

# Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection

Nicolae-Cătălin Ristea<sup>1,2</sup>, Neelu Madan<sup>3</sup>, Radu Tudor Ionescu<sup>4,5,\*</sup>, Kamal Nasrollahi<sup>3,6</sup>, Fahad Shahbaz Khan<sup>2,7</sup>, Thomas B. Moeslund<sup>3</sup>, Mubarak Shah<sup>8</sup>

<sup>1</sup>University Politehnica of Bucharest, Romania, <sup>2</sup>MBZ University of Artificial Intelligence, UAE,

<sup>3</sup>Aalborg University, Denmark, <sup>4</sup>University of Bucharest, Romania, <sup>5</sup>SecurifAI, Romania,

<sup>6</sup>Milestone Systems, Denmark, <sup>7</sup>Linköping University, Sweden, <sup>8</sup>University of Central Florida, US

## Abstract

*Anomaly detection is commonly pursued as a one-class classification problem, where models can only learn from normal training samples, while being evaluated on both normal and abnormal test samples. Among the successful approaches for anomaly detection, a distinguished category of methods relies on predicting masked information (e.g. patches, future frames, etc.) and leveraging the reconstruction error with respect to the masked information as an abnormality score. Different from related methods, we propose to integrate the reconstruction-based functionality into a novel self-supervised predictive architectural building block. The proposed self-supervised block is generic and can easily be incorporated into various state-of-the-art anomaly detection methods. Our block starts with a convolutional layer with dilated filters, where the center area of the receptive field is masked. The resulting activation maps are passed through a channel attention module. Our block is equipped with a loss that minimizes the reconstruction error with respect to the masked area in the receptive field. We demonstrate the generality of our block by integrating it into several state-of-the-art frameworks for anomaly detection on image and video, providing empirical evidence that shows considerable performance improvements on MVTec AD, Avenue, and ShanghaiTech. We release our code as open source at: <https://github.com/ristea/sspcab>.*

## 1. Introduction

Anomaly detection is an important task with a broad set of applications ranging from industrial inspection (finding defects of objects or materials on industrial production lines) [5, 7, 10, 15, 36, 56, 62, 76] to public security (detecting abnormal events such as traffic accidents, fights, explosions, etc.) [12, 13, 17–19, 27, 28, 33, 39, 41, 47–50, 52, 67, 72, 73, 77, 78]. The task is typically framed as a one-class classification (outlier detection) problem, where methods [2, 8, 12, 21, 25, 27, 29, 33, 35, 37, 40, 43–45, 49–

51, 53, 54, 57, 69, 73, 75, 81, 82] learn a familiarity model from normal training samples, labeling unfamiliar examples (outliers) as anomalies, at inference time. Since abnormal samples are available only at test time, supervised learning methods are not directly applicable to anomaly detection. To this end, researchers turned their attention to other directions such as reconstruction-based approaches [15, 19, 21, 36, 37, 43, 47, 49, 54, 62, 69, 71], dictionary learning methods [7–9, 14, 40, 55], distance-based models [6, 10, 25, 27, 50, 51, 53, 57, 58, 63, 65, 68, 70], change detection frameworks [11, 26, 38, 48], and probabilistic models [1, 2, 16, 23, 29, 44, 45, 56, 61, 74].

A distinguished subcategory of reconstruction methods relies on predicting masked information, leveraging the reconstruction error with respect to the masked information as an abnormality score. The masked information can come in different forms, e.g. superpixels [36], future frames [37], middle bounding boxes [17], among others. Methods in this subcategory mask some part of the input and employ a deep neural network to predict the missing input information. Different from such methods, we propose to integrate the capability of reconstructing the masked information into a neural block. Introducing the reconstruction task at a core architectural level has two important advantages: (i) it allows us to mask information at any layer in a neural network (not only at the input), and (ii) it can be integrated into a wide range of neural architectures, thus being very general.

We design our reconstruction block as a self-supervised predictive block formed of a dilated convolutional layer and a channel attention mechanism. The dilated filters are based on a custom receptive field, where the center area of the kernel is masked. The resulting convolutional activation maps are then passed through a channel attention module [24]. The attention module ensures the block does not simply learn to reconstruct the masked region based on linearly interpolating contextual information. Our block is equipped with a loss that minimizes the reconstruction error between the final activation maps and the masked information. In other words, our block is trained to predict the masked information in a self-supervised manner. Our self-supervised

\*corresponding author: raducu.ionescu@gmail.com

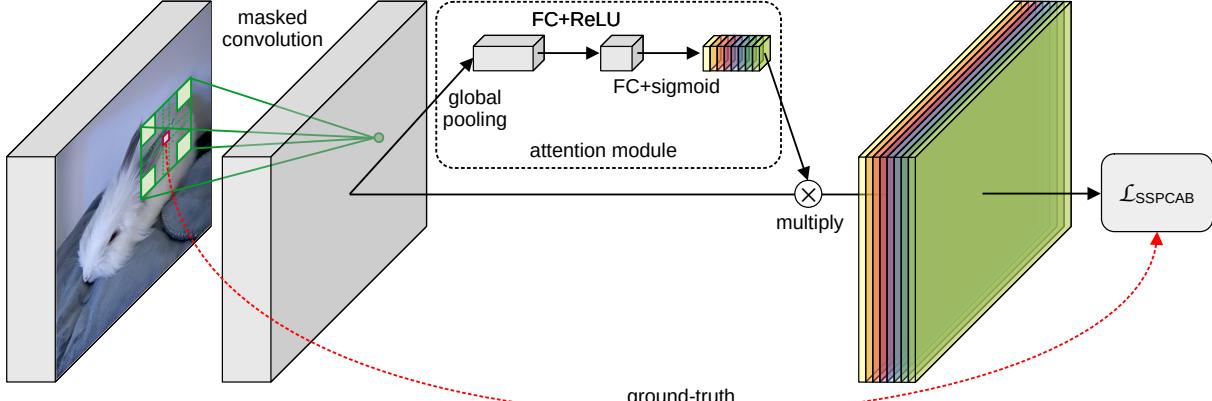


Figure 1. Our self-supervised predictive convolutional attentive block (SSPCAB). For each location where the dilated convolutional filter is applied, the block learns to reconstruct the masked area using contextual information. A channel attention module performs feature recalibration by using global information to selectively emphasize or suppress reconstruction maps. Best viewed in color.

predictive convolutional attentive block (SSPCAB) is illustrated in Figure 1. For each location where the dilated convolutional filter is applied, the block learns to reconstruct the masked area using contextual information. Meanwhile, the dilation rate becomes a natural way to control the context level (from local to global), as required for the specific application.

We integrate SSPCAB into various state-of-the-art anomaly detection frameworks [18, 34, 37, 39, 49, 79] and conduct comprehensive experiments on the MVTec AD [5], Avenue [40] and ShanghaiTech [43] data sets. Our empirical results show that SSPCAB can bring significant performance improvements, *e.g.* the region-based detection criterion (RBDC) of Liu *et al.* [39] on Avenue increases from 41% to 62% by adding SSPCAB. Moreover, with the help of SSPCAB, we are able to report new state-of-the-art performance levels on Avenue and ShanghaiTech. Additionally, we show extra results on the Avenue data set, indicating that the masked convolutional layer can also increase performance levels, all by itself.

In summary, our contribution is twofold:

- We introduce a novel self-supervised predictive convolutional attentive block that is inherently capable of performing anomaly detection.
- We integrate the block into several state-of-the-art neural models [18, 34, 37, 39, 49, 79] for anomaly detection, showing significant performance improvements across multiple models and benchmarks.

## 2. Related Work

As anomalies are difficult to anticipate, methods are typically trained only on normal data, while being tested on both normal and abnormal data [21, 49]. Therefore, outlier detection [25, 27, 50, 51, 53] and self-supervised learning [17–19, 34, 39, 41, 49, 79] approaches are extensively

used to address the anomaly detection task. Anomaly detection methods can be classified into: dictionary learning methods [7–9, 14, 40, 55], change detection frameworks [11, 26, 38, 48], probability-based methods [1, 2, 16, 23, 29, 44, 45, 56, 61, 74], distance-based models [6, 10, 25, 27, 50, 51, 53, 57, 58, 63, 65, 68, 70], and reconstruction-based methods [15, 19, 21, 36, 37, 43, 47, 49, 54, 62, 69, 71, 79].

Dictionary-based methods learn the normal behavior by constructing a dictionary, where each entry in the dictionary represents a normal pattern. Ren *et al.* [55] extended dictionary learning methods by considering the relation among different entries. Change-detection frameworks detect anomalies by quantifying changes across the video frames, *i.e.* a significant deviation from the immediately preceding event marks the beginning of an abnormal event. After quantifying the change, approaches such as unmasking [26] or ordinal regression [48] can be used to segregate anomalies. Probability-based methods build upon the assumption that anomalies occur in a low probability region. These methods estimate the probability density function (PDF) of the normal data and evaluate the test samples based on the PDF. For example, Mahadevan *et al.* [44] used a Mixture of Dynamic Textures (MDTs) to model the distribution of the spatio-temporal domain, while Rudolph *et al.* [56] employed normalizing flow to represent the normal distribution. Distance-based methods learn a distance function based on the assumption that normal events occur in the close vicinity of the learned feature space, while the abnormal events are far apart from the normal data. For instance, Ramachandra *et al.* [51] employed a Siamese network to learn the distance function. Reconstruction-based methods rely on the assumption that the normal examples can be reconstructed more faithfully from the latent manifold. Our new block belongs to the category of reconstruction-based anomaly detection methods, particularly siding with meth-

ods that predict or reconstruct missing (or masked) information [17, 36, 37].

**Reconstruction-based methods.** In the past few years, reconstruction-based methods became prevalent in anomaly detection. Such methods typically use auto-encoders [21] and generative adversarial networks (GANs) [37], as these neural models enable the learning of powerful reconstruction manifolds via using normal data only. However, the generalization capability of neural networks sometimes leads to reconstructing abnormal frames with low error [12, 18], affecting the discrimination between abnormal and normal frames. To address this issue, researchers have tried to improve the latent manifold by diversifying the architecture and training methodologies. Some works focusing on transforming the architectures include memory-based auto-encoders [12, 39, 49], which memorize the normal prototypes in the training data, thus increasing the discrimination between normal and abnormal samples. Other works remodeled the reconstruction manifold via training the models with pseudo-abnormal samples [4, 18, 79]. The adversarial training proposed in [17] applies gradient ascent for out-of-domain pseudo-abnormal samples and gradient descent for normal data, thus learning a more powerful discriminative manifold for video anomaly detection. Zavrtanik *et al.* [79] created pseudo-abnormal samples by adding random noise patches on normal images for image anomaly detection. Some variants of auto-encoders, such as Variational Auto-Encoders (VAEs), have been proposed in [39, 83] for the anomaly detection task. These works are based on the assumption that VAEs can only reconstruct the normal images. Liu *et al.* [39] used a conditional VAE, conditioning the image prediction on optical flow reconstruction, thus accumulating the error from the optical flow reconstruction task with the image prediction. However, this approach can only be applied to video anomaly detection, due to the presence of motion information in the form of optical flow.

**Reconstruction of masked information.** A surrogate task for many anomaly detection approaches [15, 22, 37, 42, 77] is to erase some information from the input, while making neural networks predict the erased information. Haselmann *et al.* [22] framed anomaly detection as an inpainting problem, where patches from images are masked randomly, using the pixel-wise reconstruction error of the masked patches for surface anomaly detection. Fei *et al.* [15] proposed the Attribute Restoration Network (ARNet), which includes an attribute erasing module (AEM) to disorient the model by erasing certain attributes from an image, such as color and orientation. In turn, ARNet learns to restore the original image and detect anomalies based on the assumption that normal images can be restored properly. The Cloze task [42] is about learning to complete a video when certain frames are removed, being recently employed by Yu *et al.* [77] for anomaly detection. In a similar direc-

tion, Georgescu *et al.* [17] proposed middle frame masking as one of the auxiliary tasks for video anomaly detection. Both approaches are based on the assumption that an erased frame can be reconstructed more accurately for regular motion. Future frame prediction [34] utilizes past frames to predict the next frame in the video. The anomaly, in this case, is detected through the prediction error. Another approach based on GANs [59] learns to erase patches from an image, while the discriminator identifies if patches are normal or irregular.

Unlike existing approaches, we are the first to introduce the reconstruction-based functionality as a basic building block for neural architectures. More specifically, we design a novel block based on masked convolution and channel attention to reconstruct a masked part of the convolutional receptive field. As shown in the experiments, our block can be integrated into a multitude of existing anomaly detection frameworks [18, 34, 37, 39, 49, 79], almost always bringing significant performance improvements.

### 3. Method

Convolutional neural networks (CNNs) [30, 31] are widely used across a broad spectrum of computer vision tasks, also being prevalent in anomaly detection [18, 20, 34, 39, 49]. CNNs are formed of convolutional layers equipped with kernels which learn to activate on discriminative local patterns, in order to solve a desired task. The local features extracted by a convolutional layer are combined into more complex features by the subsequent convolutional layers. From this learning process, a hierarchy of features emerges, ranging from low-level features (corners, edges, etc.) to high-level features (car wheels, bird heads, etc.) [80]. While this hierarchy of features is extremely powerful, CNNs lack the ability to comprehend the global arrangement of such local features, as noted by Sabour *et al.* [60].

In this paper, we introduce a novel self-supervised predictive convolutional attentive block (SSPCAB) that is purposed at learning to predict (or reconstruct) masked information using contextual information. To achieve highly accurate reconstruction results, our block is forced to learn the global structure of the discovered local patterns. Thus, it addresses the issue pointed out in [60], namely the fact that CNNs do not grasp the global arrangement of local features, as they do not generalize to novel viewpoints or affine transformations. To implement this behavior, we design our block as a convolutional layer with dilated masked filters, followed by a channel attention module. The block is equipped with its own loss function, which is aimed at minimizing the reconstruction error between the masked input and the predicted output.

We underline that our design is generic, as SSPCAB can be integrated into just about any CNN architecture, being able to learn to reconstruct masked information, while offer-

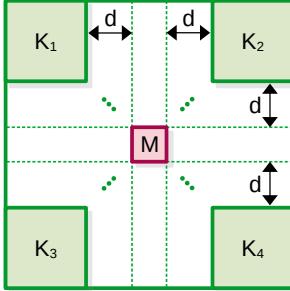


Figure 2. Our masked convolutional kernel. The visible area of the receptive field is denoted by the regions  $\mathbf{K}_i, \forall i \in \{1, 2, 3, 4\}$ , while the masked area is denoted by  $M$ . A dilation factor  $d$  controls the local or global nature of the visible information with respect to  $M$ . Best viewed in color.

ing useful features for subsequent neural layers. Although the capability of learning and using global structure might make SSPCAB useful for a wide range of tasks, we conjecture that our block has a natural and direct applicability in anomaly detection, as explained next. When integrated into a CNN trained on normal training data, SSPCAB will learn the global structure of normal examples only. When presented with an abnormal data sample at inference time, our block will likely provide a poor reconstruction. We can thus measure the quality of the reconstruction and employ the result as a way to differentiate between normal and abnormal examples. In Section 4, we provide empirical evidence to support our claims.

SSPCAB is composed of a masked convolutional layer activated by Rectified Linear Units (ReLU) [46], followed by a Squeeze-and-Excitation (SE) module [24]. We next present its components in more details.

**Masked convolution.** The receptive field of our convolutional filter is depicted in Figure 2. The learnable parameters of our masked convolution are located in the corners of the receptive field, being denoted by the sub-kernels  $\mathbf{K}_i \in \mathbb{R}^{k' \times k' \times c}, \forall i \in \{1, 2, 3, 4\}$ , where  $k' \in \mathbb{N}^+$  is a hyperparameter defining the sub-kernel size and  $c$  is the number of input channels. Each kernel  $\mathbf{K}_i$  is located at a distance (dilation rate)  $d \in \mathbb{N}^+$  from the masked region in the center of our receptive field, which is denoted by  $M \in \mathbb{R}^{1 \times 1 \times c}$ . Consequently, the spatial size  $k$  of our receptive field can be computed as follows:  $k = 2k' + 2d + 1$ .

Let  $X \in \mathbb{R}^{h \times w \times c}$  be the input tensor of our masked convolutional layer, where  $c$  is the number of channels, and  $h$  and  $w$  are the height and width, respectively. The convolutional operation performed with our custom kernel in a certain location of the input  $X$  only considers the input values from the positions where the sub-kernels  $\mathbf{K}_i$  are located, the other information being ignored. The results of the convolution operations between each  $\mathbf{K}_i$  and the corresponding inputs are summed into a single number, as if the sub-kernels  $\mathbf{K}_i$  belong to a single convolutional ker-

nel. The resulting value denotes a prediction located at the same position as  $M$ . Naturally, applying the convolution with one filter produces a single activation map. Hence, we would only be able to predict one value from the masked vector  $M$ , at the current location. To predict a value for every channel in  $M$ , we introduce a number of  $c$  masked convolutional filters, each predicting the masked information from a distinct channel. As we aim to learn and predict the reconstruction for every spatial location of the input, we add zero-padding of  $k' + d$  pixels around the input and set the stride to 1, such that every pixel in the input is used as masked information. Therefore, the spatial dimension of the output tensor  $Z$  is identical to that of the input tensor  $X$ . Finally, the output tensor is passed through a ReLU activation. We underline that the only configurable hyperparameters of our custom convolutional layer are  $k'$  and  $d$ .

**Channel attention module.** Next, the output of the masked convolution is processed by a channel attention module, which computes an attention score for each channel. Knowing that each activation map in  $Z$  is predicted by a separate filter in the presence of masked information, we infer that the masked convolution might end up producing activation maps containing disproportionate (uncalibrated) values across channels. Therefore, we aim to exploit the relationships between channels, with the goal of scaling each channel in  $Z$  in accordance with the quality of the representations produced by the masked convolutional layer. To this end, we employ the channel attention module proposed by Hu *et al.* [24]. The SE module [24] provides a mechanism that performs adaptive recalibration of channel-wise feature responses. Through this mechanism, it can learn to use global information to selectively emphasize or suppress reconstruction maps, as necessary. Another motivation to use attention is to increase the modeling capacity of SSPCAB and enable a non-linear processing between the input and output of our block.

Formally, the channel attention block reduces  $Z$  to a vector  $z \in \mathbb{R}^c$  through a global pooling performed on each channel. Subsequently, the vector of scale factors  $s \in \mathbb{R}^c$  is computed as follows:

$$s = \sigma(\mathbf{W}_2 \cdot \delta(\mathbf{W}_1 \cdot z)), \quad (1)$$

where  $\sigma$  is the sigmoid activation,  $\delta$  is the ReLU activation, and  $\mathbf{W}_1 \in \mathbb{R}^{\frac{c}{r} \times c}$  and  $\mathbf{W}_2 \in \mathbb{R}^{c \times \frac{c}{r}}$  represent the weight matrices of two consecutive fully connected (FC) layers, respectively. The first FC layer consists of  $\frac{c}{r}$  neurons, squeezing the information by a reduction ratio of  $r$ .

Next, the vector  $s$  is replicated in the spatial dimension, generating a tensor  $S$  of the same size as  $Z$ . Our last step is the element-wise multiplication between  $S$  and  $Z$ , producing the final tensor  $\hat{X} \in \mathbb{R}^{h \times w \times c}$  containing recalibrated features maps.

**Reconstruction loss.** We add a self-supervised task consist-

ing of reconstructing the masked region inside our convolutional receptive field, for every location where the masked filters are applied. To this end, our block should learn to provide the corresponding reconstructions as the output  $\hat{\mathbf{X}}$ . Let  $G$  denote the SSPCAB function. We define the self-supervised reconstruction loss as the mean squared error (MSE) between the input and the output, as follows:

$$\mathcal{L}_{\text{SSPCAB}}(G, \mathbf{X}) = (G(\mathbf{X}) - \mathbf{X})^2 = (\hat{\mathbf{X}} - \mathbf{X})^2. \quad (2)$$

When integrating SSPCAB into a neural model  $F$  having its own loss function  $\mathcal{L}_F$ , our loss can simply be added to the respective loss, resulting in a new loss function that comprises both terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_F + \lambda \cdot \mathcal{L}_{\text{SSPCAB}}, \quad (3)$$

where  $\lambda \in \mathbb{R}^+$  is a hyperparameter that controls the importance of our loss with respect to  $\mathcal{L}_F$ . We adopt this procedure when incorporating SSPCAB into various neural architectures during our experiments.

## 4. Experiments and Results

### 4.1. Data Sets

**MVTec AD.** The MVTec AD [5] data set is a standard benchmark for evaluating anomaly detection methods on industrial inspection images. It contains images from 10 object categories and 5 texture categories, having 15 categories in total. There are 3629 defect-free training images and 1725 test images with or without anomalies.

**Avenue.** The CHUK Avenue [40] data set is a popular benchmark for video anomaly detection. It contains 16 training and 21 test videos. The anomalies are present only at inference time and include people throwing papers, running, dancing, loitering, and walking in the wrong direction.

**ShanghaiTech.** The ShanghaiTech [43] benchmark is one of the largest data sets for video anomaly detection. It is formed of 330 training and 107 test videos. As for Avenue, the training videos contain only normal samples, but the test videos can contain both normal and abnormal events. Some examples of anomalies are: people fighting, stealing, chasing, jumping, and riding bike or skating in pedestrian zones.

### 4.2. Evaluation Metrics

**Image anomaly detection.** On MVTec AD, we evaluate methods in terms of the average precision (AP) and the area under the receiver operating characteristic curve (AUROC). The ROC curve is obtained by plotting the true positive rate (TPR) versus the false positive rate (FPR). We consider both localization and detection performance rates. For the detection task, the TPR and FPR values are computed at the image level, *i.e.* TPR is the percentage of anomalous images that are correctly classified, while FPR is the percentage of normal images mistakenly classified as anomalous.

For the localization (segmentation) task, TPR is the percentage of abnormal pixels that are correctly classified, whereas FPR is the percentage of normal pixels wrongly classified as anomalous. To determine the segmentation threshold for each method, we follow the approach described in [5].

**Video anomaly detection.** We evaluate abnormal event detection methods in terms of the area under the curve (AUC), which is computed by marking a frame as abnormal if at least one pixel inside the frame is abnormal. Following [18], we report both the macro and micro AUC scores. The micro AUC is computed after concatenating all frames from the entire test set, while the macro AUC is the average of the AUC scores on individual videos. The frame-level AUC can be an unreliable evaluation measure, as it may fail to evaluate the localization of anomalies [50]. Therefore, we also evaluate models in terms of the region-based detection criterion (RBDC) and track-based detection criterion (TBDC), as proposed by Ramachandra *et al.* [50]. RBDC takes each detected region into consideration, marking a detected region as *true positive* if the Intersection-over-Union with the ground-truth region is greater than a threshold  $\alpha$ . TBDC measures whether abnormal regions are accurately tracked across time. It considers a detected track as *true positive* if the number of detections in a track is greater than a threshold  $\beta$ . Following [18, 50], we set  $\alpha = 0.1$  and  $\beta = 0.1$ .

### 4.3. Implementation Choices and Tuning

For the methods [18, 34, 37, 39, 49, 79] chosen to serve as underlying models for SSPCAB, we use the official code from the repositories provided by the corresponding authors, inheriting the hyperparameters, *e.g.* the number of epochs and learning rate, from each method. Unless specified otherwise, we replace the penultimate convolutional layer with SSPCAB in all underlying models.

In a set of preliminary trials with a basic auto-encoder on Avenue, we tuned the hyperparameter  $\lambda$  from Eq. (3), representing the weight of the SSPCAB reconstruction error, considering values between 0.1 and 1, at a step of 0.1. Based on these preliminary trials, we decided to use  $\lambda = 0.1$  across all models and data sets. However, we observed that  $\lambda = 0.1$  gives a higher than necessary magnitude to our loss for the framework of Liu *et al.* [39]. Hence, for Liu *et al.* [39], we reduced  $\lambda$  to 0.01.

### 4.4. Preliminary Results

We performed preliminary experiments on Avenue to decide the hyperparameters of our masked convolution, *i.e.* the kernel size  $k'$  and dilation rate  $d$ . We consider values in  $\{1, 2, 3\}$  for  $k'$ , and values in  $\{0, 1, 2\}$  for  $d$ . In addition, we consider two alternative loss functions, namely the Mean Absolute Error (MAE) and Mean Squared Error (MSE), and several types of attention to be added after the masked convolution, namely channel attention (CA), spatial

Method	Loss type	$d$	$k'$	$r$	Attention type	AUC		RBDC	TBDC
						Micro	Macro		
Plain auto-encoder	-	-	-	-	-	80.0	83.4	49.98	51.69
	MAE	0	1	-	-	83.3	84.1	47.46	52.11
	MSE	1	1	-	-	83.9	84.6	49.05	52.21
		2	1	-	-	83.2	84.3	48.56	52.03
	MSE	0	1	-	-	83.6	84.2	47.86	52.21
		1	1	-	-	84.2	84.9	49.22	52.29
		2	1	-	-	83.6	84.3	48.44	51.98
	MSE	0	2	-	-	83.7	84.0	47.41	53.02
		1	2	-	-	84.0	85.1	48.22	51.84
		2	2	-	-	82.7	83.1	46.94	50.22
	MSE	0	3	-	-	82.6	83.7	48.28	51.91
		1	3	-	-	82.9	84.7	48.13	52.07
		2	3	-	-	83.1	83.8	47.13	49.96
	MSE	1	1	8	CA	<b>85.9</b>	<b>85.6</b>	53.81	<b>56.33</b>
		1	1	-	SA	84.3	84.4	53.31	53.41
		1	1	8	CA+SA	85.7	85.6	<b>53.98</b>	54.11
	MSE	1	1	4	CA	85.6	85.3	53.83	55.99
	MSE	1	1	16	CA	84.4	84.9	53.28	54.37

Table 1. Micro AUC, macro AUC, RBDC and TBDC scores (in %) obtained on the Avenue data set with different hyperparameter configurations, *i.e.* kernel size ( $k'$ ), dilation rate ( $d$ ), reduction ratio ( $r$ ), loss type, and attention type, for our SSPCAB. Results are obtained by introducing SSPCAB into a plain auto-encoder that follows the basic architecture designed by Georgescu *et al.* [18]. Best results are highlighted in bold.

attention (SA), and both (CA+SA).

For the preliminary experiments, we take the appearance convolutional auto-encoder from [18] as our baseline, stripping out the additional components such as optical flow, skip connections, adversarial training, mask reconstruction and binary classifiers. Our aim is to test various SSPCAB configurations on top of a basic architecture, without trying to overfit the configuration to a specific framework, such as that of Georgescu *et al.* [18]. To this end, we decided to remove the aforementioned components, thus using only a plain auto-encoder in our preliminary experiments.

The preliminary results are presented in Table 1. Upon adding the masked convolutional layer based on the MAE loss on top of the basic architecture, we observe significant performance gains, especially for  $k' = 1$  and  $d = 1$ . The performance further increases when we replace the MAE loss function with MSE. We performed extensive experiments with different combinations of  $k'$  and  $d$ , obtaining better results with  $k' = 1$  and  $d = 1$ . We therefore decided to fix the loss to MSE, the sub-kernel size  $k'$  to 1, and the dilation rate  $d$  to 1, for all subsequent experiments. Next, we introduced various attention modules after our masked convolution. Among the considered attention modules, we observe that channel attention is the one that better complements our masked convolutional layer, providing the highest performance gains for three of the metrics: 5.9% for the

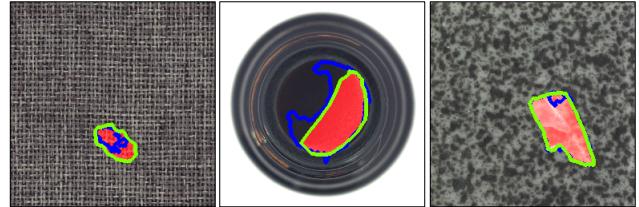


Figure 3. Anomaly localization examples of DRAEM [79] (blue) versus DRAEM+SSPCAB (green) on MVTec AD. The ground-truth anomalies are marked with a red mask. Best viewed in color.

micro AUC, 2.2% for the macro AUC, and 4.6% for TBDC. Accordingly, we selected the channel attention module for the remaining experiments. Upon choosing to use channel attention, we test additional reduction rates ( $r = 4$  and  $r = 16$ ), without observing any improvements. As such, we keep the reduction rate of the SE module to  $r = 8$ , whenever we integrate SSPCAB into a neural model.

#### 4.5. Anomaly Detection in Images

**Baselines.** We choose two recent models for image anomaly detection, *i.e.* CutPaste [34] and DRAEM [79].

Li *et al.* [34] proposed CutPaste, a simple data augmentation technique that cuts a patch from an image and pastes it to a random location. The CutPaste architecture is built on top of GradCAM [64]. The model is based on a self-supervised 3-way classification task, learning to classify samples into normal, CutPaste and CutPaste-Scar, where a scar is a long and thin mark of a random color. Li *et al.* [34] also used an ensemble of five 3-way CutPaste models trained with different random seeds to improve results.

Zavrtanik *et al.* [79] introduced DRAEM, a method based on a dual auto-encoder for anomaly detection and localization on MVTec AD. We introduce SSPCAB into both the localization and detection networks.

**Results.** We present the results on MVTec AD in Table 2. Considering the detection results, we observe that SSPCAB brings consistent performance improvements on most categories for both CutPaste [34] and DRAEM [79]. Moreover, the overall performance gains in terms of detection AUROC are close to 1%, regardless of the underlying model. Given that the baselines are already very good, we consider the improvements brought by SSPCAB as noteworthy.

Considering the localization results, it seems that SSPCAB is not able to improve the overall AUROC score of DRAEM [79]. However, the more challenging AP metric tells a different story. Indeed, SSPCAB increases the overall AP of DRAEM [79] by 1.5%, from 68.4% to 69.9%.

In Figure 3, we illustrate a few anomaly localization examples where SSPCAB introduces significant changes to the anomaly localization contours of DRAEM [79], showing a higher overlap with the ground-truth anomalies. We believe that these improvements are a direct effect induced

	Class	Localization				Detection					
		DRAEM [79]				DRAEM [79]		CutPaste [34]			
		+SSPCAB		+SSPCAB		+SSPCAB		3-way		Ensemble	
		AUROC	AUROC	AP	AP	AUROC	AUROC	AUROC	AUROC	AUROC	AUROC
Texture	Carpet	<b>95.5</b>	95.0	53.5	<b>59.4</b>	97.0	<b>98.2</b>	<b>93.1</b>	90.7	93.9	<b>96.8</b>
	Grid	<b>99.7</b>	99.5	<b>65.7</b>	61.1	99.9	<b>100.0</b>	99.9	99.9	<b>100.0</b>	99.9
	Leather	98.6	<b>99.5</b>	75.3	<b>76.0</b>	100.0	100.0	100.0	100.0	100.0	100.0
	Tile	99.2	<b>99.3</b>	92.3	<b>95.0</b>	99.6	<b>100.0</b>	93.4	<b>94.0</b>	94.6	<b>95.0</b>
	Wood	96.4	<b>96.8</b>	77.7	77.1	99.1	<b>99.5</b>	98.6	<b>99.2</b>	99.1	99.1
Object	Bottle	<b>99.1</b>	98.8	86.5	<b>87.9</b>	<b>99.2</b>	98.4	98.3	<b>98.6</b>	98.2	<b>99.1</b>
	Cable	94.7	<b>96.0</b>	52.4	<b>57.2</b>	91.8	<b>96.9</b>	80.6	<b>82.9</b>	81.2	<b>83.6</b>
	Capsule	<b>94.3</b>	93.1	49.4	<b>50.2</b>	98.5	<b>99.3</b>	96.2	<b>98.1</b>	<b>98.2</b>	97.6
	Hazelnut	99.7	<b>99.8</b>	92.9	92.6	100.0	100.0	97.3	<b>98.3</b>	98.3	<b>98.4</b>
	Metal Nut	<b>99.5</b>	98.9	96.3	<b>98.1</b>	98.7	<b>100.0</b>	99.3	<b>99.6</b>	99.9	99.9
	Pill	<b>97.6</b>	97.5	48.5	<b>52.4</b>	98.9	<b>99.8</b>	92.4	<b>95.3</b>	94.9	<b>96.6</b>
	Screw	97.6	<b>99.8</b>	58.2	<b>72.0</b>	93.9	<b>97.9</b>	86.3	<b>90.8</b>	88.7	<b>90.8</b>
	Toothbrush	98.1	98.1	44.7	<b>51.0</b>	100.0	100.0	98.3	<b>98.8</b>	99.4	<b>99.6</b>
	Transistor	<b>90.9</b>	87.0	<b>50.7</b>	48.0	<b>93.1</b>	92.9	95.5	<b>96.5</b>	96.1	<b>97.3</b>
	Zipper	98.8	<b>99.0</b>	<b>81.5</b>	77.1	100.0	100.0	<b>99.4</b>	99.1	99.9	99.9
Overall		<b>97.3</b>	97.2	68.4	<b>69.9</b>	98.0	<b>98.9</b>	95.2	<b>96.1</b>	96.1	<b>96.9</b>

Table 2. Localization AUROC/AP and detection AUROC (in %) of state-of-the-art methods on MVTec AD, before and after adding SSPCAB. The best result for each before-versus-after pair is highlighted in bold.

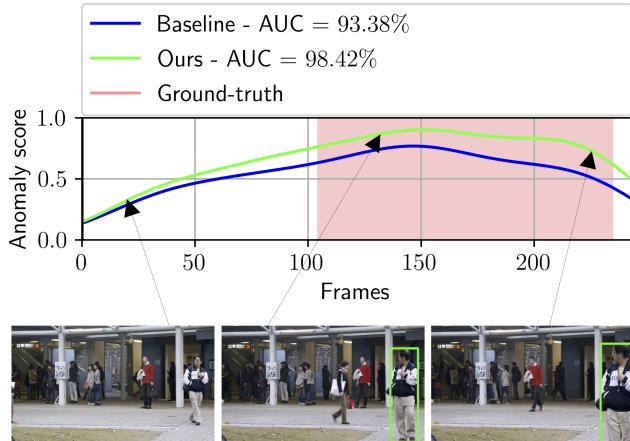


Figure 4. Frame-level anomaly scores for Liu *et al.* [37] before (baseline) and after (ours) integrating SSPCAB, for test video 18 from Avenue. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

by the reconstruction errors produced by our novel block. We provide more anomaly detection examples in the supplementary.

#### 4.6. Abnormal Event Detection in Video

**Baselines.** We choose four recently introduced methods [18, 37, 39, 49] attaining state-of-the-art performance levels in video anomaly detection, as candidates for integrating SSPCAB. We first reproduce the results using the official implementations provided by the corresponding authors [18, 37, 39, 49]. We refrain from making any modifi-

cation to the hyperparameters of the chosen baselines. Despite using the unmodified code from the official repositories, we were not able to exactly reproduce the results of Liu *et al.* [39] and Park *et al.* [49], but our numbers are very close. As we add SSPCAB into the reproduced models, we consider the reproduced results as reference. We underline that, for Georgescu *et al.* [18], we integrate SSPCAB into the auto-encoders, not in the binary classifiers. We report RBDC and TBDC results whenever possible, computing the scores using the implementation provided by Georgescu *et al.* [18].

**Results.** We report the results on Avenue and ShanghaiTech in Table 3. First, we observe that the inclusion of SSPCAB in the framework of Liu *et al.* [37] brings consistent improvements over all metrics on both benchmarks. Similarly, we observe consistent performance gains when integrating SSPCAB into the model of Park *et al.* [49]. We note that the method of Park *et al.* [49] does not produce anomaly localization results, preventing us from computing the RBDC and TBDC scores for their method. SSPCAB also brings consistent improvements for Liu *et al.* [39], the only exception being the macro AUC on Avenue. For this baseline [39], we observe a remarkable increase of 21.22% in terms of the RBDC score on Avenue. Finally, we notice that SSPCAB also improves the performance of the approach proposed by Georgescu *et al.* [18] for almost all metrics, the exceptions being the TBDC on Avenue and RBDC on ShanghaiTech. In summary, we conclude that integrating SSPCAB is beneficial, regardless of the underlying model. Moreover, due to the integration of SSPCAB, we are able

Venue	Method	Avenue			ShanghaiTech		
		AUC		RBDC	TBDC	AUC	
		Micro	Macro			Micro	Macro
BMVC 2018	Liu <i>et al.</i> [38]	84.4	-	-	-	-	-
	Sultani <i>et al.</i> [66]	-	-	-	-	76.5	-
	Lee <i>et al.</i> [32]	87.2		-	-	76.2	
WACV 2019 ICCV 2019 CVPR 2019 TNNLS 2019 TIP 2019	Ionescu <i>et al.</i> [27]	88.9	-	-	-	-	-
	Nguyen <i>et al.</i> [47]	86.9	-	-	-	-	-
	Ionescu <i>et al.</i> [25]	87.4	90.4	15.77	27.01	78.7	84.9
	Wu <i>et al.</i> [73]	86.6		-	-	-	-
	Lee <i>et al.</i> [33]	90.0		-	-	-	-
ACMMM 2020 WACV 2020 WACV 2020 PRL 2020 Access 2020 CVPRW 2020 ACMMM 2020 ACMMM 2020	Yu <i>et al.</i> [77]	89.6	-	-	74.8	-	-
	Ramachandra <i>et al.</i> [50]	72.0		35.80	80.90	-	-
	Ramachandra <i>et al.</i> [51]	87.2		41.20	78.60	-	-
	Tang <i>et al.</i> [69]	85.1		-	-	73.0	
	Dong <i>et al.</i> [12]	84.9		-	-	73.7	
	Doshi <i>et al.</i> [13]	86.4		-	-	71.6	
	Sun <i>et al.</i> [67]	89.6		-	-	74.7	
	Wang <i>et al.</i> [72]	87.0		-	-	79.3	
ICCVW 2021 BMVC 2021 CVPR 2021	Astrid <i>et al.</i> [4]	84.7	-	-	73.7	-	-
	Astrid <i>et al.</i> [3]	87.1	-	-	75.9	-	-
	Georgescu <i>et al.</i> [17]	91.5	92.8	57.00	58.30	82.4	90.2
CVPR 2018 CVPR 2022	Liu <i>et al.</i> [37]	85.1	81.7	19.59	56.01	72.8	80.6
	Liu <i>et al.</i> [37] + SSPCAB	<b>87.3</b>	<b>84.5</b>	<b>20.13</b>	<b>62.30</b>	<b>74.5</b>	<b>82.9</b>
CVPR 2020 CVPR 2022	Park <i>et al.</i> [49]	82.8	86.8	-	-	68.3	79.7
	Park <i>et al.</i> [49] + SSPCAB	<b>84.8</b>	<b>88.6</b>	-	-	<b>69.8</b>	<b>80.2</b>
ICCV 2021 CVPR 2022	Liu <i>et al.</i> [39]	89.9	<b>93.5</b>	41.05	86.18	74.2	83.2
	Liu <i>et al.</i> [39] + SSPCAB	<b>90.9</b>	92.2	<b>62.27</b>	<b>89.28</b>	<b>75.5</b>	<b>83.7</b>
TPAMI 2021 CVPR 2022	Georgescu <i>et al.</i> [18]	92.3	90.4	65.05	<b>66.85</b>	82.7	89.3
	Georgescu <i>et al.</i> [18] + SSPCAB	<b>92.9</b>	<b>91.9</b>	<b>65.99</b>	64.91	<b>83.6</b>	<b>89.5</b>

Table 3. Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech. Among the existing models, we select four models [18, 37, 39, 49] to show results before and after including SSPCAB. The best result for each before-versus-after pair is highlighted in bold. The top score for each metric is shown in red.

to report new state-of-the-art results on Avenue and ShanghaiTech, for several metrics.

In Figure 4, we compare the frame-level anomaly scores on test video 18 from Avenue, before and after integrating SSPCAB into the method of Liu *et al.* [37]. On this video, SSPCAB increases the AUC by more than 5%. We observe that the approach based on SSPCAB can precisely localize and detect the abnormal event (*person walking in the wrong direction*). We provide more anomaly detection examples in the supplementary.

## 5. Conclusion

In this paper, we introduced SSPCAB, a novel neural block composed of a masked convolutional layer and a channel attention module, which predicts a masked region in the convolutional receptive field. Our neural block is trained in a self-supervised manner, via a reconstruction loss of its own. To show the benefit of using SSPCAB in anomaly detection, we integrated our block into a series of image and video anomaly detection methods

[18, 34, 37, 39, 49, 79]. Our empirical results indicate that SSPCAB brings performance improvements in almost all cases. The preliminary results show that both the masked convolution and the channel attention contribute to the performance gains. Furthermore, with the help of SSPCAB, we are able to obtain new state-of-the-art levels on Avenue and ShanghaiTech. We consider this as a major achievement.

In future work, we aim to extend SSPCAB by replacing the masked convolution with a masked 3D convolution. In addition, we aim to consider other application domains besides anomaly detection.

## 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 work has also been funded by the Milestone Research Programme at AAU, SecurifAI, and the Romanian Young Academy, which is funded by Stiftung Mercator and the Alexander von Humboldt Foundation for the period 2020-2022.

## References

- [1] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and David Reinitz. Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008. [1](#), [2](#)
- [2] Borislav Antic and Bjorn Ommer. Video parsing for abnormality detection. In *Proceedings of ICCV*, pages 2415–2422, 2011. [1](#), [2](#)
- [3] Marcella Astrid, Muhammad Zaigham Zaheer, Jae-Yeong Lee, and Seung-Ik Lee. Learning not to reconstruct anomalies. In *Proceedings of BMVC*, 2021. [8](#)
- [4] Marcella Astrid, Muhammad Zaigham Zaheer, and Seung-Ik Lee. Synthetic Temporal Anomaly Guided End-to-End Video Anomaly Detection. In *Proceedings of ICCVW*, pages 207–214, 2021. [3](#), [8](#)
- [5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Proceedings of CVPR*, pages 9592–9600, 2019. [1](#), [2](#), [5](#)
- [6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *Proceedings of CVPR*, pages 4183–4192, 2020. [1](#), [2](#)
- [7] Diego Carrera, Fabio Manganini, Giacomo Boracchi, and Ettore Lanzarone. Defect Detection in SEM Images of Nanofibrous Materials. *IEEE Transactions on Industrial Informatics*, 13(2):551–561, 2017. [1](#), [2](#)
- [8] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. In *Proceedings of CVPR*, pages 2909–2917, 2015. [1](#), [2](#)
- [9] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Proceedings of CVPR*, pages 3449–3456, 2011. [1](#), [2](#)
- [10] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audiger. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In *Proceedings of ICPR*, pages 475–489, 2021. [1](#), [2](#)
- [11] Allison Del Giorno, J. Andrew Bagnell, and Martial Hebert. A Discriminative Framework for Anomaly Detection in Large Videos. In *Proceedings of ECCV*, pages 334–349, 2016. [1](#), [2](#)
- [12] Fei Dong, Yu Zhang, and Xiushan Nie. Dual Discriminator Generative Adversarial Network for Video Anomaly Detection. *IEEE Access*, 8:88170–88176, 2020. [1](#), [3](#), [8](#)
- [13] Keval Doshi and Yasin Yilmaz. Any-Shot Sequential Anomaly Detection in Surveillance Videos. In *Proceedings of CVPRW*, pages 934–935, 2020. [1](#), [8](#)
- [14] Jayanta K. Dutta and Bonny Banerjee. Online Detection of Abnormal Events Using Incremental Coding Length. In *Proceedings of AAAI*, pages 3755–3761, 2015. [1](#), [2](#)
- [15] Ye Fei, Chaoqin Huang, Cao Jinkun, Maosen Li, Ya Zhang, and Cewu Lu. Attribute Restoration Framework for Anomaly Detection. *IEEE Transactions on Multimedia*, pages 1–1, 2020. [1](#), [2](#), [3](#), [14](#)
- [16] Yachuang Feng, Yuan Yuan, and Xiaoqiang Lu. Learning deep event models for crowd anomaly detection. *Neurocomputing*, 219:548–556, 2017. [1](#), [2](#)
- [17] Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. Anomaly Detection in Video via Self-Supervised and Multi-Task Learning. In *Proceedings of CVPR*, pages 12742–12752, 2021. [1](#), [2](#), [3](#), [8](#)
- [18] Mariana Iuliana Georgescu, Radu Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [14](#)
- [19] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton Van Den Hengel. Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection. In *Proceedings of ICCV*, pages 1705–1714, 2019. [1](#), [2](#), [14](#)
- [20] Xin Guo, Zhongming Jin, Chong Chen, Helei Nie, Jianqiang Huang, Deng Cai, Xiaofei He, and Xiansheng Hua. Discriminative-Generative Dual Memory Video Anomaly Detection. *arXiv preprint arXiv:2104.14430*, 2021. [3](#)
- [21] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, and Larry S. Davis. Learning temporal regularity in video sequences. In *Proceedings of CVPR*, pages 733–742, 2016. [1](#), [2](#), [3](#), [14](#)
- [22] Matthias Haselmann, Dieter P. Gruber, and Paul Tabatabai. Anomaly detection using deep learning based image completion. *Proceedings of ICMLA*, pages 1237–1242, 2018. [3](#)
- [23] Ryota Hinami, Tao Mei, and Shin’ichi Satoh. Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge. In *Proceedings of ICCV*, pages 3639–3647, 2017. [1](#), [2](#)
- [24] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation Networks. In *Proceedings of CVPR*, pages 7132–7141, 2018. [1](#), [4](#)
- [25] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video. In *Proceedings of CVPR*, pages 7842–7851, 2019. [1](#), [2](#), [8](#)
- [26] Radu Tudor Ionescu, Sorina Smeureanu, Bogdan Alexe, and Marius Popescu. Unmasking the abnormal events in video. In *Proceedings of ICCV*, pages 2895–2903, 2017. [1](#), [2](#)
- [27] Radu Tudor Ionescu, Sorina Smeureanu, Marius Popescu, and Bogdan Alexe. Detecting abnormal events in video using Narrowed Normality Clusters. In *Proceedings of WACV*, pages 1951–1960, 2019. [1](#), [2](#), [8](#)
- [28] Xiangli Ji, Bairong Li, and Yuesheng Zhu. TAM-Net: Temporal Enhanced Appearance-to-Motion Generative Network for Video Anomaly Detection. In *Proceedings of IJCNN*, pages 1–8, 2020. [1](#)
- [29] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In *Proceedings of CVPR*, pages 2921–2928, 2009. [1](#), [2](#)

- [30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of NIPS*, pages 1106–1114, 2012. 3
- [31] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3
- [32] Sangmin Lee, Hak Gu Kim, and Yong Man Ro. STAN: Spatio-temporal adversarial networks for abnormal event detection. In *Proceedings of ICASSP*, pages 1323–1327, 2018. 8
- [33] Sangmin Lee, Hak Gu Kim, and Yong Man Ro. BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection. *IEEE Transactions on Image Processing*, 29:2395–2408, 2019. 1, 8
- [34] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In *Proceedings of CVPR*, pages 9664–9674, 2021. 2, 3, 5, 6, 7, 8, 12, 14
- [35] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014. 1
- [36] Zhenyu Li, Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection. In *Proceedings of BMVC*, 2020. 1, 2, 3, 14
- [37] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future Frame Prediction for Anomaly Detection – A New Baseline. In *Proceedings of CVPR*, pages 6536–6545, 2018. 1, 2, 3, 5, 7, 8, 12, 13, 14
- [38] Yusha Liu, Chun-Liang Li, and Barnabaás Póczos. Classifier Two-Sample Test for Video Anomaly Detections. In *Proceedings of BMVC*, 2018. 1, 2, 8
- [39] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction. In *Proceedings of ICCV*, pages 13588–13597, 2021. 1, 2, 3, 5, 7, 8, 12, 14
- [40] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal Event Detection at 150 FPS in MATLAB. In *Proceedings of ICCV*, pages 2720–2727, 2013. 1, 2, 5
- [41] Yiwei Lu, Frank Yu, Mahesh Kumar, Krishna Reddy, and Yang Wang. Few-Shot Scene-Adaptive Anomaly Detection. In *Proceedings of ECCV*, pages 125–141, 2020. 1, 2
- [42] Dezhao Luo, Chang Liu, Y. Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning. In *Proceedings of AAAI*, pages 11701–11708, 2020. 3
- [43] Weixin Luo, Wen Liu, and Shenghua Gao. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In *Proceedings of ICCV*, pages 341–349, 2017. 1, 2, 5, 14
- [44] Vijay Mahadevan, Wei-Xin LI, Viral Bhalodia, and Nuno Vasconcelos. Anomaly Detection in Crowded Scenes. In *Proceedings of CVPR*, pages 1975–1981, 2010. 1, 2
- [45] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Proceedings of CVPR*, pages 935–942, 2009. 1, 2
- [46] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of ICML*, pages 807–814, 2010. 4
- [47] Trong-Nguyen Nguyen and Jean Meunier. Anomaly Detection in Video Sequence With Appearance-Motion Correspondence. In *Proceedings of ICCV*, pages 1273–1283, 2019. 1, 2, 8, 14
- [48] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection. In *Proceedings of CVPR*, pages 12173–12182, 2020. 1, 2
- [49] Hyunjong Park, Jongyoun Noh, and Bumsup Ham. Learning Memory-guided Normality for Anomaly Detection. In *Proceedings of CVPR*, pages 14372–14381, 2020. 1, 2, 3, 5, 7, 8, 12, 14
- [50] Bharathkumar Ramachandra and Michael Jones. Street Scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of WACV*, pages 2569–2578, 2020. 1, 2, 5, 8
- [51] Bharathkumar Ramachandra, Michael Jones, and Ranga Vatsavai. Learning a distance function with a Siamese network to localize anomalies in videos. In *Proceedings of WACV*, pages 2598–2607, 2020. 1, 2, 8
- [52] Bharathkumar Ramachandra, Michael J. Jones, and Ranga Raju Vatsavai. A Survey of Single-Scene Video Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1
- [53] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, and Nicu Sebe. Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection. In *Proceedings of WACV*, pages 1689–1698, 2018. 1, 2
- [54] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal Event Detection in Videos using Generative Adversarial Nets. In *Proceedings of ICIP*, pages 1577–1581, 2017. 1, 2, 14
- [55] Huamin Ren, Weifeng Liu, Soren Ingvar Olsen, Sergio Escalera, and Thomas B. Moeslund. Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection. In *Proceedings of BMVC*, pages 28.1–28.13, 2015. 1, 2
- [56] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows. In *Proceedings of WACV*, pages 1907–1916, 2021. 1, 2
- [57] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017. 1, 2
- [58] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018. 1, 2

- [59] Mohammad Sabokrou, Masoud PourReza, Mohsen Fayyaz, Rahim Entezari, Mahmood Fathy, Juergen Gall, and Ehsan Adeli. AVID: Adversarial Visual Irregularity Detection. In *Proceedings of ACCV*, pages 488–505, 2018. 3
- [60] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic Routing Between Capsules. In *Proceedings of NIPS*, pages 3859–3869, 2017. 3
- [61] Babak Saleh, Ali Farhadi, and Ahmed Elgammal. Object-Centric Anomaly Detection by Attribute-Based Reasoning. In *Proceedings of CVPR*, pages 787–794, 2013. 1, 2
- [62] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution Knowledge Distillation for Anomaly Detection. In *Proceedings of CVPR*, pages 14902–14912, 2021. 1, 2, 14
- [63] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *Proceedings of CVPR*, pages 2112–2119, 2012. 1, 2
- [64] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of ICCV*, pages 618–626, 2017. 6
- [65] Sorina Smeureanu, Radu Tudor Ionescu, Marius Popescu, and Bogdan Alexe. Deep Appearance Features for Abnormal Behavior Detection in Video. In *Proceedings of ICIAP*, volume 10485, pages 779–789, 2017. 1, 2
- [66] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-World Anomaly Detection in Surveillance Videos. In *Proceedings of CVPR*, pages 6479–6488, 2018. 8
- [67] Che Sun, Yunde Jia, Yao Hu, and Yuwei Wu. Scene-Aware Context Reasoning for Unsupervised Abnormal Event Detection in Videos. In *Proceedings of ACMMM*, pages 184–192, 2020. 1, 8
- [68] Qianru Sun, Hong Liu, and Tatsuya Harada. Online growing neural gas for anomaly detection in changing surveillance scenes. *Pattern Recognition*, 64(C):187–201, Apr. 2017. 1, 2
- [69] Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, and Jian Yang. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters*, 129:123–130, 2020. 1, 2, 8, 14
- [70] Hanh T.M. Tran and David Hogg. Anomaly Detection using a Convolutional Winner-Take-All Autoencoder. In *Proceedings of BMVC*, 2017. 1, 2
- [71] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *Proceedings of ECCV*, pages 485–503, 2020. 1, 2, 14
- [72] Ziming Wang, Yuexian Zou, and Zeming Zhang. Cluster Attention Contrast for Video Anomaly Detection. In *Proceedings of ACMMM*, pages 2463–2471, 2020. 1, 8
- [73] Peng Wu, Jing Liu, and Fang Shen. A Deep One-Class Neural Network for Anomalous Event Detection in Complex Scenes. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2609–2622, 2019. 1, 8
- [74] Shandong Wu, Brian E. Moore, and Mubarak Shah. Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes. In *Proceedings of CVPR*, pages 2054–2060, 2010. 1, 2
- [75] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion. *Computer Vision and Image Understanding*, 156:117–127, 2017. 1
- [76] Jihun Yi and Sungroh Yoon. Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation. In *Proceedings of ACCV*, pages 375–390, 2020. 1
- [77] Guang Yu, Siqi Wang, Zhiping Cai, En Zhu, Chuanfu Xu, Jianping Yin, and Marius Kloft. Cloze Test Helps: Effective Video Anomaly Detection via Learning to Complete Video Events. In *Proceedings of ACMMM*, pages 583–591, 2020. 1, 3, 8
- [78] Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, and Seung-Ik Lee. Old is Gold: Redefining the Adversarially Learned One-Class Classifier Training Paradigm. In *Proceedings of CVPR*, pages 14183–14193, 2020. 1
- [79] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. DRAEM – A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In *Proceedings of ICCV*, pages 8330–8339, 2021. 2, 3, 5, 6, 7, 8, 12, 14
- [80] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Proceedings of ECCV*, pages 818–833, 2014. 3
- [81] Xinfeng Zhang, Su Yang, Jiulong Zhang, and Weishan Zhang. Video Anomaly Detection and Localization using Motion-field Shape Description and Homogeneity Testing. *Pattern Recognition*, page 107394, 2020. 1
- [82] Bin Zhao, Li Fei-Fei, and Eric P. Xing. Online Detection of Unusual Events in Videos via Dynamic Sparse Coding. In *Proceedings of CVPR*, pages 3313–3320, 2011. 1
- [83] David Zimmerer, Simon Kohl, Jens Petersen, Fabian Isensee, and Klaus Maier-Hein. Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection. In *Proceedings of MIDL*, 2019. 3

	Location of SSPCAB			AUC		RBDC	TBDC
	Early	Middle	Late	Micro	Macro		
Plain auto-encoder				80.0	83.4	49.98	51.69
	✓			81.1	83.6	50.86	52.44
		✓		84.2	85.0	52.73	54.02
			✓	85.9	85.6	53.81	56.33
	✓	✓		82.7	83.8	50.54	52.70
	✓		✓	83.2	84.1	52.33	53.01
		✓	✓	<b>86.1</b>	<b>85.7</b>	<b>54.03</b>	56.07
	✓	✓	✓	85.3	85.4	53.11	<b>56.64</b>

Table 4. Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) on Avenue, while integrating SSPCAB into an auto-encoder, at different locations. SSPCAB improves the results regardless of the integration place or the number of blocks. The option highlighted in red is used throughout the experiments presented in the main article. Best results are highlighted in bold.

Size of $M$	AUC		RBDC	TBDC
	Micro	Macro		
	80.0	83.4	49.98	51.69
$1 \times 1$	85.9	85.6	53.81	56.33
$3 \times 3$	85.9	85.5	53.93	56.31

Table 5. Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) on Avenue, while varying the size of the masked kernel  $M$ .

## 6. Supplementary

### 6.1. Ablation Study

In the main article, we mention that we generally replace the penultimate convolutional layer with SSPCAB in underlying models [18, 34, 37, 39, 49, 79]. Ideally, for optimal performance gains, the integration place and the number of SSPCAB modules should be tuned on a validation set for each framework. However, anomaly detection data sets do not have a validation set and there is no way to obtain one from the training set, as the training contains only normal examples. In this context, to fairly demonstrate the generality and utility of SSPCAB, we only used a single configuration (one block, closer to the output) across all existing frameworks. However, adding more modules could be beneficial. To test various configurations, we perform an ablation study on the number of SSPCAB modules and the places where these modules can be integrated in a plain auto-encoder. In Table 4, we present the corresponding experiments on the Avenue data set. *We observe that SSPCAB improves the results, regardless of the place of integration or the number of blocks.* The improvements seem larger when SSPCAB is integrated closer to the output. Integrating more blocks can sometimes help.

Another hyperparameter that could be tuned is the size of the masked kernel  $M$ . In our experiments, we kept  $M$

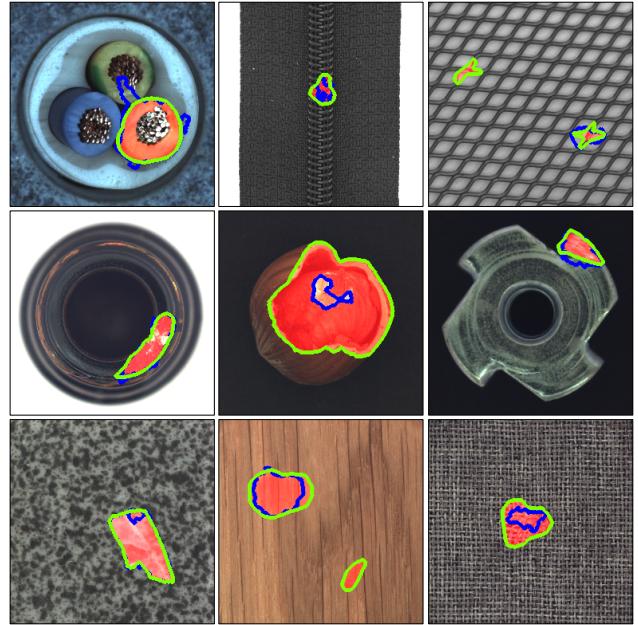


Figure 5. Additional anomaly localization examples of DRAEM [79] (blue) versus DRAEM+SSPCAB (green) on MVTec AD. The ground-truth anomalies are marked with a red mask. Best viewed in color.

to a size of  $1 \times 1$  for simplicity and speed. To study the effect of increasing the size of  $M$ , we have tested the size of  $3 \times 3$  with the plain auto-encoder on Avenue. We report the corresponding results in Table 5. When comparing the results with masked kernels of  $1 \times 1$  or  $3 \times 3$  components, we do not observe significant differences.

An additional aspect that can suffer multiple reconfigurations, given a validation set, is the pattern of the proposed kernel. In our experiments, we tried a simple pattern where the mask is placed in the center and the reception field is connected to the four corner sub-kernels denoted by  $K_i$ ,  $\forall i \in \{1, 2, 3, 4\}$ . We designed this pattern while trying to extrapolate the idea from middle frame prediction (which was shown to provide somewhat better results than future frame prediction) to a 2D kernel. Of course, other patterns are possible and are likely to work equally well.

### 6.2. Qualitative Anomaly Detection Results

**Anomaly detection in images.** In Figure 5, we present additional qualitative results produced by DRAEM [79] on the MVTec AD benchmark. The displayed examples illustrate the benefit of integrating SSPCAB, which is much better at segmenting the anomalies compared to the baseline DRAEM. We show improvements in terms of the pixel-level annotation for both objects and textures.

**Anomaly detection in videos.** In Figure 6, we show a com-

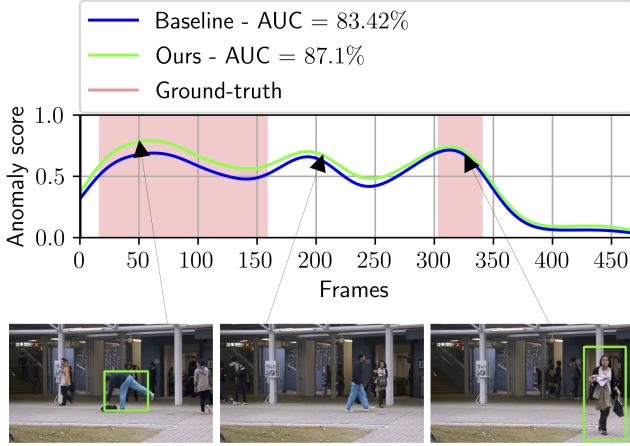


Figure 6. Frame-level anomaly scores for Liu *et al.* [37] before (baseline) and after (ours) integrating SSPCAB, for test video 10 from Avenue. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

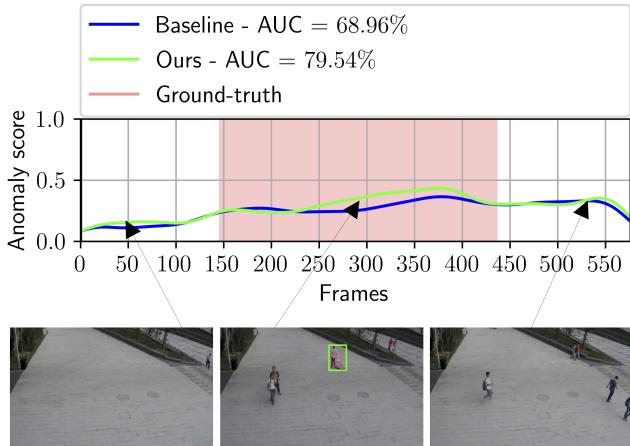


Figure 7. Frame-level anomaly scores for Georgescu *et al.* [18] before (baseline) and after (ours) integrating SSPCAB, for test video 01\_0054 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

parison of the frame-level anomaly scores on test video 10 from the Avenue data set, before and after integrating SSPCAB into the method of Liu *et al.* [37]. On this video, SSPCAB increases the AUC by nearly 4%. After introducing SSPCAB, we observe higher frame-level anomaly scores for the first abnormal event. The anomaly localization results depict *a person throwing a backpack* and *a person walking in the wrong direction*.

In Figures 7 and 8, we illustrate similar comparisons for test videos 01\_0054 and 01\_0130 from the ShanghaiTech

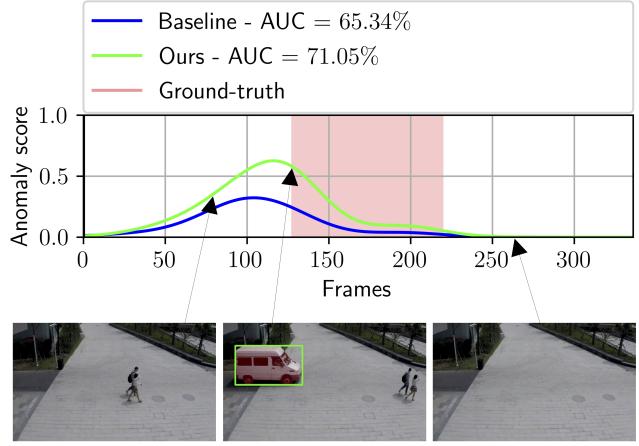


Figure 8. Frame-level anomaly scores for Georgescu *et al.* [18] before (baseline) and after (ours) integrating SSPCAB, for test video 01\_0130 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

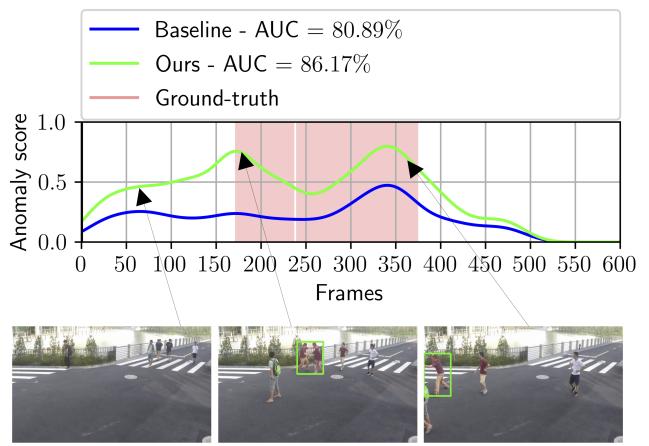


Figure 9. Frame-level anomaly scores for Georgescu *et al.* [18] before (baseline) and after (ours) integrating SSPCAB, for test video 07\_0047 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

data set, before and after adding SSPCAB into the framework of Georgescu *et al.* [18]. For test video 01\_0054, SSPCAB increases the AUC by more than 10%. For test video 01\_0130, the baseline framework seems to detect the abnormal event too early, but SSPCAB seems capable of shifting the detection towards the correct moment. As a result, SSPCAB increases the frame-level AUC score by almost 6%. We observe a similar AUC improvement from SSPCAB in Figure 9, where we compare the frame-level anomaly scores on test video 07\_0047 from the ShanghaiTech data set. For

Method	Time (ms)		Relative (%)
	Baseline	+SSPCAB	
Liu <i>et al.</i> [37]	2.1	2.4	14.2
Georgescu <i>et al.</i> [18]	1.5	1.7	13.3

Table 6. Inference times (in milliseconds) and relative time expansions (in %) for two frameworks [18, 37], before and after integrating SSPCAB. The running times are measured on an Nvidia GeForce GTX 3090 GPU with 24 GB of VRAM.

this video, we underline that the frame-level scores are visibly more correlated to the ground-truth anomalies. Moreover, in all three ShanghaiTech videos, we observe that the approach based on SSPCAB can precisely localize and detect the abnormal events (*person pulling a lever cart, car inside pedestrian area, people fighting, people running*).

### 6.3. Inference Time

Regardless of the underlying framework [18, 34, 37, 39, 49, 79], we add only one instance of SSPCAB, usually replacing the penultimate convolutional layer. As such, we expect the running time to increase. To assess the amount of extra time added by SSPCAB, we present the running times before and after integrating SSPCAB into two state-of-the-art frameworks [18, 37] in Table 6. The reported times show time expansions lower than 0.3 ms for both frameworks. Hence, we consider that the accuracy gains brought by SSPCAB outweigh the marginal running time expansions observed in Table 6.

### 6.4. Discussion

Although SSPCAB belongs to an existing family of anomaly detection methods, *i.e.* reconstruction-based frameworks [15, 19, 21, 36, 37, 43, 47, 49, 54, 62, 69, 71], we would like to underline that we are the first to integrate the reconstruction functionality at the block level. Unlike other reconstruction approaches, our contribution is more flexible, as it can be integrated in existing and future reconstruction methods. Moreover, SSPCAB can also be used to introduce reconstruction-based anomaly detection in other frameworks, which do not rely on reconstruction. We thus believe that our generic and effective approach will help ease future research in anomaly detection.

An important aspect that must be noted is that, due to the masked convolution, our block will not reconstruct the input exactly. Except for the degenerate case where the input is constant, this scenario should not occur in the real world, which means that the reconstruction performed by SSPCAB is not trivial. However, our foremost intuition about the usefulness of SSPCAB is different: our block provides a better reconstruction for normal convolutional features than for abnormal convolutional features. If the features representing normal versus abnormal examples are different at any

layer of a neural architecture, it should result in greater differences at the final output of the architecture. This idea is also supported by the experiments presented in Table 4.

Further looking at the results shown in Table 4, we observe that SSPCAB does not bring significant gains when the block is placed near the input. We aim to further investigate this limitation in future work. Aside from this small issue, we did not observe other limitations of SSPCAB during our experiments.