Anomaly Detection in Video via Self-Supervised and Multi-Task Learning

Mariana-Iuliana Georgescu1,3, Antonio Bărbălău1, Radu Tudor Ionescu1,3, Fahad Shahbaz Khan2, Marius Popescu1,3, Mubarak Shah4
1University of Bucharest, Romania, 2MBZ University of Artificial Intelligence, Abu Dhabi
3SecurifAI, Romania, 4University of Central Florida, Orlando, FL

Abstract

Anomaly detection in video is a challenging computer vision problem. Due to the lack of anomalous events at training time, anomaly detection requires the design of learning methods without full supervision. In this paper, we approach anomalous event detection in video through self-supervised and multi-task learning at the object level. We first utilize a pre-trained detector to detect objects. Then, we train a 3D convolutional neural network to produce discriminative anomaly-specific information by jointly learning multiple proxy tasks: three self-supervised and one based on knowledge distillation. The self-supervised tasks are: (i) discrimination of forward/backward moving objects (arrow of time), (ii) discrimination of objects in consecutive/intermittent frames (motion irregularity) and (iii) reconstruction of object-specific appearance information. The knowledge distillation task takes into account both classification and detection information, generating large prediction discrepancies between teacher and student models when anomalies occur. To the best of our knowledge, we are the first to approach anomalous event detection in video as a multi-task learning problem, integrating multiple self-supervised and knowledge distillation proxy tasks in a single architecture. Our lightweight architecture outperforms the state-of-the-art methods on three benchmarks: Avenue, ShanghaiTech and UCSD Ped2. Additionally, we perform an ablation study demonstrating the importance of integrating self-supervised learning and normality-specific distillation in a multi-task learning setting.

1. Introduction

In recent years, a growing interest has been dedicated to the task of detecting anomalous events in video [8, 9, 10, 13, 17, 19, 20, 24, 30, 34, 35, 36, 37, 38, 39, 49, 51, 55, 57, 61, 62, 63]. An anomalous event is commonly defined as an unfamiliar or unexpected event in a given context. For example, a person crossing the road can be viewed as anomalous if the event does not happen on the crosswalk. This example shows that context plays a key role in the definition of anomalous events and, consequently, in the problem formulation. Indeed, the reliance on context, coupled with the large variety of unexpected events, makes it extremely difficult to collect anomalous events for training. Hence, the anomaly detection problem is typically regarded as an outlier detection task. Then, a normality model is fit on normal training data, labeling events that deviate from the model as anomalous. Without being able to employ standard supervision, researchers have proposed alternative approaches ranging from distance-based [17, 19, 37, 38, 40, 44, 46, 47, 50, 52, 59] and reconstruction-based strategies [5, 13, 14, 17, 29, 31, 34, 36, 41, 51, 53] to probabilistic [1, 2, 4, 12, 16, 21, 32, 33, 58] and change detection methods [7, 18, 28, 35].

In lieu of learning to discriminate directly between normal and anomalous events, related methods approach a different yet connected task. For example, in the pioneering work of Liu et al. [27], a neural network learns to predict future video frames. During inference, an event is labeled as anomalous if the predicted future frame exhibits a high reconstruction error. Although the state-of-the-art methods attain impressive results, addressing anomaly detection through a single proxy task is suboptimal, since the proxy task is not well aligned with anomaly detection. For instance, a car stopped in a pedestrian area should be labeled as an anomaly, yet the car is trivial to reconstruct in a future frame (since it is standing still). We therefore propose to perform anomaly detection by training a model jointly on multiple proxy tasks. Following a series of recent methods [9, 10, 17, 61], we also employ an object detector, subsequently performing anomaly detection at the object level. However, these recent methods take into account a single proxy task. Different from [9, 10, 17, 61], we propose a novel anomaly detection approach that jointly learns a set of multiple proxy tasks through a single object-centric architecture.

As discussed above, we devise an object-centric approach comprising a 3D convolutional neural network (CNN) that jointly learns the following proxy tasks: (i) predicting the arrow of time (discriminating between forward and backward moving objects), (ii) predicting the irregularity of motion (discriminating between objects cap
Our anomaly detection framework based on self-supervised and multi-task learning. First, we detect the objects in video with the help of an object detector (YOLOv3). For each object, we devise three self-supervised tasks (learning the arrow of time, predicting motion irregularity and predicting the object appearance in the middle box) and a knowledge distillation task (using YOLOv3 and ResNet-50 as teachers). A 3D convolutional neural network is trained jointly on the four tasks. Models represented with dashed lines are pre-trained. Best viewed in color.

1. Learning the arrow of time.
2. Predicting motion irregularity.
3. Predicting the object appearance in the middle box.
4. Understanding normality-specific class probabilities by distilling pre-trained classification (ImageNet [43]) and detection (MS COCO [26]) teachers.

To jointly address these self-supervised and knowledge distillation tasks, we integrate a prediction head for each corresponding task, as illustrated in Figure 1. To our knowledge, we are the first to propose a multi-task learning approach that integrates a set of novel self-supervised and knowledge distillation proxy tasks in a single object-centric architecture for anomaly detection in video.

We perform comprehensive experiments on three benchmarks, namely Avenue [29], ShanghaiTech [31] and UCSD Ped2 [32]. Our approach outperforms the state-of-the-art methods [7, 8, 9, 10, 13, 14, 16, 17, 18, 19, 20, 21, 23, 24, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 40, 41, 47, 48, 49, 51, 53, 55, 57, 59, 60, 61, 62, 64] on all three data sets, achieving frame-level AUC scores of 92.8% on Avenue, 90.2% on ShanghaiTech and 99.8% on UCSD Ped2. Additionally, we present empirical evidence confirming that a jointly optimized model on the proposed proxy tasks outperforms single models optimized on individual tasks, thus indicating that modeling anomaly detection through a single proxy task is suboptimal.

In summary, our contribution is multifold:

- We introduce learning the arrow of time as a proxy task for anomaly detection.
- We introduce motion irregularity prediction as a proxy task for anomaly detection.
- We introduce model distillation as a proxy task for anomaly detection.
- We pose anomaly detection in video as a multi-task learning problem, integrating multiple self-supervised and knowledge distillation tasks into a single model.
- We conduct experiments showing that our approach attains superior results compared to the state-of-the-art methods on three benchmarks.
2. Related Work

While the early works [1, 2, 6, 25, 29, 32, 33, 46, 58] on video anomaly detection relied heavily on handcrafted appearance and motion features, the recent literature is abundant in deep learning methods [9, 10, 14, 16, 17, 27, 31, 38, 40, 41, 44, 47, 54, 59, 60]. For instance, Xu et al. [59] proposed the use of stacked denoising auto-encoders to automatically learn both appearance and motion features, which are further used as input for multiple one-class SVM models. Hasan et al. [14] diverged from using auto-encoders simply as feature extractors for subsequent models, leveraging the reconstruction error as an estimator for abnormality. More recently, Wang et al. [54] proposed a further improvement by combining CNNs with LSTMs, forming a spatio-temporal auto-encoder able to better account for the temporal evolution of spatial features. Wang et al. [54] rely on the assumption that anomalous events will cause significant discrepancies between future and past frames. Employing generative networks for video anomaly detection [8, 36, 41] is another significant line of research that relies on the same principle, that is, synthesizing future frames will prove to be significantly more challenging when an anomalous event occurs than in a normal situation. To this end, Liu et al. [27] employed a generative model to predict future frames, considering the reconstruction error as an indicator of abnormality. In another similar framework, Lee et al. [24] proposed to predict the middle frame, considering a bidirectional approach that learns from both past and future frames. Similar to future frame [8, 27] or middle frame [24] prediction frameworks, we propose a framework that incorporates middle frame prediction. Different from methods such as [8, 24, 27, 54], we study middle frame prediction at the object level, enabling the accurate localization of anomalies. Moreover, middle frame prediction is just one of our four proxy tasks. To our knowledge, we are the first to propose learning the arrow of time, motion irregularity prediction and model distillation as proxy tasks for anomaly detection in video. We note that model distillation has been studied as a single task for anomaly detection in still images [3]. However, our ablation results show that model distillation alone is not sufficient for anomaly detection in video.

Aside from the direction relying on reconstruction errors [14, 27, 29, 31, 34, 36, 41, 51, 53], other recent works, such as [9, 38], tackle the problem from completely different angles. For example, Ramachandra et al. [38] employed a Siamese network to learn a metric between spatio-temporal video patches. In this scenario, the dissimilarity between patches provides the means to estimate the level of abnormality.

In addition, anomalous event detection approaches can be divided with respect to the level of analysis. While some frameworks, such as [27, 33, 40, 41, 47], approach the problem from a global (frame-level) perspective, methods such as [7, 11, 21, 19, 28, 29, 31, 32, 44, 46, 64] extract features at a local level, e.g. by considering spatio-temporal cubes. In some cases, the detection of anomalous events is explored with multi-level frameworks, a recent example being the work of Lee et al. [24]. Aside from these mainstream perspectives, Ionescu et al. [17] introduced a novel object-centric framework, employing a single-shot object detector on each frame, before applying convolutional auto-encoders to learn deep unsupervised representations as part of a one-versus-rest classification approach based on clustering training samples into normality clusters. A few recent works [9, 10, 61] further explored the same line of research, proposing alternative object-centric frameworks. Similar to object-centric frameworks such as [9, 10, 17, 61], we employ an object detector, focusing our analysis on the detected objects. Unlike [9, 10, 17, 61], we perform the analysis through a series of proxy self-supervised and model distillation tasks, proposing a novel anomaly detection framework based on multi-task learning. Hence, the only common aspect with the other object-centric methods [9, 10, 17, 61] is the use of an object detector.

The related methods presented so far follow the mainstream formulation of anomalous event detection, which implies that an anomalous event is an unfamiliar event in a known context. In the mainstream formulation, anomalous events are not available at training time, as it is considered too difficult to collect a sufficiently wide variety of anomalous events. Although our study adopts the mainstream formulation, we acknowledge the recent effort of Sultani et al. [48], which considers anomalous events that do not depend on the context. By eliminating the reliance on context, they are able to collect and use anomalous events at training. In their formulation, anomalous event detection becomes equivalent to action recognition in video. We thus consider the line of research initiated by Sultani et al. [48] and continued by others [65] less related to our study.

3. Method

3.1. Motivation and Overview

Motivation. Modeling anomalous event detection through a single proxy task, e.g. future frame prediction [27], is suboptimal due to the lack of perfect alignment between the proxy task and the actual (anomaly detection) task. To reduce the non-alignment of the model with respect to the anomaly detection task, we propose to train the model by jointly optimizing it on multiple proxy tasks.

Training. Our framework based on self-supervised and multi-task learning is illustrated in Figure 1. First, we detect the objects in each frame using a pre-trained YOLOv3 [42] detector, obtaining a list of bounding boxes. For each detected object in the frame $i$, we create an object-centric temporal sequence by simply cropping the corresponding bounding box from frames $\{i-t, ..., i-1, i, i+1, ..., i+t\}$.
possible combinations of shallow, deep, narrow and wide architectures. These are: shallow+narrow, shallow+wide, deep+narrow and deep+wide. The detailed configuration of each 3D CNN architecture is presented in Table 1.

For each network configuration, the spatial size of the RGB input is $64 \times 64$ pixels. The 3D conv layers use filters of $3 \times 3 \times 3$. Each conv layer is followed by a batch normalization layer and a ReLU activation. Our shallow+narrow 3D CNN is formed of three 3D conv layers and three 3D max-pooling layers. Its first 3D conv layer is composed of 16 filters and the next two conv layers are composed of 32 filters each. The padding is set to “same” and the stride is set to 1. We perform only spatial pooling for the first two 3D max-pooling layers. The pooling size and the stride are both set to 2. The last 3D max-pooling layer performs global temporal pooling, keeping the same configuration as the first two pooling layers at the spatial level. Using temporal pooling only once (in the last pooling layer) enables us to employ a different temporal size for each proxy task.

In the shallow+wide configuration, we change the 3D CNN by doubling the number of filters in each conv layer. For the deep+narrow architecture, we increase the number of 3D conv layers from three to six. Finally, in the deep+wide configuration, we double the number of layers as well as the number of filters in each conv layer with respect to the shallow+narrow model.

In the middle object prediction head, we incorporate a decoder formed of upsampling and 2D conv layers based on $3 \times 3$ filters. The number of upsampling operations is always equal to the number of max-pooling layers in the 3D CNN. Similarly, the number of 2D conv layers in the decoder matches the number of 3D conv layers in the 3D CNN. Each upsampling operation is based on nearest neighbor interpolation, increasing the spatial support by a factor of $2 \times$. The last conv layer in the decoder has only three filters in order to reconstruct the RGB input.

The other three prediction heads share the same configuration, having a 2D conv layer with 32 filters and a max-pooling layer with a spatial support of $2 \times 2$. The last layer is a fully-connected layer with either two units to predict the arrow of time and motion irregularity or 1080 units to predict the teachers’ output scores for the 1000 ImageNet [43] classes and the 80 MS COCO [26] categories.

### 3.3. Proxy Tasks and Joint Learning

**Task 1: Arrow of time.** To predict the arrow of time [56] at the object level, we generate two labeled training samples from each object-centric sequence. The first sample takes the frames in their temporal order, namely $(i - t, ..., i - 1, i, i + 1, ..., i + t)$, thus being labeled as forward motion (class 1). The second sample takes the frames in reversed order, namely $(i + t, ..., i + 1, i, i - 1, ..., i - t)$, being labeled as backward motion (class 2). During inference, we expect the arrow of time to be harder to predict for objects with

Table 1. Alternative architectures considered for the 3D CNN included in our anomaly detection framework. Global temporal pooling is denoted by “$\rightarrow$”.

<table>
<thead>
<tr>
<th>Width</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3 \times 3 \times 3$ conv 16</td>
<td>$3 \times 3 \times 3$ conv 16</td>
</tr>
<tr>
<td>$1 \times 2 \times 2$ max-pooling</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$1 \times 2 \times 2$ max-pooling</td>
</tr>
<tr>
<td>$1 \times 2 \times 2$ max-pooling</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$: \times 2 \times 2$ max-pooling</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$1 \times 2 \times 2$ max-pooling</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$: \times 2 \times 2$ max-pooling</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$1 \times 2 \times 2$ max-pooling</td>
<td>$3 \times 3 \times 3$ conv 32</td>
</tr>
<tr>
<td>$3 \times 3 \times 3$ conv 32</td>
<td>$: \times 2 \times 2$ max-pooling</td>
</tr>
</tbody>
</table>
anomalous motion. Let $f$ be the shared 3D CNN and $h_{T_1}$ be the arrow of time head. Let $X^{(T_1)}$ be a forward or backward object-centric sequence of size $(2 \cdot t + 1) \times 64 \times 64 \times 3$. We use the cross-entropy loss to train the arrow of time head:

$$
\mathcal{L}_{T_1} \left( X^{(T_1)}, Y^{(T_1)} \right) = - \sum_{k=1}^{2} Y_{k}^{(T_1)} \log \left( \hat{Y}_{k}^{(T_1)} \right),
$$

where $\hat{Y}_{k}^{(T_1)} = \text{softmax} \left( h_{T_1} \left( f (X^{(T_1)}) \right) \right)$ and $Y^{(T_1)}$ is the one-hot encoding of the ground-truth label for $X^{(T_1)}$.

**Task 2: Motion irregularity.** Assuming that some anomalies can be identified through irregular motion patterns, we train our model to predict if an object-centric sequence has consecutive or intermittent frames (some frames being skipped). To learn motion irregularity, we generate two labeled training samples from each object-centric sequence. The first example captures an object in consecutive frames from $i - t$ to $i + t$, the corresponding class being regular motion (class 1). The intermittent object-centric sequence is created by retaining the frame $i$, then gradually adding $t$ randomly chosen previous frames and $t$ randomly chosen succeeding frames. The intermittent frames are chosen by skipping frames using random gaps in the range $\{1, 2, 3, 4\}$. The intermittent object-centric sequence is labeled as irregular motion (class 2). Let $h_{T_2}$ be the irregular motion head and $X^{(T_2)}$ be a regular or irregular object-centric sequence of size $(2 \cdot t + 1) \times 64 \times 64 \times 3$. We employ the cross-entropy loss to train the motion irregularity head:

$$
\mathcal{L}_{T_2} \left( X^{(T_2)}, Y^{(T_2)} \right) = - \sum_{k=1}^{2} Y_{k}^{(T_2)} \log \left( \hat{Y}_{k}^{(T_2)} \right),
$$

where $\hat{Y}_{k}^{(T_2)} = \text{softmax} \left( h_{T_2} \left( f (X^{(T_2)}) \right) \right)$ and $Y^{(T_2)}$ is the one-hot encoding of the ground-truth label for $X^{(T_2)}$.

**Task 3: Middle bounding box prediction.** Our 3D CNN model also learns to reconstruct objects detected in the normal training videos. From each object-centric sequence, we select the image samples cropped from frames $\{i - t, \ldots, i - 1, i + 1, \ldots, i + t\}$, forming the input object-centric sequence $X^{(T_3)}$ of size $(2 \cdot t) \times 64 \times 64 \times 3$. The middle image, corresponding to the bounding box in frame $i$, represents the target output $Y^{(T_3)}$ of size $64 \times 64 \times 3$. When we encounter an anomaly with unusual motion, such as a person running, the input object-centric sequence of that person will not contain enough information for the model to accurately reconstruct the middle bounding box, thus being labeled as anomalous. Let $h_{T_3}$ be the middle bounding box prediction head. We use the $L_1$ loss to learn the middle bounding box prediction task:

$$
\mathcal{L}_{T_3} \left( X^{(T_3)}, Y^{(T_3)} \right) = \frac{1}{h \cdot w \cdot c} \sum_{j=1}^{h} \sum_{k=1}^{w} \sum_{c=1}^{c} \left| Y^{(T_3)}_{jkl} - \hat{Y}^{(T_3)}_{jkl} \right|,
$$

where $\hat{Y}^{(T_3)} = h_{T_3} \left( f (X^{(T_3)}) \right)$ and $h \times w \times c$ is the size of the output, i.e., $h = 64, w = 64$ and $c = 3$.

**Task 4: Model distillation.** On the one hand, our 3D CNN model learns to predict the features from the last layer (just before softmax) of a ResNet-50 [15], which is pre-trained on ImageNet. On the other hand, our 3D CNN model learns to predict the class probabilities predicted by YOLOv3 [42], which is pre-trained on MS COCO. During distillation, our model learns the predictive behavior of the teachers on normal data. During inference, we expect high prediction discrepancies between our student and the YOLOv3 teacher when we encounter an object with unusual appearance or that belongs to an object category not seen during training. We refrain from using ResNet-50 during inference in order to save valuable computational time. We note that YOLOv3 is applied only once on each frame $i$, the corresponding class probabilities for each detected object being already available during model distillation. During training, we still need to pass each object to ResNet-50 to extract the pre-softmax features. In order to distill the knowledge from the YOLOv3 and ResNet-50 teachers, our student 3D CNN model receives the same input as ResNet-50 and learns to predict the pre-softmax features $Y_{\text{ResNet}}^{(T_4)}$ of ResNet-50 and the class probabilities $Y_{\text{YOLO}}^{(T_4)}$ predicted by YOLOv3. Let $X^{(T_4)}$ be the input image comprising a detected object and $h_{T_4}$ be the knowledge distillation head. The model distillation task is learned by minimizing the $L_1$ loss function:

$$
\mathcal{L}_{T_4} \left( X^{(T_4)}, Y^{(T_4)} \right) = \frac{1}{n} \sum_{j=1}^{n} \left| Y^{(T_4)}_{j} - \hat{Y}^{(T_4)}_{j} \right|,
$$

where $\hat{Y}^{(T_4)} = h_{T_4} \left( f (X^{(T_4)}) \right)$ and $Y^{(T_4)} = Y_{\text{ResNet}}^{(T_4)} \oplus Y_{\text{YOLO}}^{(T_4)}$ is the concatenation of the 1000 ResNet-50 pre-softmax features and the 80 YOLOv3 class probabilities, resulting in a vector of $n = 1080$ components.

**Joint loss.** Our 3D CNN model is trained by jointly optimizing it on the four proxy tasks described above. Hence, the model is training using a joint loss function:

$$
\mathcal{L}_{\text{total}} = \mathcal{L}_{T_1} + \mathcal{L}_{T_2} + \mathcal{L}_{T_3} + \lambda \cdot \mathcal{L}_{T_4},
$$

where $\lambda \in (0, 1]$ is a weight that regulates the importance of the knowledge distillation task. We empirically observed that $\mathcal{L}_{T_4}$ has a typically higher magnitude than the other loss functions, dominating the joint loss without a regularization term. In our experiments, we fine-tune $\lambda$ with respect to the validation values of the joint loss, before ever applying our framework on the anomaly detection task.

### 3.4. Inference

During inference, we utilize YOLOv3 to detect objects in each frame $i$. For each object, we extract the corresponding object-centric sequence $X$ by cropping the bounding box from the frames $\{i - t, \ldots, i - 1, i + 1, \ldots, i + t\}$. We pass each object-centric sequence through our neural model, obtaining the outputs $Y^{(T_1)}$, $Y^{(T_2)}$, $Y^{(T_3)}$ and $Y^{(T_4)}$, respectively. For the arrow of time proxy task, we
take the probability of the temporal sequence to move backward as the anomaly score. For the motion irregularity task, we consider the probability of the gapless test sequence \( X \) to be intermittent as a good abnormality indicator. We interpret the mean absolute error between the reconstructed and the ground-truth middle object as the anomaly score provided by the middle bounding box prediction head. For the knowledge distillation task, we consider the absolute difference between the class probabilities predicted by YOLOv3 and those predicted by our model. We compute the final anomaly score of an object as the average of the anomaly scores given by each prediction head:

\[
score(X) = \frac{1}{4} \left( \hat{Y}^{(T_1)}_2 + \hat{Y}^{(T_2)}_2 + \text{avg} \left( \left| Y^{(T_3)}_2 - \hat{Y}^{(T_3)}_2 \right| \right) + \text{avg} \left( \left| Y^{(T_4)}_{\text{YOLO}} - \hat{Y}^{(T_4)}_{\text{YOLO}} \right| \right) \right)
\]

(6)

Next, we reassemble the detected objects in a pixel-level anomaly map for each frame. Therefore, we can easily localize the anomalous regions in any given frame. To create a smooth pixel-level anomaly map, we apply a 3D mean filter. The anomaly score for a certain frame is given by the maximum score in the corresponding anomaly map. The final frame-level anomaly scores are obtained by applying a temporal Gaussian filter.

### 3.5. Object-Level versus Frame-Level Detection

Although performing anomaly detection at the object level enables the accurate localization of anomalies, the downside is that the detection failures of YOLOv3 (due to a limited set of object categories or poor performance) are translated into false negatives. In order to address this limitation, we can apply our framework at the frame level, eliminating YOLOv3 from the pipeline and keeping the other components in place. By fusing the frame-level and object-level anomaly scores at a late stage, we can recover some of the false negatives of our object-centric framework. In our experiments, we report the results of our framework based on late fusion, as well as the results at the object level and at the frame level, respectively.

### 4. Experiments

#### 4.1. Data Sets

We perform experiments on three benchmark data sets: Avenue [29], ShanghaiTech [31] and UCSD Ped2 [32]. Each data set has pre-defined training and test sets, anomalous events being included only at test time.

**Avenue.** The Avenue [29] data set contains 16 training videos with normal activity and 21 test videos. Examples of anomalous events in Avenue are related to people running, throwing objects or walking in the wrong direction. The resolution of each video is \(360 \times 640\) pixels.

**ShanghaiTech.** ShanghaiTech [31] is one of the largest data sets for anomaly detection in video. It consists of 330 training videos and 107 test videos. The training videos contain only normal events, while the test videos contain normal and abnormal sequences. Examples of anomalous events are: robbing, jumping, fighting and riding bikes in pedestrian areas. The resolution of each video is \(480 \times 586\) pixels.

**UCSD Ped2.** UCSD Ped2 [32] contains 16 training videos with normal activity and 12 test videos. Examples of abnormal events are bikers, skaters and cars in a pedestrian area. The resolution of each video is \(240 \times 360\) pixels.

#### 4.2. Setup and Implementation Details

**Evaluation measures.** As our main evaluation metric, we consider the area under the curve (AUC) computed with respect to the ground-truth frame-level annotations. The frame-level AUC metric is the most commonly used metric in related works [7, 13, 14, 16, 17, 27, 39, 41, 53, 55, 62]. Many related works also report the pixel-level AUC for the UCSD Ped2 data set. As explained by Ramachandra et al. [37], the pixel-level AUC is a flawed evaluation metric. We thus report our performance on UCSD Ped2 in terms of the region-based detection criterion (RBDC) and the track-based detection criterion (TBDC). These metrics were recently introduced by Ramachandra et al. [37] to replace the commonly used pixel-level and frame-level AUC metrics.

**Parameter tuning.** The first step of our framework is object detection based on YOLOv3 [42]. For Avenue and ShanghaiTech, we keep the detections with a confidence higher than 0.8. Because UCSD Ped2 has a lower resolution, we set the detection confidence to 0.5. We use the same confidence threshold during training and inference.

We use the first 85% of the frames in each training video to train our models on the proxy tasks, keeping the last 15% to validate the models on each proxy task. We fine-tune the parameters \(t\) and \(\lambda\) on our validation sets, before making the transition to anomaly detection. For \(t\), we considered values in the set \{1, 2, 3, 4\}. As we obtained optimal results with \(t = 3\), we use this value throughout all the anomaly detection experiments. Hence, an object-centric temporal sequence is a tensor of \(7 \times 64 \times 64 \times 3\) components. We fine-tune the parameter \(\lambda\) controlling the importance of \(L_{T_t}\) in Equation (5), considering values in the set \{0.1, 0.2, 0.5, 1.0\}. We obtained optimal results with \(\lambda = 0.5\) on UCSD Ped2 and \(\lambda = 0.2\) on Avenue and ShanghaiTech, respectively. We therefore report anomaly detection results with these optimal settings.

Each neural network is trained for 30 epochs using the Adam optimizer [22] with a learning rate of \(10^{-5}\), keeping the default values for the other parameters of Adam. We trained the models using mini-batches of 256 samples for the shallow+narrow architecture, 128 samples for the deep+narrow and shallow+wide architectures and 64 samples for the deep+wide architecture, being limited by our computational resources. For each model, we select the checkpoint with the lowest validation error on the proxy task.
tasks to perform anomaly detection.

4.3. Anomaly Detection Results

In Table 2, we present the comparative results of our object-level, frame-level and late fusion frameworks versus the state-of-the-art methods, reporting the frame-level AUC scores (whenever available) on the following three benchmark data sets: Avenue, ShanghaiTech and UCSD Ped2.

**Results on Avenue.** There are only two methods [17, 24] that surpass the 90% threshold on Avenue. Our framework applied at the object level obtains a frame-level AUC of 91.9%, surpassing the state-of-the-art method [17] by 1.5%. When we apply our framework at the frame level, our performance drops considerably, but the method is still able to outperform some recent works [8, 9, 20, 30, 37, 51]. When we fuse the object-level anomaly scores with the frame-level anomaly scores, our performance improves, reaching a new state-of-the-art frame-level AUC of 92.8%. In Figure 2, we illustrate a set of anomaly localization examples along with the frame-level anomaly scores for test video 04. We observe that our approach correlates well with the ground-truth frame-level annotations.

**Results on ShanghaiTech.** On ShanghaiTech, our late fusion method outperforms all previous works, reaching a new state-of-the-art performance of 90.2%, surpassing the previous state-of-the-art method [17] by a margin of 5.3%. Remarkably, we are the first to reach a frame-level AUC score of over 90% on ShanghaiTech. Aside from [17], our method surpasses all other state-of-the-art approaches by a margin of at least 10.9%. In Figure 3, we present some

**Table 2. Frame-level AUC scores (in %) of the state-of-the-art methods [7, 8, 9, 10, 13, 14, 16, 17, 18, 19, 20, 21, 23, 24, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 40, 41, 47, 48, 49, 51, 53, 55, 57, 59, 60, 61, 62, 64] versus our deep+wide architecture trained on four proxy tasks at the object level, at the frame level or based on late fusion. The top two results are shown in red and blue.**
results on UCSD Ped2. UCSD Ped2 is one of the most popular video anomaly detection benchmarks, resulting in 23 works reporting frame-level AUC scores of over 90%. The current state-of-the-art method [53] reports a frame-level AUC of 99.2%. Nevertheless, our method still manages to surpass all previous works, reaching a new state-of-the-art frame-level AUC of 99.8% on UCSD Ped2.

Since RBDC and TBDC are part of a very recent evaluation protocol, there are only two methods [37, 38] that we can compare with in Table 3. We outperform the first method [37] by significant margins in terms of all metrics. We also surpass the second method by 1.9% in terms of TBDC and by 5.8% in terms of frame-level AUC, our RBDC score being slightly lower. These results show that our approach can accurately localize anomalies.

### 4.4. Ablation Study

We perform an ablation study on Avenue and UCSD Ped2 to assess the benefit of including each proxy task in our joint multi-task framework. The corresponding results are presented in Table 4. Along with the anomaly detection performance, we report the performance levels for each proxy task on our validation sets. Considering the individual tasks, we observe that the arrow of time produces the highest frame-level AUC (83.6%) on Avenue, likely because anomalies are caused by unusual actions, e.g. people running. The most suitable tasks for UCSD Ped2 seem to be middle bounding box prediction and knowledge distillation, probably because anomalies are caused by objects with unusual appearance, e.g. bikes or cars. We observe increasingly better anomaly detection results as we gradually add more proxy tasks in our joint optimization framework.

While increasing the number of proxy tasks, we also aim to assess the effect of increasing the width and depth of our neural architecture. We observe performance improvements as we add more layers and filters to our 3D CNN, especially when we jointly optimize on three or four tasks. Hence, we conclude that it is beneficial to increase the learning capacity of the 3D CNN along with the number of proxy tasks.

### 5. Conclusion

In this work, we have proposed a novel anomaly detection method based on self-supervised and multi-task learning, presenting comprehensive results on three benchmarks: Avenue, ShanghaiTech and UCSD Ped2. To our knowledge, our method is the first and only to exceed the 90% threshold on all three benchmarks. Additionally, we performed an ablation study showing the benefits of jointly learning multiple proxy tasks for anomaly detection in video. In future work, we will consider exploring additional proxy tasks to further boost the performance of our multi-task framework.

### Acknowledgments

The research leading to these results has received funding from the EEA Grants 2014-2021, under Project contract no. EEA-RO-NO-2018-0496. This article has also benefited from the support of the Romanian Young Academy, which is funded by Stiftung Mercator and the Alexander von Humboldt Foundation for the period 2020-2022.
References


