EVAL: Explainable Video Anomaly Localization

Ashish Singh[1,2*] Michael J. Jones[2*] Erik G. Learned-Miller[1,2]
[1]CICS, University of Massachusetts Amherst [2]Mitsubishi Electric Research Labs
ashishsingh@cs.umass.edu mjones@merl.com elm@cs.umass.edu

Abstract

We develop a novel framework for single-scene video anomaly localization that allows for human-understandable reasons for the decisions the system makes. We first learn general representations of objects and their motions (using deep networks) and then use these representations to build a high-level, location-dependent model of any particular scene. This model can be used to detect anomalies in new videos of the same scene. Importantly, our approach is explainable *– our high-level appearance and motion features can provide human-understandable reasons for why any part of a video is classified as normal or anomalous. We conduct experiments on standard video anomaly detection datasets (Street Scene, CUHK Avenue, ShanghaiTech and UCSD Ped1, Ped2) and show significant improvements over the previous state-of-the-art. All of our code and extra datasets will be made publicly available.*

1. Introduction

We are interested in the problem of spatio-temporal localization of anomalous activities in videos of a given scene. Informally, anomalous activities are events that differ from those typically observed in a scene, such as a cyclist riding through an indoor shopping mall [35]. Like many other papers on anomaly detection, this work addresses the setting in which we have access to an initial set of videos that are used to define the typical, or 'nominal' activities in a particular scene. Such a situation naturally arises in surveillance and monitoring tasks [41], where it is easy to collect nominal data, but it is not practical to collect a representative set of possible anomalies for a scene. Thus, the problem set-up is as follows: provided with a set of videos of a scene which do not contain any anomalies, (called the *nominal set*), the goal is to detect any events in a test video from the same scene that differ substantially from all events in the nominal set [29, 35]. In defining anomaly detection, it is important to consider the role of *location*. In real-world surveillance scenarios, an event may be normal in one location but anomalous in another. For example, a car driving on a road is typically not anomalous, while one driving on a sidewalk typically is. In view of this, we adopt the following definition [35].

Definition 1 *An anomaly is any spatio-temporal region of test video that is significantly different from all of the nominal video in the same spatial region.*

Unlike most recent work in anomaly detection, a key goal of our work is to produce not only a set of anomalies, but a simple and clear explanation for what makes them anomalous. We are motivated by how people tasked with watching video from a stationary surveillance camera would detect an unusual incident. While monitoring a scene, we expect a person to note the types of objects seen (people, buildings, cars) and the motions of those objects (walking east or west on a sidewalk, driving northwest on the street) to characterize the given scene. The person could then notice an anomaly when the objects or motions do not match what has been seen before. The person could also explain why something was anomalous.

We design our video anomaly detection system using this sketch of how a human would solve the problem as motivation. We want to use deep networks to give a high-level understanding of the objects and motions occurring in a scene. By 'high-level', we mean at the level of whole objects and not at the level of pixels or edges. To do this, we train deep networks that take a spatio-temporal region of video (which we call a *video volume*) as input and output attribute vectors representing the object classes, the directions and speeds of motion and the fraction of stationary pixels (which gives information on the sizes of moving objects) occurring in a spatio-temporal region. The feature vectors from the penultimate layers of these deep networks yield high-level representations, or *embeddings*, of the appearance and motion content of each video volume. Ten frames are used for video volumes in our experiments.

Unlike many other recent works, we do *not* learn new embedding functions (i.e. networks) for new scenes. We use the same embedding networks for every environment. Instead, to characterize the nominal video for a new scene, we store a representative set of all the embeddings found in

*equal contribution

the nominal video. That is, for every video volume in the nominal video of a new scene, we calculate our representations of appearance, motion direction, speed, and background fraction. We then reduce this set of embeddings to a smaller set which we call *exemplars* by removing redundant embeddings. We select a separate set of exemplars for each spatial region. This results in a compact, accurate, and location-dependent model of the nominal data in a new scene. Since there is no training of deep networks for each new environment, modeling a new environment is 'lightweight' compared to many other methods, making it efficient to model new scenes. Our exemplar model also allows very efficient updating if new nominal video is introduced. This is a crucial property for video anomaly detection methods because, in practice, it is unrealistic to assume that the initial nominal video covers every possible normal change. New nominal video will occasionally need to be added, making it critical that models are easy to extend.

Given test video of the same scene, we compute our high-level features for each video volume. We then compare these to the exemplars stored in the nominal model at the corresponding spatial region. Any test feature with a high distance to *every* nominal exemplar for that region is considered anomalous. Because the feature vectors map to human-interpretable attributes, these attributes can be used to give human-understandable explanations for why our system labeled video volumes as normal or anomalous. We define a method as 'explainable' if it can give human-understandable reasons for its decisions. Details of how our system provides explanations are given in Section 4.4.

In summary, we make the following key contributions: 1) We show that modeling scenes using high-level attributes leads to robust anomaly detection. 2) We introduce the idea of directly estimating high-level motion attributes from raw video volumes using deep networks. 3) We show how these high-level attributes also allow human-interpretable explanations. 4) Finally, we demonstrate an alternative to much of the previous work that is based on learning to reconstruct the nominal data. Our alternative approach is practical since it does not require training deep networks for each new scene and allows for simple and efficient updates to a scene model given new nominal training data.

2. Related Work

Most Video Anomaly Detection (VAD) methods can be analyzed in terms of their representation learning or their detection method.

Representation of Nominal Data: Early approaches to VAD [2, 3, 7, 20, 30, 40, 47] primarily relied on the usage of handcrafted features. This included features like spatio-temporal gradients [17, 24], histogram of gradients [28, 40], flow fields [2, 3, 30, 47], histogram of flows [7, 40, 41], dense trajectories [28, 43] and foreground masks [3]. Recently, most authors have used deep learning for this task [1, 9, 10, 14–16, 21, 26, 34, 36, 37, 39, 42, 45]. These methods either use a pretrained model [15, 17, 26, 36, 39, 42] for feature extraction or train a model to specifically optimize for a particular task related to anomaly detection. These tasks can generally be categorized as either a variant of training an auto-encoder architecture to minimize the reconstruction error of nominal frames [6, 14, 16, 22, 25, 31], a generative adversarial network (GAN) to model nominal frames [21, 25], or future frame prediction given a sequence of nominal frames [21, 44]. To further improve their performance, recent works have tried specialized architectures and training methodologies. Particularly [8, 22, 32] transform their respective generative model by memory-based modules to memorize the normal prototypes in the training data. Most recently [38] proposed a module based on masked convolution and channel attention to reconstruct a masked part of the convolutional receptive field. A key drawback of reconstruction and future frame prediction methods is that they do not generate interpretable features. It is not clear what aspects of the video make it difficult to reconstruct since there is no mapping to higher-level features as in our work. Our work mostly aligns with methods utilizing pretrained models. However, unlike most of these approaches, we incorporate the output of our pretrained models to interpret model decisions. We additionally make sure that the predictions of our model generalize to a wide variety of scenes through our data generation and training procedures. One high-level motion attribute that our method learns is similar to the histogram of flow feature used in some early work [7, 40, 41]. However, instead of computing the histogram of flow from an optical flow field, our method learns a deep network to predict these features directly from RGB video volumes and then uses the network's learned feature embedding as the representation.

Detection Methods: Most methods use either standard outlier detection methods [5] as an external module or utilize the reconstruction strategy to predict anomalies. General methods of detection that most authors have used in the past include one-class SVM [17, 28, 42, 49], nearest neighbour approaches [9, 11, 15, 33, 34], and Probabilistic Graphical Models [3]. In [4, 13], pseudo-anomalous samples are used during training to improve discriminative learning. In [12], an object detector is used to focus on regions around objects and then networks are trained for various 'proxy' tasks (such as predicting the arrow of time) on the nominal data. Thus, unlike our work, they require training deep nework models for each different scene. Our work has some similarity to the work of [33, 34] in that we also use an overlapping grid of spatial regions, build exemplar-based models and use nearest neighbors distances as anomaly scores. The high-level features that we use are the biggest difference as compared to the pixel-based features used in their

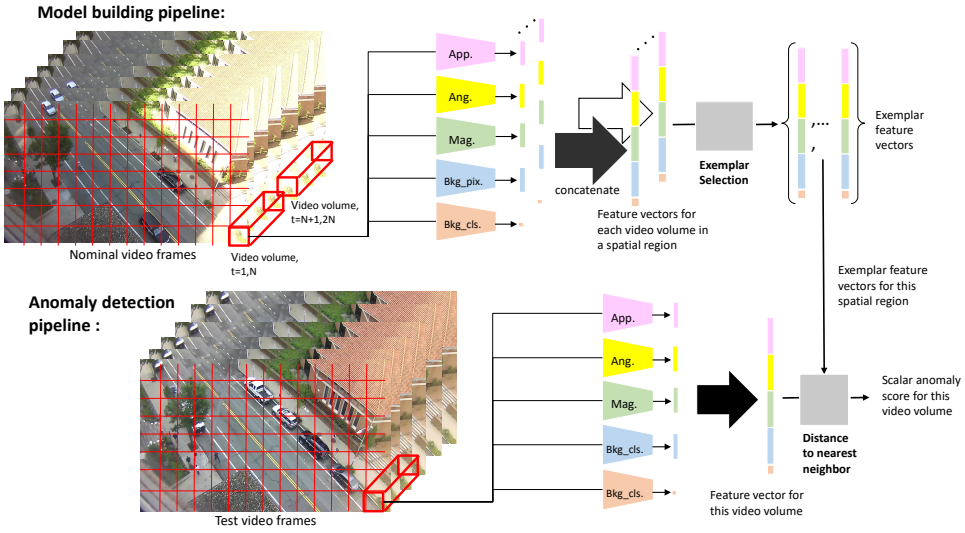


Figure 1. Our pipeline for building a location-dependent model of nominal video and detecting anomalies in test video. During the model building phase, we extract a high-level representation of each video volume using our appearance and motion networks. Using the exemplar selection method, we select a representative subset of video volumes for a given spatial region. By comparing video volumes in test video to the exemplar set we can detect anomalies.

work. Our high-level features allow for explainable models as well as much smaller models than theirs.

Explainable VAD: Our work is similar in spirit to the work of [15,46] with respect to providing explanations for detecting anomalies. In [15], the authors pre-train their feature extractor on public image datasets (MS-COCO and Visual Genome) to detect objects and predict their attributes and actions. They further use these predictions for 'event recounting' on VAD benchmarks. In [46], the authors utilize models pretrained for semantic segmentation, object classification, and multi-object tracking and use the output of these models directly as their feature representation.

Despite these coarse similarities, virtually all of the details of our methods are different. Specifically, in [15], they rely on object proposals to find candidate anomalous regions which can lead to missed detections for objects not represented in their training data. Our action/motion classes are also very different - ours being more generic (direction distributions and speed of motion) while in [15] are much more specific (bending, riding) and hence not applicable to a wide variety of scenarios. The method of [46] is specific to detecting and tracking pedestrians and is not a general video anomaly detection method. Furthermore, unlike ours, their method does not spatially localize anomalies.

3. Our Approach

Our method consists of three distinct stages: high-level attribute learning, model building, and anomaly localization. The high-level attribute learning stage is done only once and uses training samples that are independent of any video anomaly detection dataset. The resulting deep networks learn general representations of object appearances and object motions which can then be used in the subsequent two stages to build a model of a specific scene and to localize anomalies in that scene, for a wide variety of surveillance scenarios. The outside data used to train our high-level attribute models are equivalent to the outside data used in various prior works on VAD; for example, the MS-COCO and Visual Genome data used to train the models of Hinami et al. [15], the outside data used to train object detectors in [11, 12, 17, 42], as well as the many deep models pretrained on ImageNet and applied to VAD. Our outside data is not used to build models of a scene.

3.1. High-Level Attribute Learning

For this stage, our main objective is to learn features that are **(a)** transferable across scenes and **(b)** interpretable. Given this motivation, we learn an object recognizer for our appearance model as well as regression networks for estimating the following motion attributes for a given video volume: the fraction of stationary pixels, the distribution of motion directions and the average speed of movement in each direction. We also learn a classifier to indicate whether a video volume is stationary or not. We will describe each of these deep networks in the following subsections.

To clarify our terminology, we use the term *'high-level attribute'* to denote the object classes, histogram of motion directions, vector of motion speeds or fraction of stationary pixels which are the final outputs of the various deep networks that are learned. The term *'high-level feature'* denotes the feature vector from the penultimate layer of one of

the deep networks. A high-level feature can be mapped to a high-level attribute using the final layer of that network.

3.1.1 Appearance model

We formulate the task of object recognition as a multi-label image classification problem as any given input image patch may contain more than one object class (or none).

Training Data: We are particularly interested in learning to recognize objects that have high likelihood of being present in outdoor scenes. To this end, we select the following 8 categories as our primary set of object classes : [Person, Car, Cyclist, Dog, Tree, House, Skyscraper, and Bridge]. For our formulation, we want the learned features to generalize across different domains. To achieve this, we construct our training dataset of images from multiple sources. We use labeled examples of each class (as well as background images containing none of the classes) taken from the CIFAR-10 [19], CIFAR-100 [19], and MIO-TCD [27] datasets as well as a set of publicly available surveillance videos from static webcams that we collected and annotated. More details about the data collection is given in the supplemental material. In total we used 187,793 RGB training examples, resized to 64x64 pixels.

Neural Architecture: We use a modified ResNext-50 network [48] as our backbone architecture. We modified the original model by adding an extra fully connected layer that maps the 2048-dimensional feature vector after the average pooling layer to a 128-dimensional layer. The 128-dimensional layer is then mapped by a final fully connected layer to an 8-dimensional output layer with sigmoid activations that represent the categories. The extra fully connected layer gives us a 128-dimensional feature vector to represent appearance instead of the 2048-dimensional feature vector after RexNext-50's usual penultimate layer thus greatly reducing feature size. To train our model, we utilize the Binary Cross Entropy as our loss function.

Note that high-level features are usually distinctive even for unseen object classes despite the corresponding high-level attribute having low probabilities for all of the known object classes. This allows our appearance model to have unique representations for object classes other than the eight that we train on. See supplemental material for experiments on this.

3.1.2 Motion Model

To characterize the motion information for a given video volume, we train deep networks to estimate the following attributes directly from an RGB video volume: **(a)** histogram of optical flow (Y_{ang}), **(b)** a vector of the average speed of pixels in each direction of motion (Y_{speed}), **(c)** background classifier ($Y_{bkg.cls}$) and **(d)** percentage of stationary pixels ($Y_{bkg.pix}$). The histogram of optical flow consists of 12 bins each of which stores the fraction of pixels in the video volume that are estimated to be moving in one of the 30 degree directions of motion. The average speed vector consists of the average speed (in pixels per frame) of all pixels falling in each of the 12 histogram of flow bins. The background classifier classifies whether the video volume contains motion or not. The percentage of stationary pixels in a video volume gives the rough size of the moving objects in a video volume.

Motion Training Data: We use the set of surveillance videos mentioned above to learn motion attributes in a self-supervised way. For each video, we sample video volumes from regions with significant 'motion' as well as very little motion ('background'). We identify these regions by computing their pixelwise optical flow fields using the TV-L1 method [50], which is also used to automatically generate ground-truth motion attributes. (Note that optical flow is only used to create ground truth for training our motion models. It is not used in later stages.) In total we obtain $283,486$ 'background' video volumes and $2,551,376$ 'motion' video volumes. We use 90% of these for training our models and the remainder for validation.

The ground truth motion attributes for each training video volume are computed from the corresponding pixelwise flow fields as follows: We represent the $Y_{bkg.cls}$ attribute as a single binary variable denoting if a video volume is 'background'($Y_{bkg.cls} = 1$) or not ($Y_{bkg.cls} = 0$). The ground truth for Y_{ang} and Y_{bkg} are computed by first computing a 13-bin normalized histogram, wherein the first 12 bins represent the number of pixels with flow orientation in the ranges $[i * \pi/6 : (i + 1) * \pi/6)$ with $i \in [0, 11]$, while the last bin denotes the number of pixels with flow magnitude below threshold. The histogram is then normalized by the total number of pixels. The first 12 bins of this histogram are used as the ground truth for Y_{ang} and the 13^{th} bin is the ground truth for Y_{bkg}. Finally, we represent Y_{mag} as a 12-dimensional vector denoting the average flow magnitude for pixels in each of the 12 flow orientation ranges.

Learning Task and Neural Architecture: We treat each motion attribute independently and train separate models respectively. For each attribute prediction task, we use the same backbone architecture design, but train each model using different objective functions. Our model is a stack of 3D convolutions (3DConv) with batch normalization (BN) and ReLU. In total we have 3 layers of [3Dconv-BN-ReLU] followed by a fully connected layer. We provide additional details in the supplemental material.

We formulate $Y_{bkg.cls}$ attribute prediction as a standard binary classification task and train the model using cross-entropy loss function. For Y_{bkg} and Y_{mag} attribute prediction, we treat the learning task as a regression problem and train the model using mean squared error loss. And finally, for training the model to predict Y_{ang} attribute, we utilize KL Divergence loss. For all the tasks, we construct a sim-

ple light-weight CNN. The detailed configuration of our 3D CNN architecture is presented in the supplemental material.

3.2. Model Building

Once trained, the attribute deep nets are used to build a model of any scene given the nominal video. As illustrated in Figure 1, to process each nominal video, we slide a spatio-temporal window of dimension $[h \times w \times t]$ with spatial stride $(h/2, w/2)$ and temporal stride of t to construct video volumes. In the experiments, we select $h = w$ and choose h to be roughly the height in pixels of a person in a particular dataset. For each RGB video volume, we extract its features using the previously trained appearance net and four motion nets. To get a single appearance feature vector for a video volume, the feature vectors computed by the appearance network for each frame of the video volume are averaged. We concatenate the feature vectors from the penultimate layers of the appearance, angle, speed and background pixel nets along with the binary output of the background classifier net to create a combined feature vector. We use F to denote a combined feature vector and app, ang, mag, and bkg to denote the appearance, angle, magnitude and background pixel fraction feature vectors, each of size 1×128. Finally, cls denotes the binary background classification of size 1×1. F is of size 1×513.

After computing features, we use the exemplar selection approach of [18, 34] to create a region-specific compact model of the nominal data. For each region, we use the following greedy exemplar selection algorithm:

1. Add the first feature vector to the exemplar set.
2. For each subsequent feature vector, compute its distance to each feature vector in the exemplar set and add it to the exemplar set only if all distances are above a threshold, th.

To be clear, a separate set of exemplars is selected for each different spatial region in a scene. This allows our model to represent the fact that different object and motions occur in different regions of a scene. To compute the distance between two feature vectors $F_1 = [app_1; ang_1; mag_1; bkg_1; cls_1]$ and $F_2 = [app_2; ang_2; mag_2; bkg_2; cls_2]$ we use L_2 distances between corresponding components normalized by a constant to make the maximum distance for each component approximately 1. When a video volume does not contain motion (as determined by the background classification, cls), the motion component vectors are set to 0. The distance function can be written as follows:

$$d_A(F_1, F_2) = \|A_1 - A_2\|_2 \tag{1}$$

where $A \in \{app, ang, mag, bkg\}$,

$$d(F_1, F_2) = \frac{d_{app}}{Z_{app}} + \frac{d_{ang}}{Z_{ang}} + \frac{d_{bkg}}{Z_{bkg}} + \frac{d_{mag}}{Z_{mag}}. \tag{2}$$

The normalization factors, Z_{app}, Z_{ang}, Z_{mag} and Z_{bkg} are computed once by finding the max L_2 distances between a large set of feature vector components computed from a validation set (UCSD Ped1 and Ped2 in our experiments).

One big advantage of the exemplar learning approach is that updating the exemplar set in a streaming fashion is possible. This makes the approach scalable and adaptable to environmental changes over time.

3.3. Anomaly Detection

At test time, we process each test video in the same way (by sliding a $[h \times w \times t]$ spatio-temporal window with spatial stride $(h/2, w/2)$ and temporal stride of t) to generate video volumes. For each video volume, we compute the combined feature vector as before using the pre-trained nets. Each combined feature vector is compared with every exemplar for the corresponding region using the distance function in Equation 2. The anomaly score for the given test video volume is the minimum distance over the set of all exemplars from the same spatial region. A pixelwise anomaly score map is maintained by assigning the anomaly score to all pixels corresponding to every frame of the video volume. If a pixel has already been assigned an anomaly score (because of partially overlapping video volumes), then the maximum of the previous score and the current score is assigned. Figure 1 shows our anomaly detection pipeline.

3.4. A Note on Computational Efficiency

For both model building and anomaly detection, most of the time is spent computing feature vectors (forward passes of 5 networks). This is greatly sped up by testing whether a video volume is the same as the previous video volume in time. If there is no change then the anomaly score for the new video volume should be the same as the one before it and no computation of feature vectors is needed. This allows our method to run at 20 to 100 fps (dataset dependent). Details are in the supplemental material.

4. Experiments

4.1. Datasets and Evaluation Criteria

We experiment on five benchmark datasets: UCSD Ped1 and Ped2 [29], CUHK Avenue [24], Street Scene [33] and ShanghaiTech [26]. We use UCSD Ped1 and Ped2 with modified ground truth for parameter tuning and CUHK Avenue, Street Scene and ShanghaiTech for evaluation.

UCSD Ped1 & Ped2: UCSD Ped1 dataset contains 34 training videos and 36 test videos while UCSD Ped2 dataset contains 16 training videos and 12 test videos. Anomalies consists of bikers, skaters and cars in a pedestrian area.

CUHK Avenue: The Avenue [24] dataset contains 16 training videos with normal activity and 21 test videos. Examples of abnormal events in Avenue are related to people run-

ning, throwing objects or walking in wrong direction.

Street Scene: The Street Scene [33] dataset contains 46 training videos defining the normal events and 35 test videos. Prominent examples of anomalies include jaywalking, loitering and bikes or cars driving outside their lanes.

ShanghaiTech: The ShanghaiTech [26] dataset is a multi-scene benchmark for video anomaly detection. It consists of 330 training and 107 test videos. Major categories of anomalies include people fighting, stealing, chasing, jumping, and riding bikes or skating in pedestrian zones.

While our primary focus is on single scene video anomaly detection task, we consider the ShanghaiTech dataset only to highlight the ease of usability and robustness of our method to multi-scene benchmarks. Our method is applied to ShanghaiTech without modification even though the location-dependent aspect of our model is not necessary for a multi-scene dataset. Improvements in accuracy are likely if we specialize our model to multi-scene datasets.

Evaluation Criteria. We use the Region-Based Detection Criterion (RBDC) and the Track-Based Detection Criterion (TBDC) as proposed in [33] for quantitative evaluation of our framework. These criteria correctly measure the accuracy of spatially and temporally localizing anomalous regions (RBDC) and anomalous tracks (TBDC) versus false positive detections per frame. We report the area under the curve (AUC) for false positive rates per frame from 0 to 1 for each of these criteria. As pointed out in [33], frame-AUC [29] is not an appropriate evaluation metric for video anomaly detection methods that spatially localize anomalies. However, we report frame-AUC scores of our method for completeness and comparison with other older methods. We also do not use the pixel-level criterion [29] because of its serious flaws as mentioned in [33].

4.2. Implementation

Feature Learning. To train our **appearance model**, we use SGD with a 0.001 learning rate and 0.9 momentum and train for 50 epochs. The model with lowest classification error on the validation set is selected. For **motion models** we optimize with AdamW [23] with a 0.001 learning rate and train for 30 epochs. We select the best model for each attribute using the validation set.

Video volume parameters. We define the dimensions (w, h) of a video volume for each dataset so that h is roughly the height of a person in pixels and $w = h$. Specifically, for Ped1, Ped2, Avenue, Street Scene and ShanghaiTech, our region dimensions are $(32, 32)$, $(32, 32)$, $(128, 128)$, $(64, 64)$ and $(100, 100)$ respectively. Zero-padding was used for edge regions as needed. The number of frames in a video volume, t, is 10 for all datasets.

Parameter Tuning. To set a threshold th for exemplar selection without fitting to test data, a validation data set is needed. We chose Ped1 and Ped2 for this purpose, both because these data sets are performance-saturated, and because previous works [34] have identified inconsistencies in their ground truth. Specifically, ground-truth annotations of Ped1 and Ped2 do not label every location-specific anomaly. To rectify this, we augment the existing ground truth annotations to include all anomalies consistent with Definition 1. This is justified because we are using Ped1 and Ped2 to set our hyperparameters and not to compare against previous methods. Table 1 shows region-based and track-based AUC for different values of the threshold th used for exemplar selection for both Ped1 and Ped2. We see that the accuracy of our method is robust to large variations of th. However, larger values of th lead to smaller numbers of exemplars and thus smaller models of the nominal video which is desirable. We select $th = 1.5$ as a good trade-off between accuracy and model-size. We use this value on all datasets in our experiments. For Ped2, the average number of exemplars selected per region is about 13 ($\approx 0.5\%$ of the total number of video volumes in the nominal video). Exemplar selection typically finds tens to sometimes low hundreds of exemplars (for Street Scene) for regions with lots of activity. Regions with very little activity typically have only 1 or 2 exemplars. This leads to very compact models of the nominal video.

Th	UCSD Ped1			UCSD Ped2		
	RBDC	TBDC	NUM	RBDC	TBDC	NUM
3	36.866	77.83	288	64.808	89.13	350
2.5	49.36	89.43	424	78.813	93.716	761
2	57.524	89.6	944	84.66	95.97	1339
1.5	**61.65**	**88.9**	**4201**	**87.44**	**95.08**	**4470**
1	61.496	87.54	19926	87.408	95.776	19138
0.5	61.435	87.72	49113	87.195	95.12	34862
0.25	61.49	87.81	57636	87.199	95.127	45795

Table 1. RBDC and TBDC scores (in %) of our method for different thresholds (th) on UCSD Ped1 and Ped2. NUM denotes the total number of exemplars across all regions.

4.3. Quantitative Results

Tables 2 and 3 compare our method to other top methods on Avenue, ShanghaiTech and Street Scene. On Avenue, we improve over all previous methods for the region-based detection criterion (RBDC) and are second best for the track-based detection criterion (TBDC). On ShanghaiTech, we improve over the next best method for both RBDC and TBDC by significant margins. For the frame-level criterion which does not measure spatial localization we are in the middle of the pack compared to other recent methods for both Avenue and ShanghaiTech. On the difficult Street Scene dataset (Table 3), we improve the previous state of the art for both RBDC and TBDC, the latter by more than 11%. The good results across five different datasets (including Ped1 and Ped2) show the generality of the high-level features that we use in our models.

Method	Avenue			ShanghaiTech		
	RBDC	TBDC	Frame	RBDC	TBDC	Frame
Ionescu *et al.* [16]	15.77	27.07	87.4	20.65	44.54	78.7
Ramachandra *et al.* [33]	35.80	80.90	72.0	-	-	-
Ramachandra *et al.* [34]	41.20	78.60	87.2	-	-	-
Georgescu *et al.* [12]	57.00	58.30	91.5	42.80	83.90	**90.02**
Liu *et al.* [21]	19.59	56.01	85.1	17.03	54.23	72.8
Liu *et al.* [22]	41.05	86.18	89.9	44.41	83.86	74.2
Georgescu *et al.* [13]	65.05	66.85	**92.3**	41.34	78.79	82.7
Liu *et al.* [21] + Ristea *et al.* [38]	20.13	62.30	87.3	18.51	60.22	74.5
Liu *et al.* [22] + Ristea *et al.* [38]	62.27	**89.28**	90.9	**45.45**	**84.50**	75.5
Georgescu *et al.* [13] + Ristea *et al.* [38]	**65.99**	64.91	**92.9**	40.55	83.46	**83.6**
Our Method	**68.2**	**87.56**	86.02	**59.21**	**89.44**	76.63

Table 2. RRBDC, TBDC and Frame AUC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech datasets. The top score for each metric is highlighted in **red**, while the second best score is in **blue**.

Methods	RBDC	TBDC
Auto-encoder [14]	0.29	2.0
Dictionary method [24]	1.6	10.0
Flow baseline [33]	11.0	52.0
FG Baseline [33]	**21.0**	**53.0**
Our Method	**24.3**	**64.5**

Table 3. RBDC and TBDC AUC scores (in %) of various baseline methods on Street Scene dataset. The top score for each metric is highlighted in **red**, while the second best score is highlighted in **blue**.

4.4. Qualitative Results: Explainability

One of the big advantages of our method in addition to its accuracy is that it allows intuitive explanations of what the model has learned and why it labels a particular test video volume as anomalous or not. To visualize the feature vector representing a video volume, the appearance and motion components of the combined feature vector are mapped using the last fully connected layer of the respective network to the high-level appearance and motion attributes. We can then visualize these attributes as illustrated in Figure 2.

As an illustration of our model's explainability, the top of Figure 3 visualizes the exemplars learned from the nominal video for a spatial region in the middle of the street. Cars travel down and to the right in this lane of the street. The top six exemplars learned show mainly cars (or unknown objects, since video volumes containing only parts of cars are often not classified as cars) traveling down and to the right, as expected. There are also exemplars for stationary background, as well as stationary cars (since occasionally traffic stops on this part of the street). Thus, our learned model is understandable and consistent with what one expects. Furthermore, for a video volume containing a person jaywalking, the visualization in the bottom, left of the figure shows that our networks correctly identify it as containing a person walking mainly to the right at moderate speed. The closest exemplar to this test volume is an unknown object moving down and to the right which yields a high anomaly score of 2.08. (A threshold of 1.8 yields high detection rates with low false positive rates across all the datasets.) Thus, the explanation of this anomaly is that there is an unusual object (person) walking in an unusual direction.

Another example is shown for Ped2 in Figure 4. Here we analyze a region on the sidewalk. The exemplars learned for this region (shown at the top of Figure 4) are mainly background with very little movement or people moving mainly left or right at slow speeds. Some video volumes containing only parts of people are not classified as people which leads to exemplars of unknown objects moving left or right. Overall, these exemplars are again what we would expect for this region. For the test frame shown, a cyclist is riding on the sidewalk. The visualization of the video volume centered on that frame at that spatial region shows that it was classified as a cyclist traveling down and right at a high speed. These high-level attributes differs from the nearest exemplar in terms of its object class and speed and therefore leads to a high anomaly score.

As a final illustration (we show more in the supplemental material), we look at an example from CUHK Avenue in Figure 5. The top six exemplars for the region highlighted at the left of the figure show that the model has learned that this region contains either background with very little movement or else people or unknown objects moving mainly left or right slowly. For the anomalous test video volume shown containing a person running to the left, the high-level features estimate an unknown object moving left at high speed. Even though the object recognizer did not correctly predict that the video volume contains a person, the person class is the most likely out of the eight classes. The nearest exemplar is an unknown object (closest to a person class) moving slowly to the left. The main difference between the test video volume and the closest exemplar is the unusual speed which correctly explains this anomaly.

4.5. Ablation Study

We perform an ablation study on all the benchmarks to evaluate the benefit of each attribute in detecting anomalies. We consider features from each model separately, combining features from only motion models and finally our full model which uses all features. We accordingly change the distance function to compare features of two video volumes so that it only uses the provided features. We present our results in Table 4. We see that different attributes can be important for different types of scenarios. However, we get best results across all the benchmarks only when we combine all the motion and appearance features. This highlights the importance of modeling both appearance and different components of motion, especially to be able to predict anomalies under wide variety of scenarios.

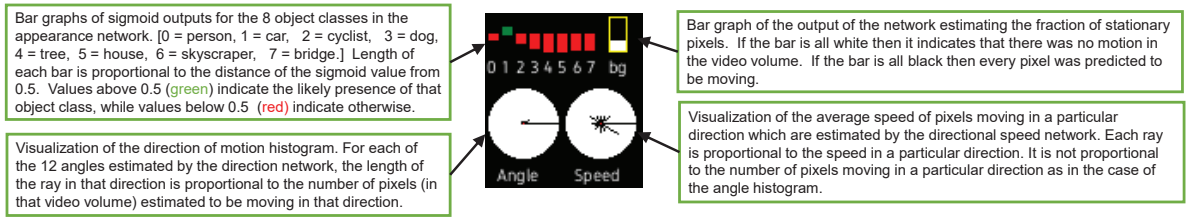
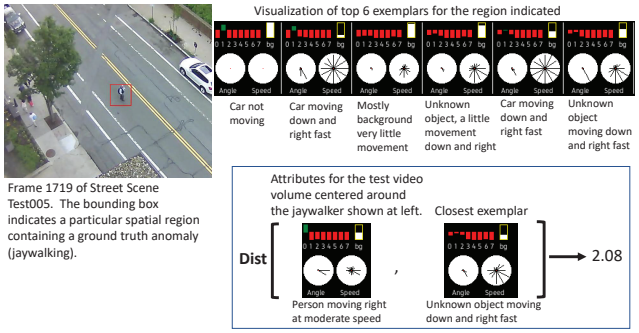
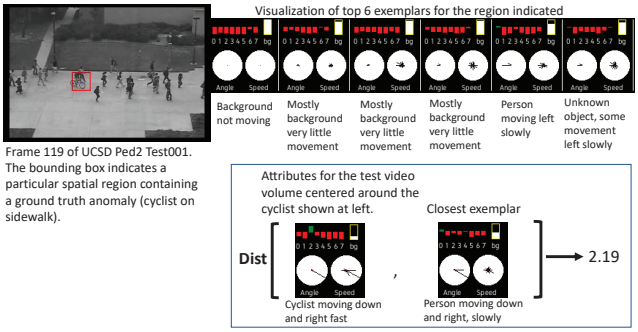


Figure 2. Explanation of our "instrument panel" showing the estimated attributes for a video volume. The interpretation of this visualization would be (roughly) a car (class 1) taking up most of the video volume, moving right at a high speed.



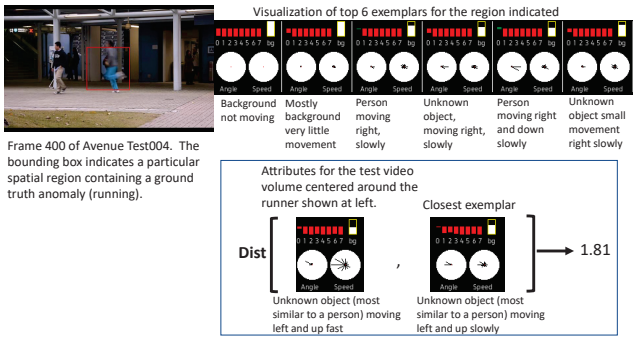
Frame 1719 of Street Scene Test005. The bounding box indicates a particular spatial region containing a ground truth anomaly (jaywalking).

Figure 3. 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.



Frame 119 of UCSD Ped2 Test001. The bounding box indicates a particular spatial region containing a ground truth anomaly (cyclist on sidewalk).

Figure 4. Visualization example for a region of UCSD Ped2 showing an explanation of the anomaly.



Frame 400 of Avenue Test004. The bounding box indicates a particular spatial region containing a ground truth anomaly (running).

Figure 5. Visualization example for a region of CUHK Avenue showing an explanation of the anomaly.

Attributes	Ped1	Ped2	Avenue	Street Scene
App	29.6 / 64.8	77.7 / 92.5	75.6 / 67.4	1.1 / 4.8
Motion	59.2 / 83.7	81.6 / 93.3	69.0 / 89.0	22.6 / 64.1
Angle	40.9 / 70.6	70.8 / 88.1	66.4 / 89.3	24.5 / 65.9
Mag	60.6 / 90.1	70.1 / 88.5	59.5 / 91.1	16.6 / 50.8
Bkg	45.5 / 86.4	71.7 / 95.2	49.9 / 81.5	11.9 / 49.8
App+Mot	61.7 / 88.9	87.4 / 97.1	69.6 / 86.7	23.9 / 65.3

Table 4. RBDC / TBDC AUC scores (in %) of our method when using only appearance, only motion (using angle, magnitude and background pixel predictions combined), each motion component separately and all the features

Attributes	Ped1	Ped2
ImageNet	25.148 / 44.63	64.67 / 83.17
Ours	29.6 / 64.8	77.7 / 92.5

Table 5. RBDC / TBDC AUC scores (in %) of our method when using our pre-trained model versus ImageNet pe-trained model as appearance feature extractor.

We further perform an ablation study on UCSD Ped1 and Ped2 datasets to empirically evaluate the benefit of our appearance model to represent object features versus using ImageNet pre-trained features. For the ImageNet pre-trained model, we use the pre-classification layer output of ResNext-50 as features. We present our results in Table 5. The superiority of our model is most likely due to the loss function used (binary cross-entropy) which allows an image patch to contain zero or multiple object classes.

5. Discussion and Conclusions

We have presented a novel method for explainable video anomaly localization that has a number of desired properties. Foremost, the method is accurate and general. We have shown that it works very well on five different datasets and, in particular, achieves state-of-the-art results on CUHK Avenue, Street Scene and ShanghaiTech. Setting it apart from almost all previous work, our model is understandable by humans and the decisions that our method makes are explainable. Finally, because our method does not require an expensive training phase on the nominal data, it is easy to expand our model when new nominal data is available.

References

[1] Detecting anomalous events in videos by learning deep representations of appearance and motion. *Comput. Vis. Image Underst.*, 156(C):117–127, mar 2017. 2

[2] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):555–560, 2008. 2

[3] Borislav Antić and Björn Ommer. Video parsing for abnormality detection. In *2011 International conference on computer vision*, pages 2415–2422. IEEE, 2011. 2

[4] Marcella Astrid, Muhammad Zaigham Zaheer, and Seung-Ik Lee. Synthetic temporal anomaly guided end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 207–214, 2021. 2

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 2

[6] Yunpeng Chang, Zhigang Tu, Wei Xie, and Junsong Yuan. Clustering driven deep autoencoder for video anomaly detection. In *European Conference on Computer Vision*, pages 329–345. Springer, 2020. 2

[7] Yang Cong, Junsong Yuan, and Ji Liu. Abnormal event detection in crowded scenes using sparse representation. *Pattern Recognition*, 46(7):1851–1864, 2013. 2

[8] Fei Dong, Yu Zhang, and Xiushan Nie. Dual discriminator generative adversarial network for video anomaly detection. *IEEE Access*, 8:88170–88176, 2020. 2

[9] Keval Doshi and Yasin Yilmaz. Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 934–935, 2020. 2

[10] Keval Doshi and Yasin Yilmaz. Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 254–255, 2020. 2

[11] Keval Doshi and Yasin Yilmaz. An efficient approach for anomaly detection in traffic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4236–4244, 2021. 2, 3

[12] 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. 2, 3, 7

[13] 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. 2, 7

[14] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016. 2, 7

[15] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE international conference on computer vision*, pages 3619–3627, 2017. 2, 3

[16] 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. 2, 7

[17] Radu Tudor Ionescu, Sorina Smeureanu, Marius Popescu, and Bogdan Alexe. Detecting abnormal events in video using narrowed normality clusters. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1951–1960. IEEE, 2019. 2, 3

[18] Michael Jones, Daniel Nikovski, Makoto Imamura, and Takahisa Hirata. Exemplar learning for extremely efficient anomaly detection in real-valued time series. *Data mining and knowledge discovery*, 30(6):1427–1454, 2016. 5

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

[20] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2013. 2

[21] 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. 2, 7

[22] 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. 2, 7

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[24] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013. 2, 5, 7

[25] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pages 125–141. Springer, 2020. 2

[26] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE international conference on computer vision*, pages 341–349, 2017. 2, 5, 6

[27] 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. 4

[28] Ke Ma, Michael Doescher, and Christopher Bodden. Anomaly detection in crowded scenes using dense trajectories. *University of Wisconsin-Madison*, 2015. 2

[29] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1975–1981. IEEE, 2010. 1, 5, 6

[30] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE conference on computer vision and pattern recognition*, pages 935–942. IEEE, 2009. 2

[31] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1273–1283, 2019. 2

[32] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020. 2

[33] Bharathkumar Ramachandra and Michael Jones. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2569–2578, 2020. 2, 5, 6, 7

[34] Bharathkumar Ramachandra, Michael Jones, and Ranga Vatsavai. Learning a distance function with a siamese network to localize anomalies in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2598–2607, 2020. 2, 5, 6, 7

[35] Bharathkumar Ramachandra, Michael Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[36] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, and Nicu Sebe. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1689–1698. IEEE, 2018. 2

[37] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE international conference on image processing (ICIP)*, pages 1577–1581. IEEE, 2017. 2

[38] Nicolae-Catalin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. *arXiv preprint arXiv:2111.09099*, 2021. 2, 7

[39] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017. 2

[40] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2112–2119. IEEE, 2012. 2

[41] Venkatesh Saligrama, Janusz Konrad, and Pierre-Marc Jodoin. Video anomaly identification. *IEEE Signal Processing Magazine*, 27(5):18–33, 2010. 1, 2

[42] Sorina Smeureanu, Radu Tudor Ionescu, Marius Popescu, and Bogdan Alexe. Deep appearance features for abnormal behavior detection in video. In *International Conference on Image Analysis and Processing*, pages 779–789. Springer, 2017. 2, 3

[43] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):747–757, 2000. 2

[44] Chenxu Wang, Yanxin Yao, and Han Yao. Video anomaly detection method based on future frame prediction and attention mechanism. In *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021. 2

[45] Lin Wang, Fuqiang Zhou, Zuoxin Li, Wangxia Zuo, and Haishu Tan. Abnormal event detection in videos using hybrid spatio-temporal autoencoder. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2276–2280. IEEE, 2018. 2

[46] Chongke Wu, Sicong Shao, Cihan Tunc, Pratik Satam, and Salim Hariri. An explainable and efficient deep learning framework for video anomaly detection. *Cluster Computing*, pages 1–23, 2021. 3

[47] Shandong Wu, Brian E Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2054–2060. IEEE, 2010. 2

[48] 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. 4

[49] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015. 2

[50] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 4