 dimensional feature vector.

For all the motion attribute learning tasks, we train separate models. For each task-specific model, we use the same backbone architecture described above with an additional task specific prediction head. For angle prediction, we add a fully connected layer with 13 units. For magnitude and background pixel percentage prediction we add a fully connected layer with 12 units and 1 unit each. Finally, for background classifier, we add a fully connected layer with 1 output unit.

3. Experiments on the Accuracy of Appearance and Motion Networks

The results in the main paper on 5 different video anomaly detection datasets show that the features learned by our appearance and motion networks are very effective for detecting anomalies in video. It is also interesting to analyze how accurate our networks are on the object recognition and motion attribute prediction tasks they are trained for. Table 1 shows correct detection and false posi-

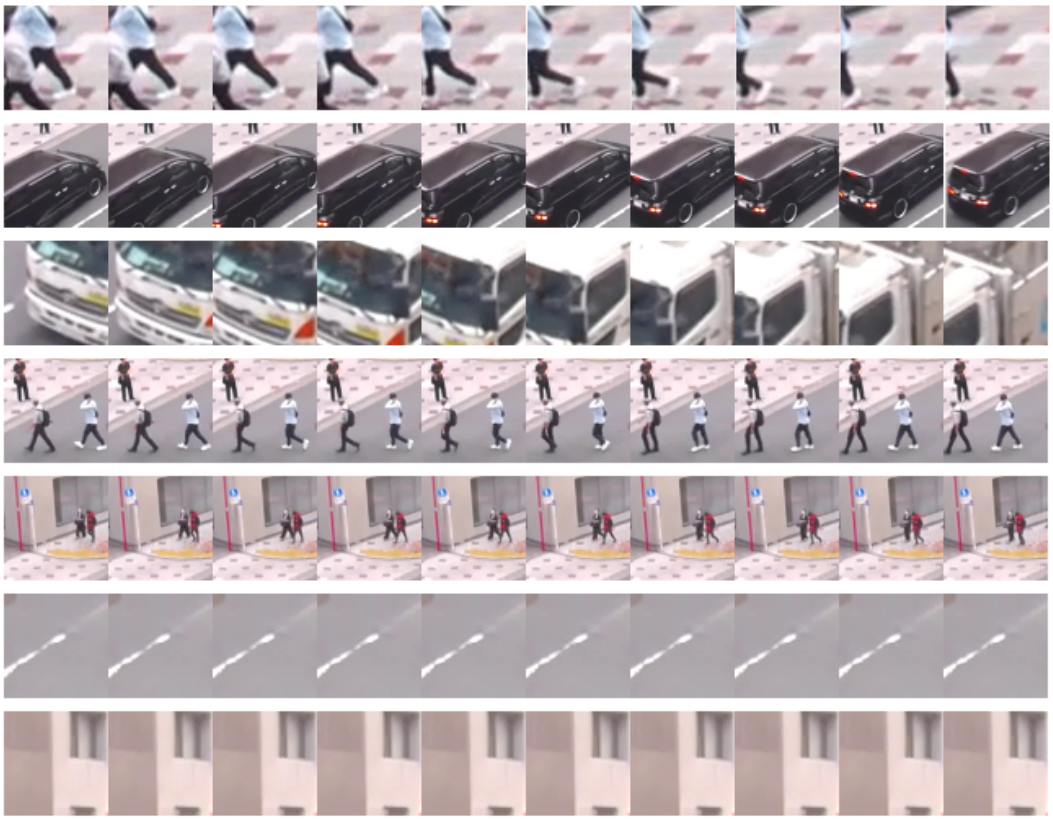

Figure 3. Examples of automatically generated video volumes for training motion attribute models. Rows 1-5 shows example video volumes from 'motion' regions, while Rows 6-7 shows 'background' video volumes.

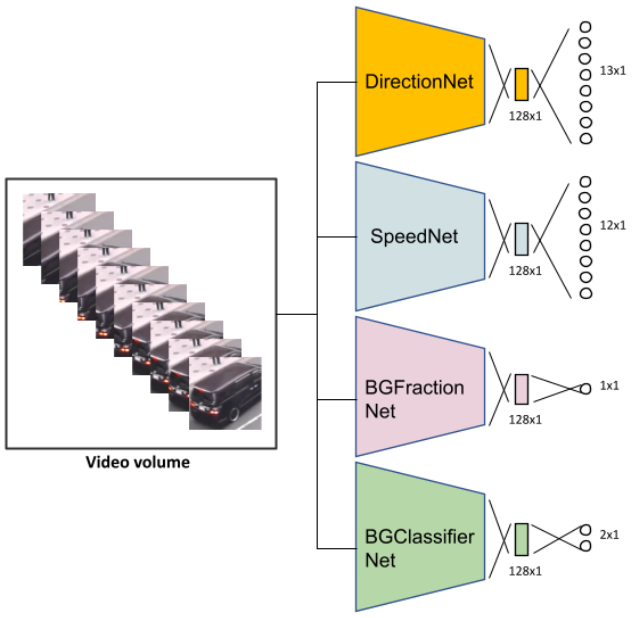

Figure 4. Motion attribute models.

tive rates for our appearance network on a held-out test set of 64x64 pixel images containing person, car, cyclist, dog, tree, house, skyscraper, bridge and background (none of the above) classes. Overall, accuracy is quite good. The cyclist class has the lowest accuracy due to the fact that for some views of cyclists, the bike is heavily occluded by the rider which can cause the cyclist to be classified as a person. This also explains why the person class a somewhat higher false positive rate than other classes.

In Table 2 we show the error rates computed for each motion attribute network. Specifically, for 'Background classifier' (BGClassifierNet) we report the classification error percentage on the held-out validation set. The table shows that the background classifier is correct over 98% of the time (1.69% error). For the 'Background Fraction' (BGFractionNet) attribute model, we report the average L1 error. This network outputs values between 0 and 1, so 0.053 average error is quite low. In the case of the 'Angle' (DirectionNet) and 'Magnitude' (SpeedNet) attribute models, which output 12 values for the 12 different angle bins, we are interested in evaluating the average deviation of the predicted estimate to the ground-truth value over all possible angle bins. To this end we compute the mean of abso-

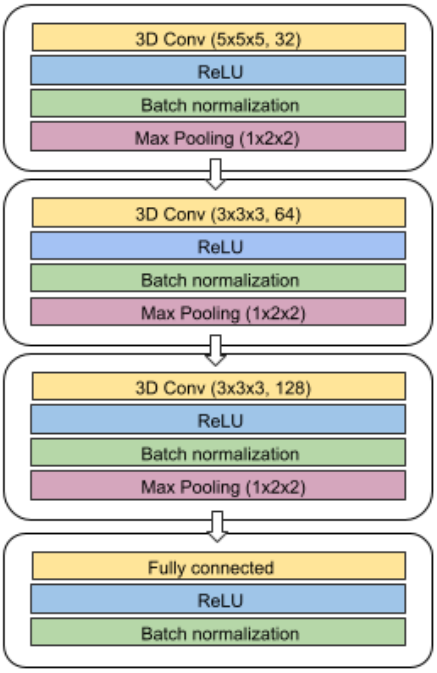

Figure 5. Backbone architecture for our motion model

Class	Correct Detection Rate	False Positive Rate
person	95.5%	3.6%
car	94.2%	1.8%
cyclist	77.6%	1.1%
dog	99.0%	0.3%
tree	99.0%	0.5%
house	89.0%	0.5%
skyscraper	97.0%	0.3%
bridge	97.0%	0.7%

Table 1. Detection and false positive rates for our appearance network on a held-out test set of 64x64 pixel RGB images.

lute difference for both the normalized angle histogram and the magnitude vector. For DirectionNet, the output is a histogram so all values are between 0 and 1 and an average L1 error of 0.0184 shows good accuracy. For SpeedNet, values do not have an upper bound but are typically between 0 and 10 pixels/frame. An average L1 error of 0.331 shows low error. Our results demonstrate that our models can accurately predict motion attributes for unseen videos with small errors.

Further improving network accuracy will lead to increases in video anomaly localization accuracy.

4. How well do appearance feature vectors for unknown classes cluster together?

In the introduction we mention that video volumes containing unknown object classes do not cause a problem for our method because the appearance feature vectors (output by our appearance network) for different images of the same object class tend to have small distance. This is the main advantage of using the network's embedding as our appearance feature as opposed to using the output class probabilities. In order to back this claim up with data, we used a set of 1000 horse images and 1000 ship images from Cifar-10 which are very different object classes from the 8 classes our appearance network was trained on. For each image, we computed its embedding using our appearance network and then computed separately the average L_2 distance between all horse images, between all ship images and between horse and ship images. The average L_2 distance between horse image embeddings was 11.1, the average distance between ship image embeddings was 18.3, and the average distance between horse versus ship embeddings was 22.0. This shows that embeddings for images of the same class tend to be closer than embeddings for images of different classes.

Attributes	Error Rate
BGClassifierNet	1.69%
BGFractionNet	0.053
DirectionNet	0.0184
SpeedNet	0.331

Table 2. Error rates for our motion networks on a held-out test set of 10x64x64x3 video volumes.

Furthermore, we ran k-means clustering using two clusters on the horse and ship embeddings. The two resulting clusters approximately separated the two object classes. One cluster contained 91% horse embeddings (and 9% ship embeddings) and the other cluster contained 77% ship embeddings and 23% horse embeddings. Again this shows that the embedding learned by our object recognizer does a good job of clustering unknown object classes.

5. Computational Analysis

We analyze computational speed of our method on the Ped2, Avenue and Street Scene datasets. For each dataset, we compute the processing speed for anomaly detection stage. The running time for model building (exemplar selection) is almost identical to anomaly detection. We compute the total time taken by adding the time taken to extract features from our high-level models and perform nearest neighbor matching. The main computational bottleneck for our method is computing feature vectors, which requires evaluating 5 different neural networks, on every video volume. A simple but effective method was used to speed this up. The important insight is that the feature vector for a video volume should not change from one time step to the next if the pixels of the video volume have not changed. If the feature vector does not change then the anomaly score will

not change either. So, for any video volume that is almost identical to the previous video volume in time, we do not need to compute its feature vector and the anomaly score for the previous video volume can simply be used for the new video volume. We use normalized cross correlation to determine whether two video volumes are nearly identical. Note that this speed-up does not prevent our method from detecting static anomalies (such as loiterers).

For each dataset, the size of the spatial regions and thus the number of regions differs since it is chosen depending on the approximate height of a person in the dataset. Furthermore, the size of frames in each dataset differs. As a result, the computational speed differs for each dataset.

Dataset	Anomaly Detection
Ped2	32 fps
Avenue	112 fps
Street Scene	12 fps

Table 3. Computational speed for our pipeleine. We show speed for each stage in frames/second.

We present our results in Table 3. For each dataset, we report results in frames per second. We used a single NVIDIA Quadro RTX 8000 GPU for feature extraction and Intel Xeon E5-2680 v4 @ 2.40GHz CPU for nearest neighbour computations. For the Avenue dataset with 640×360 resolution frames and a region-size of 128×128 resulting in 45 regions, the speed is relatively fast at over 6 frames/sec. For Ped2 (with 360×240 frames and 345 spatial regions) and especially for Street Scene (with 1280×720 frames and 897 spatial regions) our method is under 1 frame/sec.

We also show in Table 4 the running times for other published VAD methods. These times are for the anomaly detection phase only. (Note that different methods are benchmarked using different GPUs so the numbers are not directly comparable.) For the model building phase, most other methods require training a deep network on the nominal video which makes those methods much slower than ours since ours requires no network training in the model building (exemplar learning) or anomaly detection stages.

Method	Detection Speed	GPU type
Ionescu et al [3]	11 fps	Titan XP
Georgescu et al [1]	21 fps	GTX 1080Ti
Georgescu et al [2]	18 fps	GTX 3090
Liu et al [5]	25 fps	GeForce TI-TAN
Liu et al [6]	10 fps	RTX 3090
Ours	12 to 112 fps	Quadro RTX 8000

Table 4. Computational speed for our pipeleine. We show speed for each stage in frames/second.

6. Example Result Frames

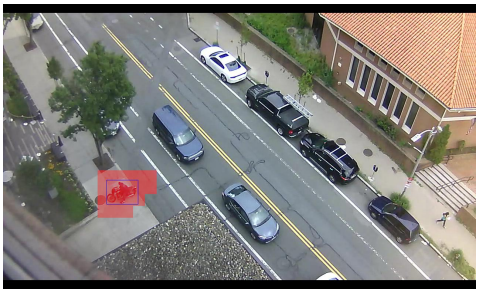

Figure 6. A test frame from Street Scene (Test031) showing the areas detected as anomalous by our method (shaded in red) and the ground truth bounding box in blue.

Figure 7. A test frame from CUHK Avenue (Test006) showing the areas detected as anomalous by our method (shaded in red) and the ground truth bounding boxes in blue.

Figure 8. A test frame from UCSD Ped1 (Test006) showing the areas detected as anomalous by our method (shaded in red) and the ground truth bounding box in blue.

We show a few frames from Street Scene, CUHK Avenue, UCSD Ped1 and Ped2 with the areas detected as anomalous from our method shaded in red and the ground truth anomalies shown as blue bounding boxes in Figures 6 - 9. We also include example results videos from each datasets in our supplementary material.

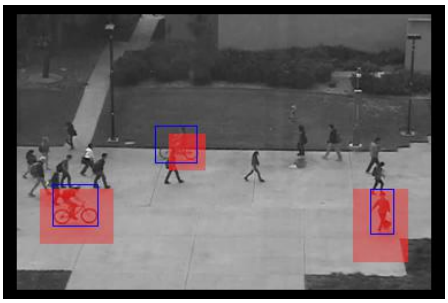

Figure 9. A test frame from UCSD Ped2 (Test006) showing the areas detected as anomalous by our method (shaded in red) and the ground truth bounding boxes in blue.

7. Additional Visualizations of Results

Figure 10 shows a visualization of the exemplars learned for a region of Street Scene on the edge of the sidewalk as well as a visualization of the high-level attributes estimated for a video volume in this region around an anomalous cyclist who is outside of the bike lanes (shown on the left side of the image). The top ten exemplars for this region (along the top of the figure) show either background/unknown objects with little motion or people moving in the direction of the sidewalk at low speed, as expected. The visualization of the high-level attributes for the video volume centered on the cyclist shown in the frame on the left, show that the video volume was estimated to contain a person moving downward at a fast speed. Although the object class is incorrect (it should be class 2, cyclist), the direction and speed are still different from the exemplars learned for this region. The closest exemplar (shown at the bottom right of the figure) is estimated to contain an unknown object (although person is the most likely class) moving down and to the right at a slow speed. The distance between the test feature vector and the closest exemplar feature vector is 2.47 which is high and indicates an anomaly.

Figure 11 shows a region of Street Scene on the street. As expected, the visualization of the top ten exemplars shows either background with little or no movement or cars/unknown objects moving mainly down and right (the direction of the street) at various speeds. The attributes of a video volume centered around a car that is making a u-turn is visualized at the bottom, left of the figure. It shows a car moving right at a fast speed. The nearest exemplar is a car moving down and right at a slow speed. The exemplar-based model does not have any examples of cars moving in this direction from the nominal data. Therefore, the test video volume has a high anomaly score and is detected as anomalous.

Figure 12 shows an example from a region on the sidewalk. The exemplars for this region show either background/unknown objects with very little movement or people/unknown objects moving either up and left or down and right. The visualization of the high-level attribues estimated for a video volume centered on a person riding a motorcycle onto the sidewalk is shown at the bottom, left of the figure. It shows that the video volume was estimated to contain a cyclist moving left at high speed. Although this is not a cyclist, it is a reasonable classification for a motorcyclist. The nearest exemplar is a person walking down and right at a fast speed. The distance between the test video volume and the nearest exemplar is large (2.55) and indicates an anomaly.

Figure 13 shows an example of a false positive anomaly detection in Street Scene. For the region on the street shown in the frame at the left of the figure, the visualized exemplars show mainly non-moving background/unknown objects or cars/unknown objects moving down and right at various speeds. The test video volume centered at the frame and region shown at the left of the figure contains the back of a car that is coming to a stop as it moves down and right. The visualization of this video volume shows that it is estimated by our appearance and motion networks to be a car moving at moderate speed up and left. This is the opposite direction to how the car is actually travelling and opposite to how cars normally travel in this spatial region. The angle network has made a mistake in this case. Thus, the closest exemplar is an unknown object (whose highest likelihood is the car class) barely moving. Because of the wrongly estimate direction of motion, the anomaly score is high, and an anomaly is falsely indicated.

As a final example, in Figure 14 we show a missed anomaly detection on Street Scene. The region we focus on is on the street and the particular video volume is centered on a cyclist who is outside of the bike lane. As expected, the exemplars learned for this spatial region show either background/unknown objects with very little movement or cars/unknown objects moving down and right at fast speeds. The visualization of the video volume containing the anomalous cyclist shows that it was estimated to contain a person moving down and right at a fast speed. The closest exemplar is an unknown object (although with relatively high likelihoods for person and car) traveling down and to the right at a fast speed. Because the motion angle and speed match fairly closely and the appearance feature vector is similar, the resulting distance (1.59) is not high enough to indicate an anomaly. This is mainly a failure of the appearance model to correctly classify the cyclist.

The visualizations of correct anomaly detections as well as false positives and missed detections illustrate how the high-level attributes estimated for each video volume lead to human-understandable explanations of the decisions our system makes. Analyzing the errors also shows that despite state-of-the-art accuracy on Street Scene, CUHK Avenue and ShanghaiTech datasets, the appearance and mo-

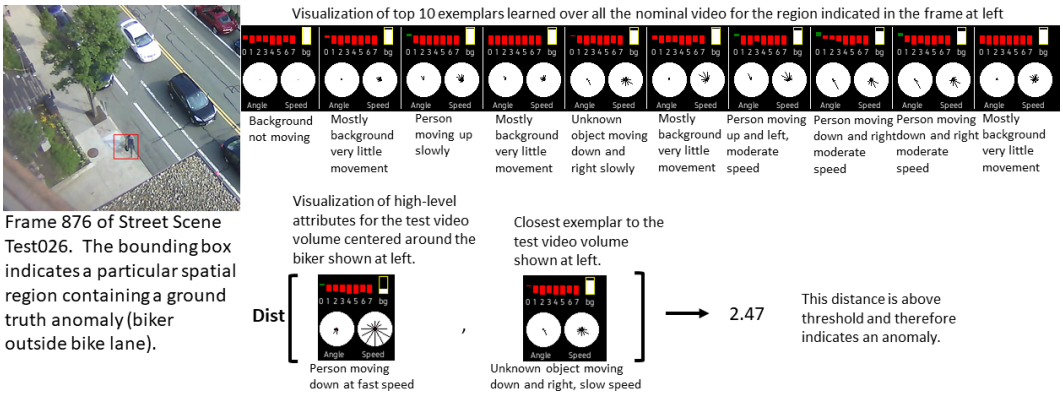

Figure 10. Visualization of the learned exemplars for a region of Street Scene and visualization of a test video volume explaining why it was detected as an anomaly.

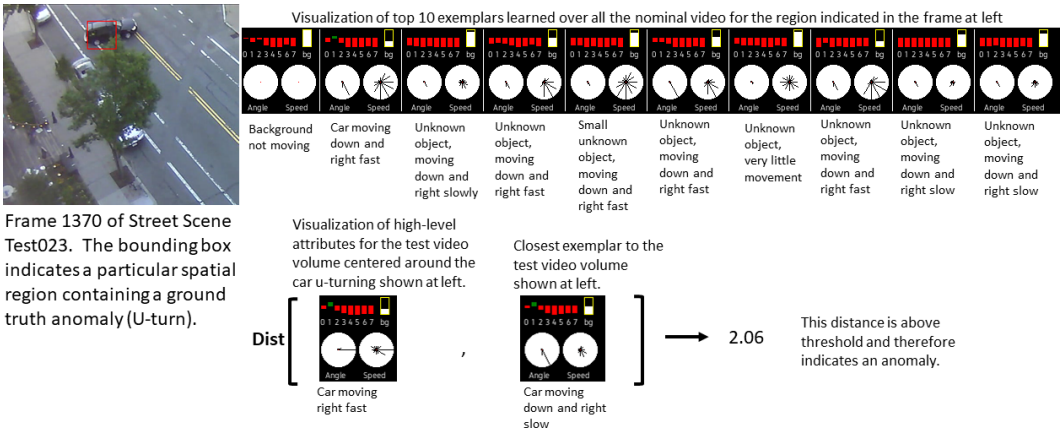

Figure 11. Visualization of the learned exemplars for a region of Street Scene and visualization of a test video volume explaining why it was detected as an anomaly.

tion deep networks are far from perfect and improvements to these networks will directly translate to higher accuracy for video anomaly localization.

8. Limitations

One general limitation of our approach is that it relies on the appearance and motion networks that estimate high-level features from a video volume. If these networks are wrong, our method may make a mistaken determination of anomalous/normal, depending on how wrong the networks are. In general, the more accurate the appearance and motion networks are, the more accurate our anomaly detection method will be.

Our current system has difficulty with a few classes of anomalies in the datasets we have tested on. On Street Scene, we tend to fail to detect anomalies consisting of cyclists or cars that are slightly outside of their proper lanes. This could be improved with a finer grid of spatial regions, but at the cost of a higher computational cost. We also tend to miss very small anomalies in Street Scene (mainly small dogs being walked on the sidewalk).

On the Ped1 and Ped2 datasets, our method has difficulty with skateboarders, especially ones that are traveling about the same speed as pedestrians. There are often only very subtle motion differences between skateboarders and pedestrians in Ped1 and Ped2 since the skateboard itself is usually barely visible.

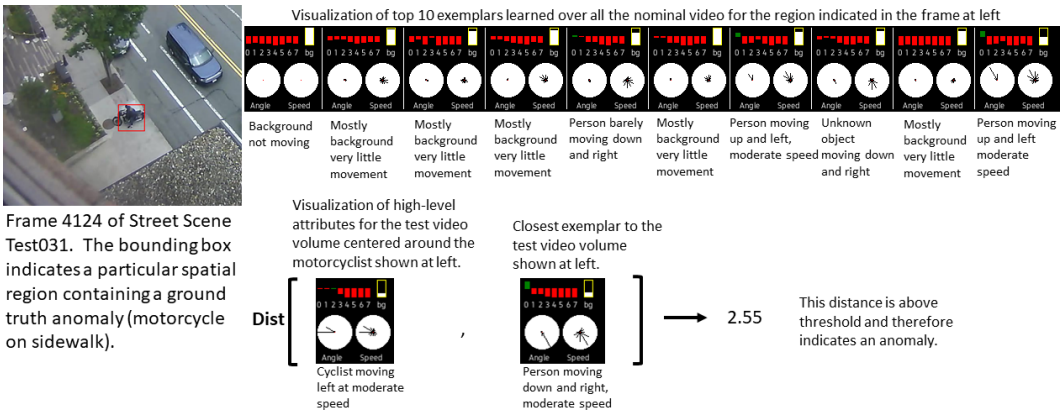

Figure 12. Visualization of the learned exemplars for a region of Street Scene and visualization of a test video volume explaining why it was detected as an anomaly.

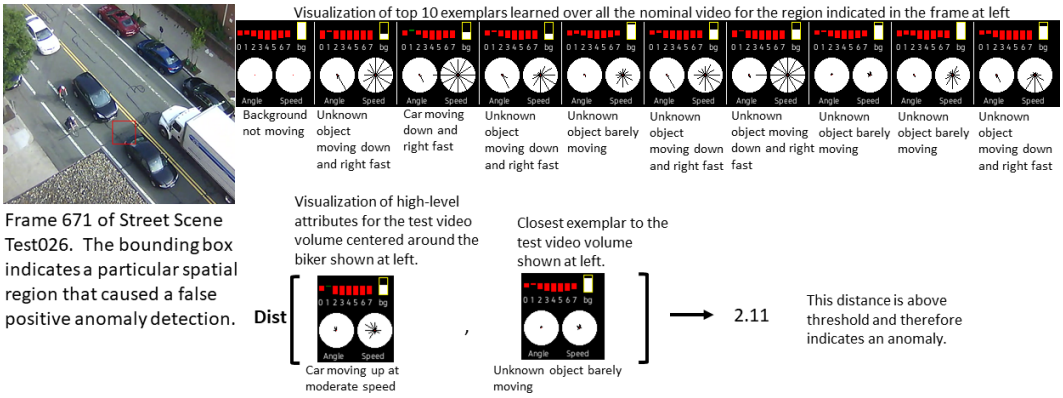

Figure 13. Visualization of the learned exemplars for a region of Street Scene and visualization of a test video volume explaining why it was falsely detected as an anomaly.

References

[1] Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. Anomaly detection in video via self-supervised and multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12742–12752, 2021. 5

[2] Mariana Iuliana Georgescu, Radu Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5

[3] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019. 5

[4] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 1

[5] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6536–6545, 2018. 5

[6] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13588–13597, 2021. 5

[7] Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018. 1

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1

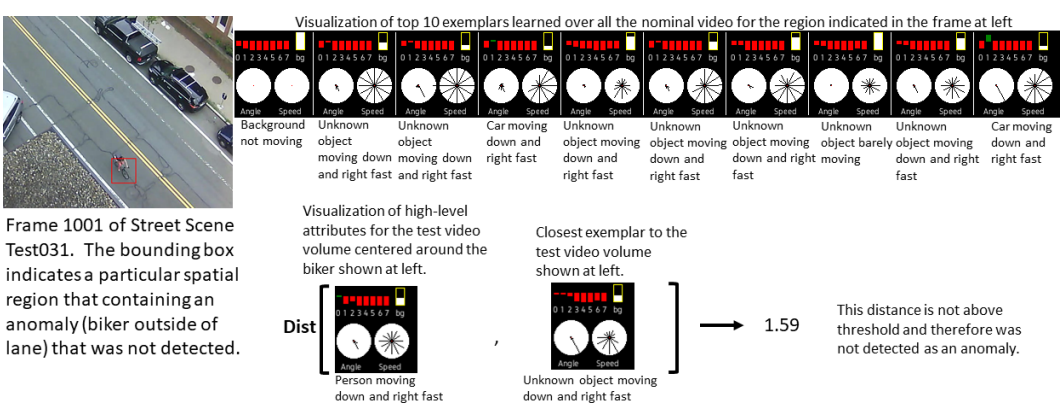

Figure 14. Visualization of the learned exemplars for a region of Street Scene and visualization of a test video volume explaining why it was not detected as an anomaly.