MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Perceptual Metric Learning for Video Anomaly Detection

Ramachandra, Bharathkumar; Jones, Michael J.; Vatsavai, Ranga

TR2021-028 April 17, 2021

Abstract

This work introduces a new approach to localize anomalies in surveillance video. The main novelty is the idea of using a Siamese convolutional neural network (CNN) to learn a metric between a pair of video patches (spatio-temporal regions of video). The learned metric, which is not specific to the target video, is used to measure the perceptual distance between each video patch in the testing video and the video patches found in normal training video. If a testing video patch is far from all normal video patches then it must be anomalous. We further generalize the approach from operating on video patches from a fixed grid to arbitrary-sized region proposals. We compare our approaches to previously published algorithms using 4 evaluation measures and 3 challenging target benchmark datasets. Experiments show that our approaches either surpass or perform comparably to current state-of-the-art methods whilst enjoying other favorable properties.

Machine Vision and Applications

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Perceptual Metric Learning for Video Anomaly Detection

Bharathkumar Ramachandra · Michael Jones · Ranga Raju Vatsavai

Abstract This work introduces a new approach to localize anomalies in surveillance video. The main novelty is the idea of using a Siamese convolutional neural network (CNN) to learn a metric between a pair of video patches (spatio-temporal regions of video). The learned metric, which is not specific to the target video, is used to measure the perceptual distance between each video patch in the testing video and the video patches found in normal training video. If a testing video patch is far from all normal video patches then it must be anomalous. We further generalize the approach from operating on video patches from a fixed grid to arbitrary-sized region proposals. We compare our approaches to previously published algorithms using 4 evaluation measures and 3 challenging target benchmark datasets. Experiments show that our approaches either surpass or perform comparably to current state-of-the-art methods whilst enjoying other favorable properties.

Keywords video anomaly detection · metric learning · video surveillance · Siamese neural networks

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

B. Ramachandra
Department of Computer Science
North Carolina State University
890 Oval Dr, Box 8206. Raleigh, NC 27695, USA.
E-mail: bramach2@ncsu.edu

M. Jones (corresponding)
Mitsubishi Electric Research Laboratories
201 Broadway, 8th floor. Cambridge, MA 02139, USA.
E-mail: mjones@merl.com

R. R. Vatsavai
Department of Computer Science
North Carolina State University
890 Oval Dr, Box 8206. Raleigh, NC 27695, USA.
E-mail: rrvatsav@ncsu.edu

1 Introduction

Video anomaly detection is the task of localizing (spatially and temporally) anomalies in videos, where anomalies refer simply to unusual activity. Unusual activity is scene dependent; what is unusual in one scene may be normal in another. In order to define what is normal, video of normal activity from the scene is provided. In the formulation of video anomaly detection that we focus on in this paper, we assume both the normal training video as well as the testing video come from the same single fixed camera, the most common surveillance setting. In this application, normal video (i.e. not containing any anomalies) is simple to gather while anomalous video is not. This is why it makes sense to provide normal video (and only normal video) for training. Given this formulation, the problem becomes one of building a model of normal activity from the normal training video and then detecting large deviations from the model in testing video of the same scene as anomalous. The reader is directed to [5, 35, 39, 42, 43, 55] for surveys on anomaly detection in videos.

Most previous methods have limitations that can be attributed to one or more of the following, which serve as the motivation for our approach: (1) The features used in many methods are hand-crafted. Examples include spatio-temporal gradients [30], dynamic textures [32, 50], histogram of gradients [14], histogram of flows [7, 14, 42], flow fields [1–3, 33, 52] and foreground masks [37]. (2) Almost every method requires a computationally expensive model building phase requiring expert knowledge which may not be practical for real applications. (3) Many previous works focus on detecting only specific deviations from normality as anomalous.

To overcome these limitations, we propose an exemplar-based nearest neighbor approach to video

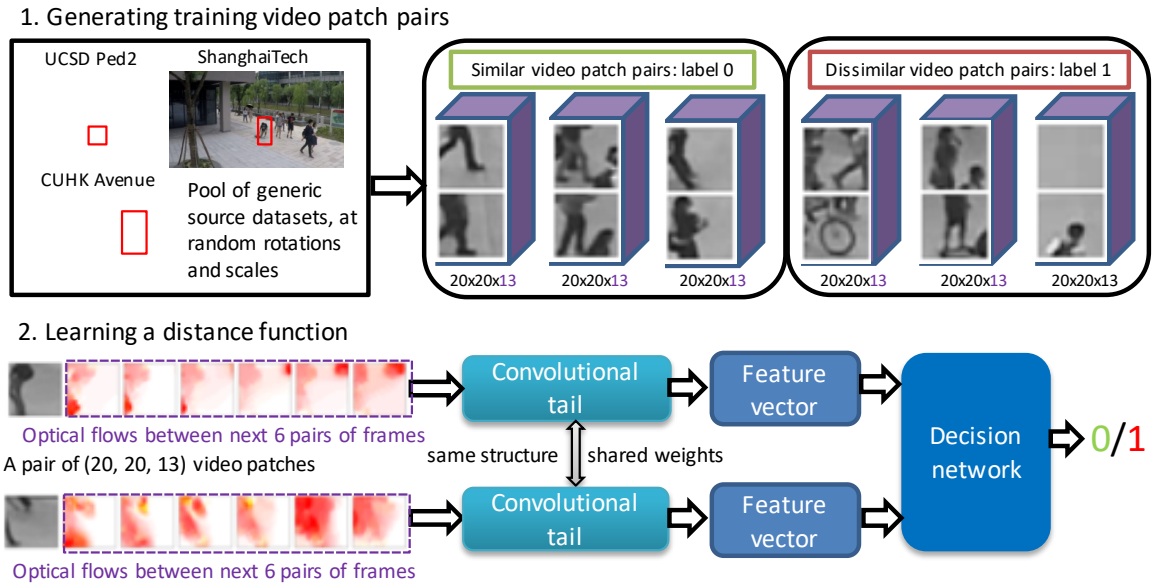

Fig. 1 An illustration of the scenario where UCSD Ped2, ShanghaiTech and CUHK Avenue are used as source datasets to learn a distance function from. Best viewed in electronic form in color.

anomaly detection that uses a distance function learned by a Siamese CNN to measure how similar activity in testing video is to normal activity. Our approach builds on the work of [37], in which normal video is used to create a model of normal activity consisting of a set of exemplars for each spatial region of the video. An exemplar is a feature vector representing a video patch, i.e., a spatio-temporal block of video of fixed size $H \times W \times T$ where H, W and T are the height, width and temporal depth of a video patch [9]. The exemplars for a spatial region of video represent all of the *unique* video patches that occur in the normal video in that region. Exemplars are region-specific because of the simple fact that anomalies are region-specific. To detect anomalies, video patches from a particular spatial region in testing video are compared to the exemplars for that region, and the anomaly score is the distance to the nearest exemplar. If a testing video patch is dissimilar to every exemplar video patch, then it is anomalous. In [37], hand-crafted features (either foreground masks or flow fields) were used to represent video patches and a predefined distance function (either L_2 or normalized L_1) was used to compute distances between feature vectors. We propose *learning* a better feature vector and distance function by training a Siamese CNN to measure the perceptual distance between pairs of video patches. By *perceptual* distance, we mean a distance that corresponds not to some straightforward distance between pixel values but rather to a human-like perception of whether two video patches are similar or not. In particular, two video patches with the same number and types of objects and similar motion should have a small distance. Our CNN is not specific to a particular scene, but is trained from video patches from several different source video anomaly detection datasets. This idea is also similar in spirit to past work on learning a CNN for matching patches [13, 54], except extended to video. In our approach, the training split consists of video patch pairs from source datasets and the test split contains video patch pairs from the target dataset, but each of these datasets also contains a training (normal) and test (contains anomalies) split. In order to avoid overload of the commonly used term "split", we henceforth refer to splits in the latter sense as "partitions".

Experiments show that our method either surpasses or performs comparably to the current state-of-the-art on the UCSD Ped1, Ped2 [32] and CUHK Avenue [30] test sets. We further extend this approach from operating on fixed-size video patches extracted from a fixed grid on the camera frame to arbitrary-sized video patches from unsupervised region proposals, while retaining the performance gains observed.

In summary, our major contributions are:

1. Our approach transforms the problem of training a CNN to classify video patches as normal or anomalous (which cannot be done since we have no anomalous training examples) to the problem of training a CNN that computes the perceptual distance between two video patches (a problem for which we can generate plenty of examples). We use the *same parameters for training the CNN from source datasets regardless of the target dataset.*

2. This approach allows task-specific feature learning, allows for efficient exemplar model building from normal video and detects a wide variety of deviations from normality as anomalous.

3. By shifting the complexity of the problem to the distance function learning task, the simple 1-NN distance-to-exemplar anomaly detection becomes highly interpretable. To the best of our knowledge, our work is the first to take this approach to anomaly detection.

4. In order to remove the dependence of choice of source datasets for a particular target dataset, we generalize our approach to operate on arbitrary-sized region proposals without loss in performance.

2 Related Work

We focus here on video anomaly detection methods that follow the formulation of the problem outlined previously. A number of methods such as [8, 19, 28, 46] use other formulations of the video anomaly detection problem which we do not discuss here, although we organize this section similar to [46].

2.1 Distance-based approaches

Distance-based approaches involve creating a model from a training partition and measuring deviations from this model to determine anomaly scores in the test partition.

The authors in [42] use the insight that 'optimal decision rules to determine local anomalies are local irrespective of normal behavior exhibiting statistical dependencies at the global scale' to collapse the large ambient data dimension. They propose local nearest neighbor based statistics to approximate these optimal decision rules to detect anomalies.

In [53], stacked denoising auto-encoders are used to learn both appearance and motion representations of video patches which are used with one-class SVMs to perform anomaly detection.

The authors in [41] derive an anomaly score map by consolidating the change in image features from a pre-trained CNN over the length of a video block.

[20] also uses object-centric processing by extracting proposals from a single-shot detector, but in a pipeline involving latent code learning using convolutional auto-encoders followed by one-versus-rest SVM anomaly scoring scheme.

2.2 Probabilistic approaches

Probabilistic approaches are similar to distance-based approaches, except that the model has a probabilistic interpretation, for example as a probabilistic graphical model or a high-dimensional probability distribution.

The authors in [1] use multiple fixed-location monitors to extract optical flow fields and compute the likelihood of an observation given the distribution stored in that monitor's buffer.

In [32], the authors propose a representation comprising a mixture of dynamic textures (MDT), modeling a generative process for MDTs and discriminant saliency hypothesis test for anomaly detection. In [50], they build off the MDT representation to detect anomalies at multiple scales in a conditional random field framework.

Authors in [2] contend that anomaly detection should try to "explain away" the normality in the test data using information learned from the training data. To this end, they use foreground object hypotheses and take a video parsing approach, treating those object hypotheses at test time which are necessary to explain the foreground but not explained by the exemplar training hypotheses are anomalous. In [3], they further build on this idea by extending the atomic unit of processing from an image patch to a video pipe.

2.3 Reconstruction approaches

Reconstruction approaches aim to break down inputs into their common constituent pieces and put them back together to reconstruct the input, minimizing "reconstruction error".

[6, 14, 27, 40] are examples of methods that use this approach. In our experience, reconstruction based approaches seem to be naively biased against reconstructing faster motion, for the simple reason that absence of motion is much more common and easier to reconstruct.

A subset of reconstruction approaches, **sparse reconstruction approaches** have an additional constraint in that the reconstruction must be minimialistic, that is, using only a few essential features from a dictionary to perform the reconstruction. [7, 30, 31] are examples of methods that use this approach.

Many of the methods mentioned above use deep networks. All of the previous papers that use deep networks for video anomaly detection that we are aware of use them in one of two techniques: (1) either to provide higher level features to represent video frames or (2) to learn to reconstruct only normal video frames. Much of the previous work builds on the basic idea of using a CNN, either pre-trained on image classification or other

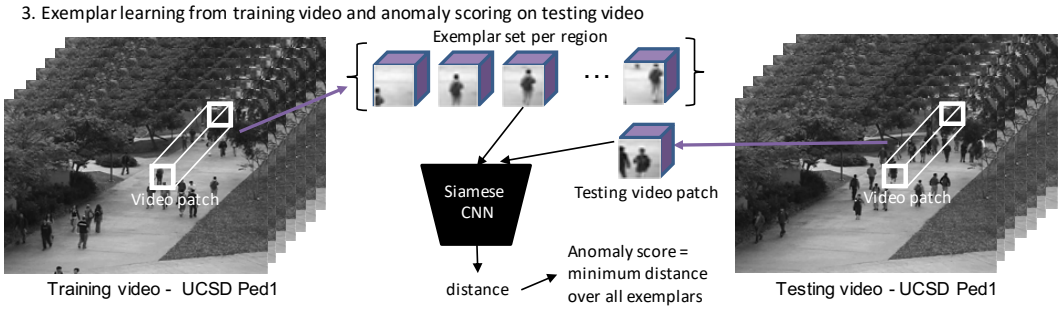

Fig. 2 An illustration of using the learned distance function to perform exemplar extraction and anomaly scoring on the target UCSD Ped1 dataset. Best viewed in electronic form in color.

tasks [16, 31, 41, 44] or trained on the training partitions of each video anomaly detection dataset [53], to provide a feature vector for representing video frames. The CNN feature maps provide higher level features than raw pixels. The other major theme of deep network approaches is to learn an auto-encoder [6, 14] or generative adversarial network [27, 40] to learn to reconstruct or predict only normal video frames. Reconstruction error is then used as an anomaly score. Our method follows neither of these previous techniques and instead presents a new way to take advantage of the power of deep networks for video anomaly detection. Namely, we use a CNN to learn a distance function between pairs of video patches. Thus, ours is a novel distance-based approach.

3 Method

By building on the exemplar-based nearest neighbor approach of [37], our main problem is to learn a distance function for comparing video patches from testing video to exemplar video patches that represent all of the unique video patches found in the normal video. To do this we use a Siamese network (see Figure 1) similar to the one first introduced by Bromley and LeCun [4]. In essence, by making the anomaly detection task itself a rather simple nearest neighbor distance computation (see Figure 2), we seek to offload the burden of modeling the complexity in this problem to the task of learning a distance function. This learning problem can be done offline and has a large amount of training data available from source datasets. Ideally this can be done once and the resulting feature representation and distance function used on a wide variety of different target datasets.

Of course, the more the statistics of the target dataset match the source datasets, the more suitable the learned distance function becomes for the anomaly detection task. This problem is called *dataset shift* and [36] is a comprehensive resource on the dataset shift problem and how to minimize its effects. We use some simple steps such as data augmentation and estimating class priors to deal with the problem here.

In this section, we go into more detail in each of the steps shown in Figures 1 and 2, provide justifications for our design decisions and setup some language essential for the Experiments section.

3.1 Generating training video patch pairs

The main difficulty with training a Siamese network to estimate the distance between a pair of video patches is determining how to generate the training set of similar and dissimilar video patch pairs. One training example consists of a pair of video patches plus a binary label indicating whether the two video patches are similar or dissimilar (see Figure 1 part 1). Video patch pairs should be selected to correctly correspond to their ground truth labels of "similar" or "dissimilar". Pairs should also be picked such that coverage of the possible domain of inputs to the CNN during test time is high. This is to ensure that the CNN is not asked to operate on out-of-domain inputs at test time.

How can we determine whether two video patches are similar or dissimilar and how can we select a varied set of video patch pairs that are relevant to video anomaly detection? An important insight is that we can use existing video anomaly detection datasets to do this. We use a source set of labeled video anomaly detection datasets to generate similar and dissimilar video patch pairs. The labeled datasets used to generate training examples should of course be disjoint from the target video anomaly detection dataset on which testing will eventually be done. The basic insight is as follows: for each source dataset,

(1) A non-anomalous video patch from the test partition is similar to at least one video patch from the same spatial region in the train partition. If it were not

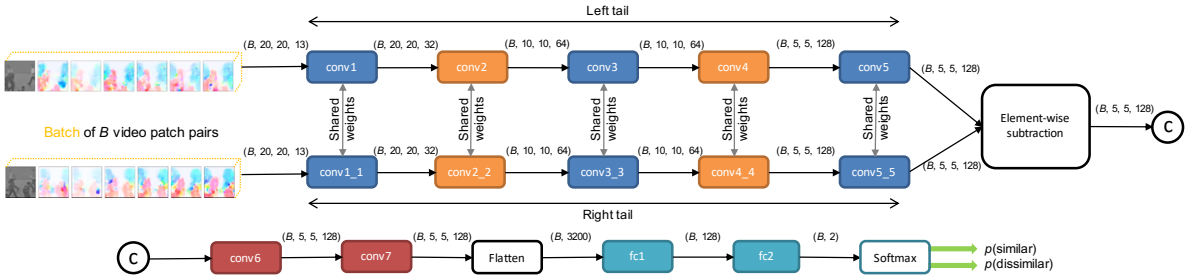

Fig. 3 Architecture of the Siamese neural network that learns a distance function between video patches. Best viewed in electronic form in color; color coding denotes unique structure.

similar to any normal video patches it would be anomalous.

(2) An anomalous video patch from the test partition is dissimilar to all possible patches from the same spatial region in the train partition. Moreover, it is dissimilar to even the most similar video patch.

The first rule generates a single pair for each normal video patch in a test video, although since there are many normal video patches in any test video, this rule can generate many similar pairs. The second rule generates many different dissimilar pairs for each anomalous video patch in a test video. The first rule requires a distance function to find the most similar train video patch to a test video patch. It is also useful in the second rule to have a distance function to know which dissimilar pairs are the most difficult (i.e. similar) since these are the most useful for training. We use a simple normalized L1 distance as our distance function along with the representation of video patches described in Section 3.2.

A reasonable concern about using a predefined distance function to help select training examples is that the Siamese network might simply learn this distance function. This does not happen for a few reasons. One is that the label for each example pair is not the L1 distance, but rather a 0 or 1 indicating whether the pair is similar or dissimilar, respectively. Secondly, it is possible for the L1 distance between two similar pairs to be larger than the L1 distance between two dissimilar pairs.

One important point to note is that normalized L1 distance is far from ideal to measure distance between video patches. For example, this distance does not take into account many variations in natural images such as scale, illumination and pose of objects. Because these variations mostly exist *across different regions* in the camera's field of view, we determine an adaptive threshold on normalized L1 distance below which to perform these pairings. The threshold for a region is determined by taking into account the above rules in combination with inspecting the distribution of nearest neighbor distances in a given region. Specifically, an adaptive threshold for a given region in the camera frame is determined simply as $\mu + \alpha * \sigma$ where μ is the mean of nearest neighbor distances between testing video patches and training video patches, σ is the corresponding standard deviation and α is determined by identifying an elbow in the distribution of nearest neighbor distances (we set it to 0.2 consistently in experiments). The adaptive threshold is common across the source datasets but different for similar and dissimilar pairs. Notice that dissimilar pairs that have large distances are more likely to be easy to discriminate for the Siamese network; on the other hand, we require some of these pairings despite this property to achieve high domain converage. Thus, we include candidate pairs with probability inversely proportional to the distance between them, achieving high domain coverage, but also a sufficient number of examples close to the decision boundary. We also include as similar pairs random video patches paired with slightly augmented (random translation and/or central scaling) versions of them. Our final video patch pair dataset consists of an equal number of similar and dissimilar pairs from each source dataset. For all experiments, we fix this number to 25,000.

3.2 Learning a distance function

Choice of representation: At this point, it is important to choose how video patches are represented, such that the learned distance function will perform well in the anomaly detection task. Our choice of representation consists of a $H \times W \times C$ cuboid. In light of all anomalies being appearance or motion based, we adopt a multi-modal representation. In all our experiments that follow, the first channel is a grayscale image patch and the next 12 channels are image patches from absolute values of x and y directional gradients of dense optical flow fields (we use [25]) between the subsequent 6 pairs of image patches. This sets $C = 13$ and we set

$H = 20$ and $W = 20$ for all experiments. See Figure 1 (part 2) for an illustration.

Pre-processing: Data augmentation of a random amount is performed on every video patch pair x_1, x_2 during training in order to improve the robustness of the learned distance function to these variations. The data augmentation involves randomly flipping left to right, centrally scaling in [0.7, 1] and brightness jittering of the first channel in [-0.2, 0.2] in a stochastic manner on both video patches in a pair. Pre-processing also involves linearly scaling intensity values of each video patch from [0, 255] to [-1, 1].

Network architecture and training: Figure 3 outlines our network architecture. Each video patch in a pair is first processed independently using conv-relu-batchnorm operations with 2×2 max-pooling after every other convolution in what we call convolutional twin "tails". Weight tying between the tails guarantees that two extremely similar video patches could not possibly have very different intermediate representations because each tail computes the same function. Finally, flattened feature vectors from the two twin tails (conv5, conv5_5) are subtracted element-wise and processed consequently in a typical classification pipeline minimizing a cross-entropy loss. All convolutions use 3×3 filters with a stride of 1. We find that subtracting the feature maps at conv5 produces faster optimization when compared to concatenation. We think this is because element-wise subtraction induces a stronger structural prior on the network architecture. Let B represent minibatch size, where i indexes the minibatch and $\mathbf{y}(x_1^{(i)}, x_2^{(i)})$ be a length-B vector which contains the labels for the mini-batch, where we assume $y(x_1^{(i)}, x_2^{(i)}) = 0$ whenever x_1 and x_2 are similar video patches and $y(x_1^{(i)}, x_2^{(i)}) = 1$ otherwise. The cross-entropy loss is of the form:

$$\begin{aligned}\mathcal{L}(x_1^{(i)}, x_2^{(i)}) = &-\gamma * \mathbf{y}(x_1^{(i)}, x_2^{(i)}) \log \mathbf{p}(x_1^{(i)}, x_2^{(i)}) \\ &-(1 - \mathbf{y}(x_1^{(i)}, x_2^{(i)})) \log (1 - \mathbf{p}(x_1^{(i)}, x_2^{(i)}))\end{aligned} \tag{1}$$

where $\mathbf{p}(x_1^{(i)}, x_2^{(i)})$ is the probability of the patches being dissimilar as output by the softmax function. Note that in the loss, we set class weight for the dissimilar class γ as 0.2 to penalize incorrectly classified dissimilar pairs less than incorrectly classified similar pairs. This further serves our objective at the anomaly detection phase to have low false positive rates at high true positive rates (where anomalies are denoted positive class). For training, the objective is combined with the standard backpropagation algorithm with the Adam optimizer [23], saving the best network weights by testing on the validation set (a set of held-out training examples) periodically. The gradient is additive across the twin tails due to tied weights. We use a batch size of 128 with an initial learning rate of 0.001 and train for a maximum of 500 iterations. Xavier-Glorot weight initialization [12] sampling from a normal distribution is used in tandem with ReLU activations in all layers. One important point to note is that, rather than save the network weights that maximize validation accuracy or minimize validation loss, we save that which maximizes validation area under the receiver operating characteristic curve (AUC) for false positive rates up to 0.3. This ROC curve is obtained by plotting true positive rate as a function of false positive rate, where the dissimilar class is denoted positive. By maximizing this AUC, the network that *orders distances in a way that achieves high true positive rate at low false positive rates* is preferred, the behavior we would like to see when it comes time for the anomaly detection phase. We use label smoothing regularization [47] set to 0.1 to aid generalization. We find that adding label smoothing regularization is helpful for two reasons. The first is that the video patch pairing process has to in a sense guess what a *future learned function* should call similar and different in order to achieve good performance on anomaly detection, so it produces a dataset with noisy labels. The second arises from the observation that minimizing the cross entropy is equivalent to maximizing the log-likelihood of the correct label, which makes the network try to increase the logit corresponding to the correct label and make it much larger than the other logits, causing it to overfit to the training data and become too confident about its predictions. Label smoothing helps with both of these by making the network less confident about its predictions. We also use dropout [45] of 0.3 on the activations of the second to last fully connected layer (fc1).

3.3 Exemplar learning and anomaly detection on target dataset

Detecting anomalies on a target dataset involves two stages: exemplar model building using the train partition of the dataset and anomaly detection on the test partition. Both stages use the previously trained Siamese network to measure distance between video patches. This is done by simply treating the softmax of the logit value that corresponds to the video patches being different as a measure of distance between the patches. Because the softmax output can also be interpreted as a probability, the distance measured can also be interpreted as the *probability of patches being different*. We emphasize that the training of the Siamese network is independent of the exemplar model building and anomaly detection stages. The Siamese network is

Table 1 Traditional frame-level and pixel-level evaluation criteria on the UCSD Ped1, UCSD Ped2 and CUHK Avenue benchmark datasets from related literature, ordered chronologically, complied from this same list. Our approach either surpasses or performs comparably on these evaluation criteria when compared to previous methods. *Some of the earlier works unfortunately use only a partially annotated subset available at the time to report performance.

Method	**UCSD Ped1 frame AUC/EER**	**UCSD Ped1 pixel AUC***	**UCSD Ped2 frame AUC/EER**	**UCSD Ped2 pixel AUC**	**CUHK Avenue frame AUC/EER**
Adam [1]	65.0%/38.0%	46.1%	63.0%/42.0%	18.0%	-/-
Social force [33]	67.5%/31.0%	19.7%	63.0%/42.0%	21.0%	-/-
MPPCA [32]	59.0%/40.0%	20.5%	77.0%/30.0%	14.0%	-/-
Social force + MPPCA [32]	67.0%/32.0%	21.3%	71.0%/36.0%	21.0%	-/-
MDT [32]	81.8%/25.0%	44.1%	85.0%/25.0%	44.0%	-/-
AMDN [53]	92.1%/16.0%	67.2%	90.8%/17.0%	-	-/-
Video parsing [2]	91.0%/18.0%	83.6%	92.0%/14.0%	76.0%	-/-
Local statistical aggregates [42]	92.7%/16.0%	-	-/-	-	-/-
Detection at 150 FPS [30]	91.8%/15.0%	63.8%	-/-	-	-/-
Sparse reconstruction [7]	86.0%/19.0%	45.3%	-/-	-	-/-
HMDT CRF [50]	-/17.8%	82.7%	-/18.5%	-	-/-
ST video parsing [3]	93.9%/12.9%	**84.2%**	94.6%/**10.6%**	81.1%	-/-
Conv-AE [14]	81.0%27.9%	-	90.0%/21.7%	-	70.2%/25.1%
Deep event models [10]	92.5%/15.1%	69.9%	-/-	-	-/-
Compact feature sets [24]	82.0%/21.1%	57.0%	84.0%/19.2%	-	-/-
Convex polytope ensembles [48]	78.2%/24.0%	62.2%	80.7%/19.0%	-	-/-
Joint detection and recounting [16]	-/-	-	92.2%/13.9%	89.1%	-/-
Sparse coding revisit [31]	-/-	-	92.2%/-	-	81.7%/-
GAN [40]	**97.4%/8.0%**	70.3%	93.5%/14.0%	-	-/-
Future frame prediction [27]	83.1%/-	-	95.4%/-	-	85.1%/-
Plug and play CNN [41]	95.7%/**8.0%**	64.5%	88.4%/18.0%	-	-/-
Narrowed normality clusters [21]	-/-	-	-/-	-	88.9%/-
Object-centric auto-encoders [20]	-/-	-	**97.8%**/-	-	**90.4**%/-
NN on video patch FG masks [37]	77.3%/25.9%	69.3%	88.3%/18.9%	83.9%	72.0%/33.0%
Ours	86.0%/23.3%	80.4%	94.0%/14.1%	**93.0%**	87.2%/**18.8%**

trained on a different set of source datasets than the target video anomaly detection dataset.

Exemplar learning on train partition of target dataset: Since videos contain a large amount of temporal redundancies, we use the exemplar learning approach of [22] to build a model of normal activity in the target dataset. The exemplar model consists of sets of region-specific exemplar video patches from the videos in the train partition using a sliding spatio-temporal window with spatial stride ($H/2$, $W/2$) and temporal stride of 1. The point of exemplar learning is to represent the set of all video patches in the train partition using a smaller set of unique, representative video patches. The feature vector learned by the Siamese network is used to represent a video patch and the distance function learned by the Siamese network measures the distance between two feature vectors. A video patch is added to the exemplar set for a particular spatial region if its distance to the nearest exemplar for that region is above a threshold, which we set to 0.3 for all experiments. Figure 2 illustrates a subset of exemplar video patches extracted from one region of the camera's field of view in the UCSD Ped1 dataset by our CNN. 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.

Anomaly detection on test partition of target dataset: At test time, overlapping patches with spatial stride ($H/2$, $W/2$) and temporal stride of 1 are extracted from the test partition and distances to nearest exemplars produce anomaly scores (see Figure 2). In both the exemplar learning and anomaly scoring phases, we achieve additional speedup by ignoring video patches that contain little or no motion. Specifically, a video patch is ignored if under 20% of its pixels across the channel dimension do not satisfy a threshold on flow magnitude or a threshold on the raw pixel value difference between the current and the previous frame. Furthermore, the brute-force nearest neighbor search used in the experiments could be replaced by a fast approximate nearest neighbors algorithm [34] for further speed-up. Anomaly scores are stored and aggregated in a pixel map and the final anomaly score of a pixel is simply the mean of all anomaly scores it received as part of patches it participated in (due to overlap of patches in space and time). The anomaly detection is region-specific, so a patch is only compared to exemplars extracted from the same region.

4 Experiments

4.1 Experimental setup - Datasets and evaluation measures

Datasets: We perform experiments on 3 benchmark datasets: UCSD Ped1, UCSD Ped2 [32] and CUHK Avenue [30]. Each of these datasets includes predefined train and test partitions from a single static camera where train partitions contain sequences of normal activity only and test partitions contain sequences with both normal and anomalous activity, and with spatial anomaly annotations per frame.

Evaluation measures: To compare against other works we use the widely-used **frame-level** and **pixel-level** area under the curve (AUC) and equal error rate (EER) criteria proposed in [32].

In addition, we report performance using two new criteria presented in [37], which are more representative of real-world performance as argued in that paper. The first is a **region-based criterion**: A true positive occurs if a ground truth annotated region has a minimum intersection over union (IOU) of 0.1 with a detection region. Detected regions are formed as connected components of detected pixels. The total number of positives is correspondingly the total number of anomalous regions in the test data. A false positive occurs if a detected region simply does not satisfy the minimum IOU threshold of 0.1 with any ground truth region. The region-based ROC curve plots the true positive rate (which is the fraction of ground truth anomalous *regions* detected) versus the false positive rate per frame. The second is a **track-based criterion**: A true positive occurs if at least 10% of the frames comprising a ground truth anomaly's track satisfy the region-based criterion. The total number of positives is the number of ground truth annotated tracks in the test data. False positives are counted identically to the region-based criterion. The track-based ROC curve plots the true positive rate (which is the fraction of ground truth anomalous *tracks* detected) versus the false positive rate per frame. AUCs for both criteria are calculated for false positive rates from 0.0 up to 1.0. Because the track-based criterion requires ground truth annotations to have a track ID, we relabeled the Ped1, Ped2, and Avenue test sets with bounding boxes that include a track ID. These new labels will be made publicly available. Old labels are used for the frame and pixel-level criteria.

4.2 Comparison against state of the art

To evaluate our approach, we compare against results reported on the traditional evaluation measures by papers in the recent literature. For each of our experiments, a new CNN was trained using only datasets other than the target dataset to curate the training data for the Siamese network (see Table 2), but each newly trained network used the same aforementioned regularization parameters. A simple heuristic was used to choose which source datasets should be used for a given target dataset - those datasets in which the scale of objects roughly match that in the target dataset for a $H \times W$ image patch. In future work, we plan to use more labeled videos to train a single Siamese network that works well across many different target datasets.

Table 2 Source dataset configuration for each target dataset.

Target dataset	Source datasets
UCSD Ped1	Shanghai Tech camera 06 (quarterscale), Shanghai Tech camera 10 (quarterscale), UCSD Ped2 (halfscale, rotated at 45 degrees), CUHK Avenue (quarterscale)
UCSD Ped2	UCSD Ped1, CUHK Avenue (halfscale)
CUHK Avenue (halfscale)	UCSD Ped1, UCSD Ped2

Table 1 presents frame and pixel-level AUC measures on the UCSD Ped1, UCSD Ped2 and CUHK Avenue datasets. Our approach sets new state of the art on UCSD Ped2 pixel-level AUC by around 4% as well as on CUHK Avenue frame EER by around 6%. Upon visualizing the detections, we find that our approach finds it particularly difficult to detect anomalies at very small scales that exist in the UCSD Ped1 test set. Also, our method, like most others in Table 1, is unable to detect loitering anomalies present in the CUHK Avenue dataset. This is mainly due to our use of a "motion check" that ignores video patches with little or no motion for efficiency reasons. This could be replaced by a more sophisticated background model that is slower to absorb stationary objects.

A natural question one might ask is to do with what might happen if we used the Train partition of the target dataset to create additional similar video patch pairs and train the network. We added an auxiliary set of video patch pairs, created only from the Train partition of the target dataset, to the training data for the Siamese network and observed performance. Interestingly, we noticed that performance degrades slightly upon adding this auxiliary set for all 3 datasets. We think this might be because adding only similar pairs from the target dataset might send too strong of a signal such that the distance function learned judges pairs to be similar too easily, causing many false negatives (missed detections).

Further, we report AUC for false positive rates up to 1.0 for the track and region based criteria in Table 3. We reimplemented the work of [37] for these results.

Table 3 Track and region-based criteria, area under the curve for false positive rates up to 1.0.

Method	track AUC			region AUC		
	Ped1	**Ped2**	**Avenue**	**Ped1**	**Ped2**	**Avenue**
[37] (FG masks)	84.6%	80.5%	**80.9%**	46.6%	62.5%	35.8%
[37] (Flow)	86.5%	83.2%	78.4%	48.3%	55.0%	27.3%
Ours	**90.0%**	**89.3%**	78.6%	**59.2%**	**74.0%**	**41.2%**

Clearly, our approach surpasses that of [37], meaning we detect more anomalous events (tracks and regions) while also producing fewer false positives per frame overall.

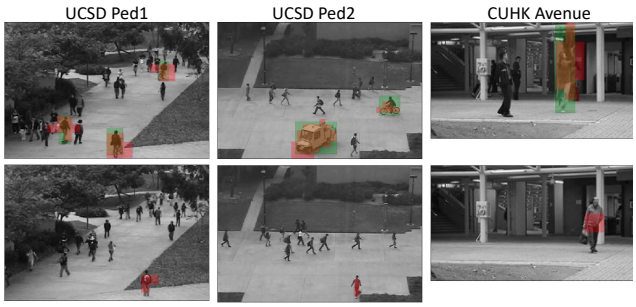

Fig. 4 Examples of true positives (first row) and false positives (second row) from our detector on all 3 datasets. Green bounding box annotations denote ground truth anomalies and red regions our model's detections (intersections are orangeish). Best viewed in electronic form in color.

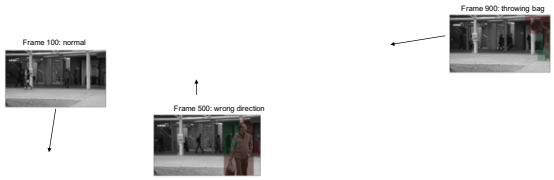

Fig. 5 Anomaly score as a function of frame number for CUHK Avenue Test sequence number 6. Green shading on the plot denotes ground truth anomalous frames. Best viewed in electronic form in color.

These ROC curves and AUC measures do not completely capture the behavior of video anomaly detection approaches. In [29], the authors present an excellent analysis of the problems with an evaluation measure such as AUC. Thus, we present a set of qualitative results here. Figure 4 shows some detection results at a fixed anomaly score threshold. We notice that the *quality of false positives* in our approach is high, and often we are able to attribute reasons for these errors. For example, the false positive shown in the figure for UCSD Ped1 dataset is due to the fact that a person is never seen walking across the grass in this specific manner in the train partition. A similar argument explains the false positives shown for the other two datasets as well. This could either indicate that the train partition is incomplete, or highlight the subjectivity involved in ground truth annotation processes. Figure 5 illustrates how anomaly score per frame, computed as the maximum of anomaly scores of pixels in the frame, varies for one test sequence of CUHK Avenue. The high variance in anomaly scores during the "bag throwing" anomaly even indicates how this event might intersperse normal and anomalous frames, seeming normal when the bag leaves the camera frame and vice versa.

Table 4 Ablation study on the choice of source datasets for a particular target dataset. 'Y' denotes that the dataset was used in the source pool.

Source datasets			Target = Ped2	
Ped1	**Avenue**	**ShanghaiTech**	**Frame AUC**	**Pixel AUC**
Y			90.9%	89.4%
	Y		90.4%	88.7%
		Y	93.7%	93.0%
Y	Y		94.0%	93.0%
	Y	Y	91.8%	91.0%
Y		Y	91.7%	90.7%
Y	Y	Y	93.0%	91.9%

4.3 Ablation study on source datasets used

We perform an ablation study to understand the effect of picking source datasets for a particular target dataset. Since it is prohibitive to perform a complete ablation study, for this study we set the target to be UCSD Ped2 and vary all non-empty subsets of source datasets from the set {UCSD Ped1, CUHK Avenue, ShanghaiTech (cameras 06 and 10)}, training only once. The results presented in Table 4 show that while *there is some sensitivity to the choice of source datasets*, on both the frame and pixel level measures, we see a variation of $< 5\%$. This variation is from a combination of variation due to stochasticity during training (batching, random initialization, dropout) and choice of source datasets.

Fig. 6 Examples of large prediction errors made by our model on UCSD Ped1. Classes 0 and 1 refer to similar and dissimilar pairs respectively. Best viewed in color.

5 Understanding the distance function learned

We also tried to gain some insight into what properties the distance function learned by the CNN possesses. To this end, we recorded the video patch pairs on which the CNN makes large errors, that is, either classifying similar pairs as dissimilar or vice versa, with high predicted probability. Figure 6 is a visualization of 4 such video patch pairs when the target dataset is UCSD Ped1. Remarkably, the CNN seems to find it hard to correctly classify examples that are conceivably hard for humans. Specifically, the dissimilar pairs that have been misclassified seem to contain a skateboarder moving only slightly faster than a pedestrian would, and the similar pairs that have been misclassified exhibit some distinct differences in their flow fields.

6 Going from fixed-size video patches from a fixed grid to unsupervised region proposals

We notice that the property that seems to be the biggest limitation of our previous approach ([38] denoted VADSiamese henceforth) is that it operates at a single scale by extracting video patches from a fixed grid on the camera frame. Although we present an ablation study on the effect of the choice of source datasets on performance and observe only small variance (see Section 4.3 and Table 4), the sensitivity is a cause for concern nonetheless. Specifically, the concern is that we still had to scale the source datasets to match the scale of objects for a particular target dataset (see scaling details in Table 2). This tuning is an imperfect process, especially when object scale within a dataset varies drastically, such as in CUHK Avenue. To summarize, the effects of operating at a single scale are:

1. It reduces the quality of localization, because video patches at the fixed scale often exclude parts of objects or contain multiple objects; this also makes the learning task harder.
2. It forces pairing of same sized patches that contain objects of vastly different sizes, for example, when a dataset itself has high variance in object scales.
3. It forces us to rescale and resize entire datasets (target and source) in order to make the objects inside video patches between them "match scales" before learning a distance function.

We assert that operating at arbitrary scales takes us one step closer to being able to learn a general distance function and applying it on a target dataset out-of-the-box. In this section, we explore the feasibility of this.

One solution that has been commonly used in computer vision for detection tasks is the use of image pyramids and operating at many different scales. But in VADSiamese: (i) it is not clear whether using image pyramids at anomaly detection time on target dataset (*and not to* curate training data) is sufficient; (ii) the overhead of using image pyramids is not negligible.

We instead propose a more elegant solution - using unsupervised region/object proposals to extract arbitrary sized (in spatial dimensions) video patches (denoted g-VADSiamese henceforth). This entails a set of non-trivial modifications to VADSiamese that we now discuss. Finally, we show that the generalized approach removes the dependence of target dataset performance on choice of (and scaling of) source datasets while also achieving results that either surpass or are comparable to the single-scale version.

We use the unsupervised region proposal extraction method called Selective Search [49] on background-subtracted frames to extract arbitrary-sized bounding boxes around objects to extract video patches from. These new variable-sized video patches serve as our new atomic unit of processing in Siamese distance learning. Each dataset we use for experiments has a set of parameters for region-proposal extraction that work best that needs to be done once and are trivial enough that a non-expert is able to make choices that produce object proposals with a high recall and low false positive rate. We describe how these region proposals are further used in the 3 steps of our approach - curating a training dataset for a Siamese CNN, training a Siamese CNN and testing on the target video anomaly detection dataset.

6.1 Generating training video patch pairs

Curating a training set for the Siamese CNN follows from similar premises as with VADSiamese:

1. Every normal testing video patch must be similar to at least one training video patch picked from a *similar* region and of *similar* size.

2. Every anomalous testing video patch picked must be dissimilar to all training video patches picked from a *similar* region and of *similar* size.

A challenge in generating training pairs that followed from these rules is the use of the hand-crafted normalized L1 distance on video patches of different size. Empirically, we chose to crop or edge-pad a video patch to match the other's size in every pair before computing their distance. Other options such as resizing in different ways proved to be sub-optimal and our best guess as to why is that resizing a patch resizes the object it contains as well (along with the flow fields) and introduces artifacts. With region proposals, we found that setting α to 0.15 for use in adaptive threshold selection for a region works consistently well. All other aspects of creating the training set remain identical to those in VADSiamese.

6.2 Learning a distance function

Variable-size inputs to the network: The biggest obstacle to operating with variable sized video patch pairs in our approach is that VADSiamese's network architecture only accepted fixed-sized inputs caused by the fully connected layers at the end (see Figure 3). Unfortunately, creating a fully convolutional architecture is not straightforward. We instead propose using the Spatial Pyramid Pooling layer [15] to convert variable-sized feature maps to fixed-size feature maps. Spatial pyramid pooling works by dynamically determining the size of regions to pool over on a given input feature map to arrive at an output feature map of given size; the pooling operation could be either max pooling or average pooling and could be optionally done at different levels of an input-feature pyramid. While training, since the pooling operations are differentiable, backpropagating through the layer simply becomes an extension of the gradient routing that takes place with implementing, for example, a max pooling layer in the first place. Figure 7 describes the use of the SPP layer in our new network architecture. In the figure, each 2D conv operation actually refers to a conv-relu-brn sequence of operations where 'brn' refers to batch re-normalization [18], and each fully connected operation actually refers to a fc-relu sequence of operations.

Batching inputs to the network: Although this network architecture is able to accept arbitrary-sized video patches as input, all elements (where each element is a *pair* of input video patches) within a single batch during training must be of the same shape to avoid jagged tensors. To clarify, this does not mean that both inputs video patches in a pair must be of the same shape, but that all the *pairs* in a batch should have the same *paired-shape*. This has the implication that our batch sizes cannot be fixed since a different number of exemplars of each shape is stored and the number of pairs of video patches of the same paired-shape is highly variable. We found batch re-normalization to perform better than batch normalization because with with these changes to batching, while most of our batch sizes are close to 128, some of our batch sizes are as small as 1, and batch re-normalization is known to work better with smaller batch sizes.

Margin loss optimization: We also use a more recent state-of-the-art deep metric learning loss function called the margin loss [51]. We found empirically that the margin loss performs better on average than the cross-entropy loss with label smoothing that VADSiamese used, especially while operating on the newly curated training dataset. The margin loss between a pair of inputs i and j is given by:

$$l^{margin}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+. \quad (2)$$

where D_{ij} is the estimated distance between the inputs, β determines the boundary between positive and negative pairs and α controls the margin of separation, and $y_{ij} \in \{-1, 1\}$ denotes negative and positive pairs respectively. Notice that when $y_{ij} = -1$, $l^{margin}(i,j) = 0 \iff D_{ij} \geq \beta + \alpha$ and when $y_{ij} = 1$, $l^{margin}(i,j) = 0 \iff D_{ij} \leq \beta - \alpha$. In the original paper that introduced the margin loss [51], the authors denote D_{ij} to be the Euclidean distance between the embeddings of inputs i and j. We found that setting D_{ij} to be a *learned function* represented by an elementwise absolute value subtraction of embeddings followed by a simple fully connected 2-layer decision neural network instead works very well in practice (see Figure 7).

6.3 Exemplar learning and anomaly detection on target dataset.

The main challenge in the testing phase with the use of region proposals and the breaking of the fixed grid on the camera frame is in determining for a given video patch, which exemplar video patches to compare it against. When fixed-size patches were extracted from a fixed grid, patches in the same grid location on the camera frame were compared to each other. Since the fixed grid assumption is no longer valid with the arbitrary-sized video patches extracted from arbitrary positions of region proposals, we chose to compare video patches to others extracted from a spatial neighborhood with similar heights and widths, automatically determined by a reference video patch's size and the size of the

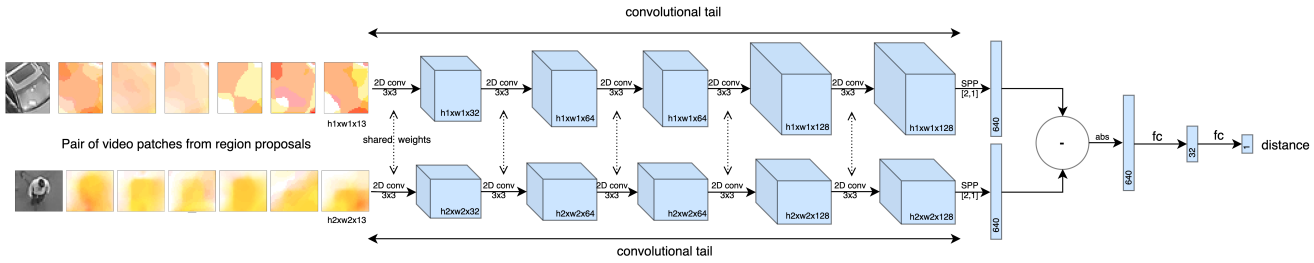

Fig. 7 Architecture of our Siamese neural network that learns a distance function between arbitrary-sized video patches. Best viewed in electronic form in color.

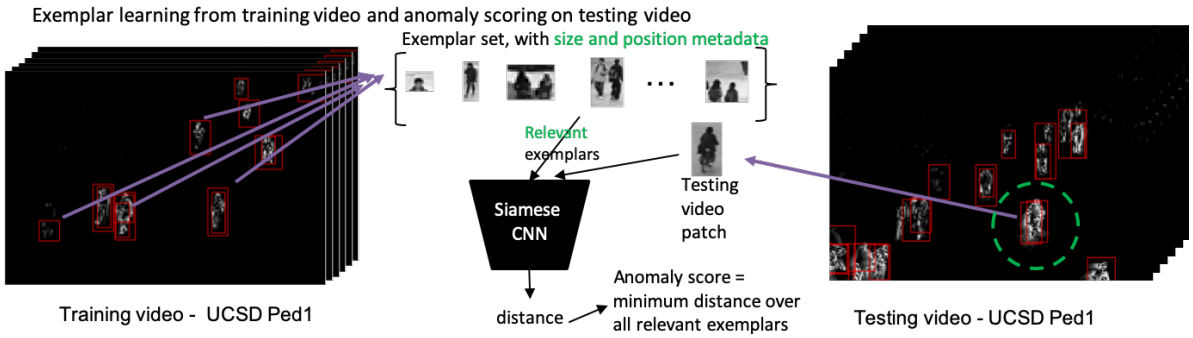

Fig. 8 Exemplar extraction and anomaly detection on the target dataset using region proposals. Exemplar video patches (depth dimension not shown) are stored along with size and position metadata, which are used to determine relevancy for comparison for each test video patch. Best viewed in electronic form in color.

camera frame for each dataset. Across all experiments and datasets, the "spatial neighborhood" is determined by $sqrt(I_h \times I_w)//5$, where I_h and I_w are the dimensions of the image frame; the "position buffer" is determined by $(clip(rh//3, 5, 20), clip(rw//3, 5, 20))$, where rh and rw are the dimensions of the reference video patch and the $clip()$ function enforces reasonable minimum and maximum values in pixel units. See Figure 8 for an illustration of exemplar extraction and anomaly detection in this new setting. Other aspects of exemplar extraction and anomaly detection are identical to that in VADSiamese, the only trivial hyperparameter being the exemplar threshold that governs the completeness of the exemplar set as a representation of the variations in the training set of the target dataset.

We use exactly the same datasets, evaluation protocol and hyperparameter settings as with VADSiamese to validate our approach. Regarding the training settings unique to our new generalized approach, we set $\beta = 10.0$ and $\alpha = 5.0$ for the margin loss as constant non-trainable parameters throughout training. The exemplar threshold was set to a natural choice of $\beta - \alpha$ (see Section 6.2 for the intuition behind this). Dropout [45] was found to be unnecessary with the margin loss. Additionally, we empirically found that using the average pooling operation over max pooling for the spatial pyramid pooling layer performs slightly better.

6.4 Comparison against VADSiamese

For quantitative evaluation, we compare the performance of g-VADSiamese to VADSiamese on the same benchmark datasets and evaluation measures. Table 5 presents frame and pixel AUCs on the 3 benchmark datasets for both methods. It is clear from the table that in the majority of cases, the region proposal based method either outperforms or is comparable to the single-scale version. Further, these results were generated with the source dataset configuration shown in Table 6, confirming the *removal of the dependence on the choice of source datasets for a particular target dataset* and the need to rescale source datasets to match the scale of objects in the target dataset. Table 7 also presents the track and region-based AUCs for comparison. Similar conclusions can be drawn. Specifically, the *huge jump in region-based AUC on CUHK Avenue of over 10% can be explained by the fact that this dataset contains objects of much larger scale than Ped1 or Ped2, which VADSiamese did not previously handle very well.*

In the qualitative sense, the detection results are even more impressive. The new detections are *almost always of higher quality and false positives almost always carry reasonable justifications upon visualizing them.* In Figure 9, we try to summarize the improvement in de-

Table 5 Frame-level and pixel-level AUCs on Ped1, Ped2 and CUHK Avenue.

Method	UCSD Ped1 frame AUC/EER	UCSD Ped1 pixel AUC*	UCSD Ped2 frame AUC/EER	UCSD Ped2 pixel AUC	CUHK Avenue frame AUC/EER
VADSiamese	86.0%/23.3%	80.4%	94.0%/14.1%	93.0%	87.2%/18.8%
g-VADSiamese	82.7%/**21.5%**	74.9%	**95.0%/14.1%**	**93.8%**	**89.4%/15.8%**

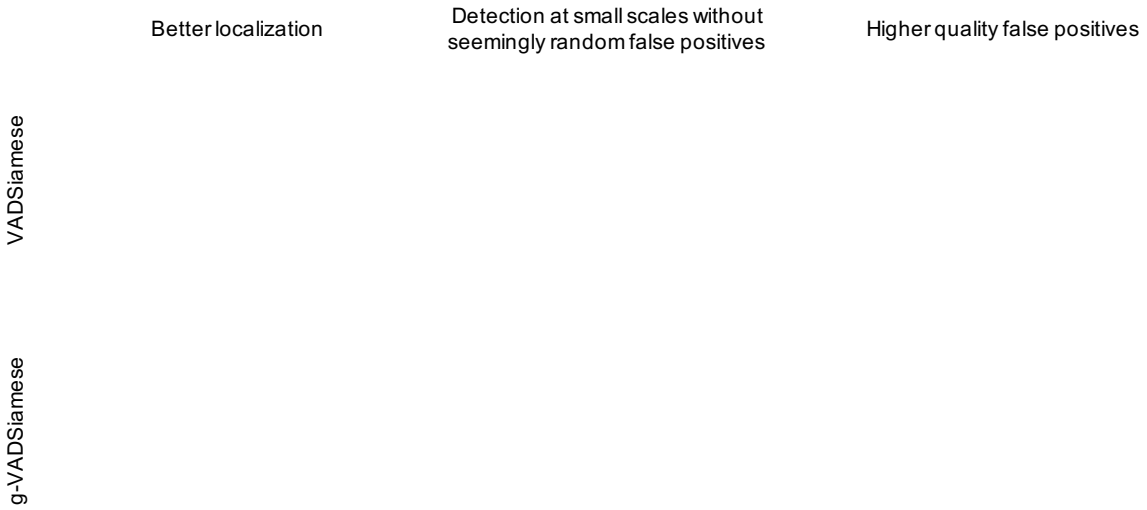

Fig. 9 Improvement in detections in moving from operating at a single fixed scale to using region proposals. Ground truth in green and detection in red. Best viewed in electronic form in color.

Table 6 Source dataset configuration for each target dataset.

Target dataset	Source datasets
UCSD Ped1	UCSD Ped2 (rotated at 45 degrees), CUHK Avenue
UCSD Ped2	UCSD Ped1, CUHK Avenue
CUHK Avenue	UCSD Ped1, UCSD Ped2

Table 7 Track and region-based AUC for false positive rates up to 1.0.

Method	track AUC			region AUC		
	Ped1	Ped2	Avenue	Ped1	Ped2	Avenue
VADSiamese	90.0%	89.3%	78.6%	59.2%	74.0%	41.2%
g-VADSiamese	89.4%	**92.6%**	78.2%	56.3%	**76.7%**	**55.2%**

tections obtained by g-VADSiamese. Specifically, improvements are observed in 3 aspects:

1. Since extracting video patches from a fixed grid on the camera frame often produced patches that had multiple objects or parts of objects, the region proposal localization which does not suffer from this drawback displays much *better localization*. Figure 9 (left) shows an example of how better localization is achieved by using region proposals.
2. Since the variation in scale of objects across the camera frame does not affect g-VADSiamese adversely, again the quality of localization is much better and the *number of false positives at smaller scales is reduced*. Figure 9 (center) shows an example of detections at smaller scales do not cause false positives at the large scales within a testing frame.
3. Finally, in our observation, the *quality of false positives is much higher* in g-VADSiamese. We believe that this is mainly because of improved localization as well. Also, we believe the ground truth annotation task by the dataset creators being inherently ambiguous has resulted in a conservative annotation approach. Figure 9 (right) shows an example of how a group of persons surveying a building is flagged as anomalous by g-VADSiamese because the region proposal captured the whole group. A different set of annotators might have called this anomalous since no such behavior exists in the training set. VADSiamese does not deem this region anomalous because it never sees a patch of the whole group due to the lack of an object proposal scheme. On the other hand, VADSiamese displays many more "seemingly random" false positives of patches that contain "parts of objects".

6.5 Ablation study on choice of exemplar threshold

We felt it was important to show that the main parameter choice for deploying a trained network on a target camera, being the choice of exemplar threshold,

is trivial to select. To this end, we performed an ablation study on exemplar threshold while fixing the target dataset as UCSD Ped1. Table 8 shows the result of this experiment. It is clear that as the exemplar threshold is increased, the number of exemplars stored rapidly decreases and unless the exemplar threshold is very large, a wide range of exemplar thresholds produce similar excellent anomaly detection performance. This is because a large exemplar threshold renders the exemplar model to be an undercomplete representation of the variations in the training split of the target dataset, causing many video patches at test time to have distant nearest neighbors, resulting in rapid accumulation of false positives as the anomaly score threshold is decreased in computation of the AUCs.

6.6 Nearest neighbor visualizations

Another popular proof-of-concept experiment in image retrieval and patch matching tasks is to visualize for some test instances, their nearest neighbors as determined by the trained model. We show in Figure 10, for a subset of region proposals extracted from a single random testing frame from UCSD Ped2, their nearest neighbor exemplar video patches as determined by our best model along with the real-valued distance determined by the model. From the figure, it is clear that large distances are assigned to the ground truth anomalous patches. Moreover, the nearest neighbors are of high quality visually and the real-valued distances predicted are arguably *plausible to human perception in terms of ordering.*

6.7 A note on computational requirements

The most expensive step of our approach is video patch pair dataset creation, but this only needs to be done once and offline. At inference time, anomaly detection the computational complexity per frame of this approach is heavily dependent on a few factors: i) resolution of the test dataset ii) size of region proposals iii) amount of variation in the normal data (or by proxy, the size of the exemplar set). This makes a discussion of inference time and memory requirements in a general manner difficult. Additionally, we acknowledge that in practice, one could realistically employ a single-shot object detector such as SSD [26] fine-tuned to the object categories of interest to vastly speed up object proposal extraction. In this work, our focus was to address the issue of operating at a fixed scale, while attempting to improve localization performance of VADSiamese. We leave further discussion and optimization for future work as it is out of the scope of what this paper is trying to accomplish.

7 Conclusion

We have presented a novel approach to video anomaly detection that introduces a new paradigm to use a deep network for this problem. We substitute the problem of classifying a video patch as anomalous or not for the problem of estimating distance between two video patches, for which we can generate plenty of labeled training data. The learned metric (which also learns a feature vector to represent a video patch) can then be used in a straightforward video anomaly detection method that measures the perceptual distance from each testing video patch to the nearest exemplar video patch for that region.

We further generalized the approach from operating on fixed-size video patches extracted from a fixed grid overlaid on the camera frame to arbitrary-sized video patches extracted from unsupervised region proposals. We showed with extensive experiments that operating at arbitrary scale brings us a step closer to learning a generic distance function from random surveillance video to use in an out-of-the-box fashion for a target video anomaly detection task.

We have shown that our approaches either surpass or perform comparably to the previous state-of-the-art without any training of the Siamese network on data from the target dataset. Our approaches also possess some favorable properties in being a plug-and-play method (learned distance function can be used out-of-the-box on target dataset), and in being scalable and resistant to environmental changes (updating the exemplar set is easy).

We think this work lays a foundation in metric learning based video anomaly detection research for several future directions such as: (1) the use of more sophisticated deep metric learning losses such as a triplet loss [17], (2) the use of dynamic sampling strategies [51] during metric learning, (3) using single-shot detectors [26] for fast region proposal extraction with high-resolution RGB datasets such as Street Scene [37] and (4) domain adaptation between source and target domains such as with domain-adversarial learning [11].

Acknowledgements The authors would like to thank Zexi Chen and Benjamin Dutton of the STAC lab at NC State University for relevant stimulating discussions.

Table 8 Ablation study on exemplar threshold.

Exemplar threshold	Size of exemplar set	Track AUC	Region AUC	Frame AUC	Pixel AUC
5	19.5k	89.4%	56.3%	82.7%	74.9%
10	3.8k	89.3%	56.6%	82.2%	74.0%
15	1.6k	87.3%	56.6%	82.9%	73.7%
25	900	85.1%	50.9%	81.2%	70.2%

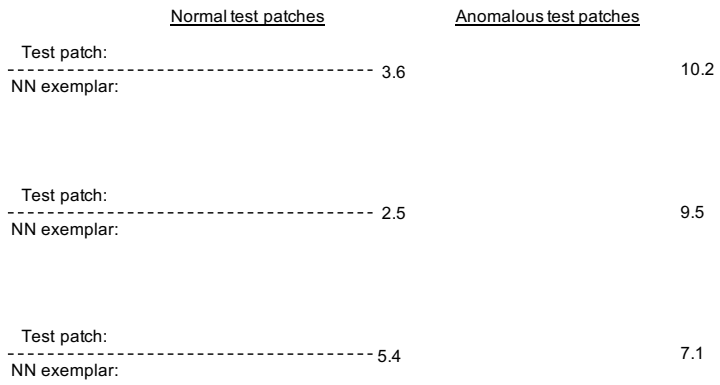

Fig. 10 Nearest neighbor exemplars for some video patches extracted from a test frame along with their estimated distances. Best viewed in electronic form in color.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Adam A, Rivlin E, Shimshoni I, Reinitz D (2008) Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3):555–560, DOI 10.1109/TPAMI.2007.70825, URL http://ieeexplore.ieee.org/document/4407716/
2. Antic B, Ommer B (2011) Video parsing for abnormality detection. In: IEEE International Conference on Computer Vision, Barcelona, Spain, pp 2415–2422, DOI 10.1109/ICCV.2011.6126525, URL http://ieeexplore.ieee.org/document/6126525/
3. Antić B, Ommer B (2015) Spatio-temporal Video Parsing for Abnormality Detection. arXiv preprint arXiv:150206235
4. Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R (1994) Signature verification using a siamese time delay neural network. In: Advances in neural information processing systems, pp 737–744
5. Chong YS, Tay YH (2015) Modeling Representation of Videos for Anomaly Detection using Deep Learning: A Review. arXiv preprint arXiv:150500523 URL https://arxiv.org/abs/1505.00523
6. Chong YS, Tay YH (2017) Abnormal Event Detection in Videos using Spatiotemporal Autoencoder. Advances in Neural Networks - ISNN 2017 Lecture Notes in Computer Science
7. Cong Y, Yuan J, Liu J (2013) Abnormal event detection in crowded scenes using sparse representation. Pattern Recognition 46(7):1851–1864, DOI 10.1016/j.patcog.2012.11.021, URL http://linkinghub.elsevier.com/retrieve/pii/S0031320312005055
8. Del Giorno A, Bagnell JA, Hebert M (2016) A Discriminative Framework for Anomaly Detection in Large Videos. In: European Conference on Computer Vision (ECCV), pp 334–349, DOI 10.1007/978-3-319-46454-1_21, URL http://link.springer.com/10.1007/978-3-319-46454-1_21
9. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. VS-PETS Beijing, China
10. Feng Y, Yuan Y, Lu X (2017) Learning deep event models for crowd anomaly detection. Neurocomputing 219:548–556, DOI 10.1016/j.neucom.2016.09.063, URL https://linkinghub.elsevier.com/retrieve/pii/S0925231216310980
11. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. The Journal of Machine Learning Re-

search 17(1):2096–2030
12. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of Machine Learning Research (PMLR), pp 249–256, URL http://proceedings.mlr.press/v9/glorot10a.html
13. Han X, Leung T, Jia Y, Sukthankar R, Berg AC (2015) Matchnet: Unifying feature and metric learning for pbased matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3279–3286
14. Hasan M, Choi J, Neumann J, Roy-Chowdhury AK, Davis LS (2016) Learning Temporal Regularity in Video Sequences. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp 733–742, DOI 10.1109/CVPR.2016.86, URL http://ieeexplore.ieee.org/document/7780455/
15. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9):1904–1916
16. Hinami R, Mei T, Satoh S (2017) Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge. In: IEEE International Conference on Computer Vision (ICCV), Venice, pp 3639–3647, DOI 10.1109/ICCV.2017.391, URL http://ieeexplore.ieee.org/document/8237653/
17. Hoffer E, Ailon N (2015) Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, Springer, pp 84–92
18. Ioffe S (2017) Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In: Advances in Neural Information Processing Systems, pp 1945–1953
19. Ionescu RT, Smeureanu S, Alexe B, Popescu M (2017) Unmasking the Abnormal Events in Video. In: IEEE International Conference on Computer Vision (ICCV), Venice, pp 2914–2922, DOI 10.1109/ICCV.2017.315, URL http://ieeexplore.ieee.org/document/8237577/
20. Ionescu RT, Khan FS, Georgescu MI, Shao L (2019) Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 7842–7851
21. Ionescu RT, Smeureanu S, Popescu M, Alexe B (2019) Detecting Abnormal Events in Video Using Narrowed Normality Clusters. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1951–1960, DOI 10.1109/WACV.2019.00212
22. Jones M, Nikovski D, Imamura M, Hirata T (2016) Exemplar learning for extremely efficient anomaly detection in real-valued time series. Data Mining and Knowledge Discovery (DMKD) 30(6):1427–1454
23. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980
24. Leyva R, Sanchez V, Li CT (2017) Video Anomaly Detection With Compact Feature Sets for Online Performance. IEEE Transactions on Image Processing 26(7):3463–3478, DOI 10.1109/TIP.2017.2695105, URL http://ieeexplore.ieee.org/document/7903693/
25. Liu C (2009) Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. MIT PhD Thesis
26. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European Conference on Computer Vision (ECCV), Springer, pp 21–37
27. Liu W, Luo W, Lian D, Gao S (2018) Future frame prediction for anomaly detection–a new baseline. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6536–6545
28. Liu Y, Li CL, Póczos B (2018) Classifier Two-Sample Test for Video Anomaly Detections. In: British Machine Vision Conference (BMVC)
29. Lobo JM, Jiménez-Valverde A, Real R (2008) AUC: a misleading measure of the performance of predictive distribution models. Global Ecology and Biogeography 17(2):145–151, DOI 10.1111/j.1466-8238.2007.00358.x, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1466-8238.2007.00358.x
30. Lu C, Shi J, Jia J (2013) Abnormal Event Detection at 150 FPS in MATLAB. In: IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, pp 2720–2727, DOI 10.1109/ICCV.2013.338, URL http://ieeexplore.ieee.org/document/6751449/
31. Luo W, Liu W, Gao S (2017) A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In: IEEE International Conference on Computer Vision (ICCV), Venice, pp 341–349, DOI 10.1109/ICCV.2017.45, URL http://ieeexplore.ieee.org/document/8237307/
32. Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1975–1981, DOI 10.1109/CVPR.2010.5539872

33. Mehran R, Oyama A, Shah M (2009) Abnormal crowd behavior detection using social force model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 935–942, URL http://ieeexplore.ieee.org/abstract/document/5206641/
34. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application VISSAPP'09), INSTICC Press, pp 331–340
35. Popoola OP, Kejun Wang (2012) Video-Based Abnormal Human Behavior Recognition– A Review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(6):865–878, DOI 10.1109/TSMCC.2011.2178594, URL http://ieeexplore.ieee.org/document/6129539/
36. Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (2009) Dataset Shift in Machine Learning. The MIT Press
37. Ramachandra B, Jones M (2020) Street Scene: A new dataset and evaluation protocol for video anomaly detection. In: IEEE Winter Conference on Applications of Computer Vision (WACV)
38. Ramachandra B, Jones M, Vatsavai R (2020) Learning a distance function with a siamese network to localize anomalies in videos. In: The IEEE Winter Conference on Applications of Computer Vision, pp 2598–2607
39. Ramachandra B, Jones M, Vatsavai RR (2020) A survey of single-scene video anomaly detection. IEEE Transactions on Pattern Analysis and Machine Intelligence
40. Ravanbakhsh M, Nabi M, Sangineto E, Marcenaro L, Regazzoni C, Sebe N (2017) Abnormal event detection in videos using generative adversarial nets. In: IEEE International Conference on Image Processing (ICIP), pp 1577–1581, DOI 10.1109/ICIP.2017.8296547
41. Ravanbakhsh M, Nabi M, Mousavi H, Sangineto E, Sebe N (2018) Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection. In: IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, pp 1689–1698, DOI 10.1109/WACV.2018.00188, URL https://ieeexplore.ieee.org/document/8354292/
42. Saligrama V, Chen Z (2012) Video anomaly detection based on local statistical aggregates. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp 2112–2119
43. Sjarif NNA, Shamsuddin SM, Hashim SZ (2012) Detection of abnormal behaviors in crowd scene: a review. Int J Advance Soft Comput Appl 4(1), URL https://www.researchgate.net/profile/Nilam_Sjarif/publication/288703678_Detection_of_abnormal_behaviors_in_crowd_scene_A_review/links/56b0065808ae9c1968b4904c.pdf
44. Smeureanu S, Ionescu RT, Popescu M, Alexe B (2017) Deep Appearance Features for Abnormal Behavior Detection in Video. In: Battiato S, Gallo G, Schettini R, Stanco F (eds) International Conference on Image Analysis and Processing (ICIAP), Springer International Publishing, Cham, pp 779–789, DOI 10.1007/978-3-319-68548-9_70, URL http://link.springer.com/10.1007/978-3-319-68548-9_70
45. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research
46. Sultani W, Chen C, Shah M (2018) Real-World Anomaly Detection in Surveillance Videos. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, pp 6479–6488, DOI 10.1109/CVPR.2018.00678, URL https://ieeexplore.ieee.org/document/8578776/
47. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the Inception Architecture for Computer Vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp 2818–2826, DOI 10.1109/CVPR.2016.308, URL http://ieeexplore.ieee.org/document/7780677/
48. Turchini F, Seidenari L, Del Bimbo A (2017) Convex Polytope Ensembles for Spatio-Temporal Anomaly Detection. In: Battiato S, Gallo G, Schettini R, Stanco F (eds) International Conference on Image Analysis and Processing (ICIAP), Springer International Publishing, Lecture Notes in Computer Science, pp 174–184
49. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. International Journal of Computer Vision 104(2):154–171
50. Weixin Li, Mahadevan V, Vasconcelos N (2014) Anomaly Detection and Localization in Crowded Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(1):18–32, DOI 10.1109/TPAMI.2013.111, URL http://ieeexplore.ieee.org/document/6531615/

51. Wu CY, Manmatha R, Smola AJ, Krahenbuhl P (2017) Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 2840–2848
52. Wu S, Moore BE, Shah M (2010) Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2054–2060
53. Xu D, Ricci E, Yan Y, Song J, Sebe N (2015) Learning deep representations of appearance and motion for anomalous event detection. arXiv preprint arXiv:151001553
54. Zagoruyko S, Komodakis N (2015) Learning to compare image patches via convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4353–4361, DOI 10.1109/CVPR.2015.7299064, URL `http://ieeexplore.ieee.org/document/7299064/`
55. Zhan B, Monekosso DN, Remagnino P, Velastin SA, Xu LQ (2008) Crowd analysis: a survey. Machine Vision and Applications 19(5-6):345–357, DOI 10.1007/s00138-008-0132-4, URL `http://link.springer.com/10.1007/s00138-008-0132-4`