Investigating 3D Convolutional Layers as Feature Extractors for Anomaly Detection Systems Applied to Surveillance Videos

Tiago S. Nazare, Rodrigo F. de Mello and Moacir A. Ponti
ICMC, Universidade de São Paulo, São Carlos, Brazil

Keywords: Video Surveillance, Unusual Event Detection, Deep Learning.

Abstract: Over the last few years, several strategies have been leveraged to detect unusual behavior in surveillance videos. Nonetheless, there are still few studies that compare strategies based on 3D Convolutional Neural Networks to tackle such problem. This research gap has motivated the this work in which we aim at investigating the features from a pre-trained C3D model and the training of fully 3D-convolutional auto-encoders for automated video anomaly detection systems, comparing them with respect to the anomaly detection performance and the processing power demands. Additionally, we present an auto-encoder model to detect anomalous behavior based on the pixel reconstruction error. While C3D features coming from the first layers were shown to be both better descriptors and faster to be computed, the auto-encoder achieved results comparable to the C3D, while requiring less computational effort. When compared to other studies using two benchmark datasets, the proposed methods are comparable to the state-of-the-art for the Ped2 dataset, while inferior when detecting anomalies on the Ped1 dataset. Additionally, our experimental results support the development of future 3D-CNN-based anomaly detection methods.

1 INTRODUCTION

One of the main applications of automated video analysis systems is the surveillance of public places such as squares, malls and airports (Mabrouk and Zagrouba, 2018). In this particular domain, a computer vision model can assist humans in attempt to avoid missing out suspicious activities and, therefore, make surveillance more effective. Such assistance is needed due to two main reasons: (i) usually a single person is in charge of monitoring several video feeds at the same time, which increases the chance of missing out some important event (Dee and Velastin, 2008); and (ii) it has been shown that the attention of an individual drastically decreases just after monitoring surveillance cameras for 20 minutes (Haering et al., 2008). Therefore, a model that, for instance, highlights cameras with higher probability of having some abnormal event would be of great help.

In this context, a great deal of computer vision methods have been proposed to detect abnormal events in surveillance videos using approaches such as: optical-flow features (Colque et al., 2015; Adam et al., 2008), Generative Adversarial Networks – GANs (Ravanbakhsh et al., 2017), time series decomposition (Ponti et al., 2017a), auto-encoders (Xu et al., 2015), dictionary learning (Li et al., 2015), pre-trained CNNs (Nazare et al., 2018b), (Ravanbakhsh et al., 2018), (dos Santos et al., 2019) and contextual approaches (Colque et al., 2018). Those methods improved the results over time, nevertheless, most – specially the ones based on CNNs – do not take into account the amount of processing power required to perform such task. This can pose a problem in situations where processing resources are limited, for instance, the video may need to be processed using the video recorder (e.g. a regular desktop) due to bandwidth limitations (Muthusenthil and Kim, 2018). Also, many studies do not justify well the use of specific deep network backbones and architectures.

Motivated by the aforementioned issues, we devote our efforts to **better understand the usage of 3D CNNs for abnormal behavior detection in security videos**. In order to do so, we first use a pre-trained C3D model (Tran et al., 2015) as feature extractor for abnormal behavior detection in security videos. We look at how the features generated by the convolutional part of a pre-trained C3D model affects both anomaly detection and computational requirements from two angles: (i) when we vary the input shape of the video, and (ii) when we extract features from different convolutional layers. Therefore, this paper

Nazare, T., F. de Mello, R. and Ponti, M.
Investigating 3D Convolutional Layers as Feature Extractors for Anomaly Detection Systems Applied to Surveillance Videos.
DOI: 10.5220/0010347907530762
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 753-762
ISBN: 978-989-758-488-6

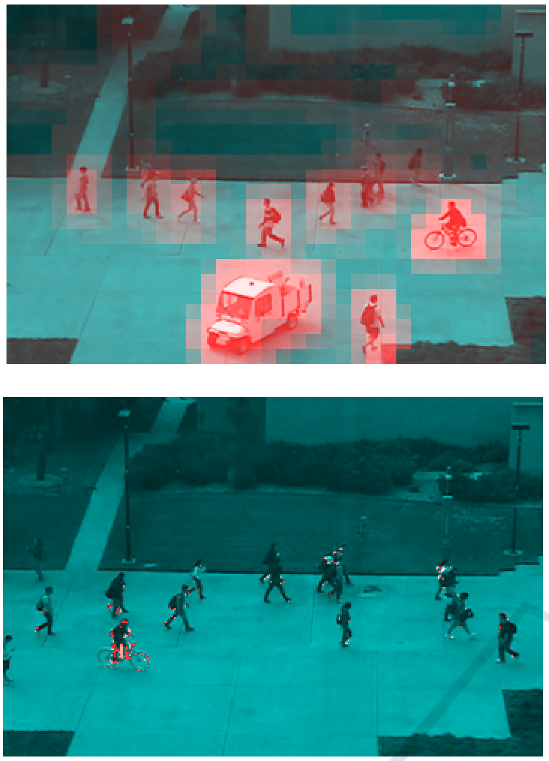


Figure 1: Anomaly score detection strategy comparison for the two proposed methods. The first image illustrate the results from C3D features which are based on local feature distances (less precise), while the second image shows the pixel-level reconstruction provided by the auto-encoder (more precise).

does not intent to overcome state-of-the-art results, which often employ more sophisticated training and post-processing mechanisms, but rather **shed light on the use of 3D convolutions as a way to better represent video data**, so that future work can make better use of such models as backbones to improve results.

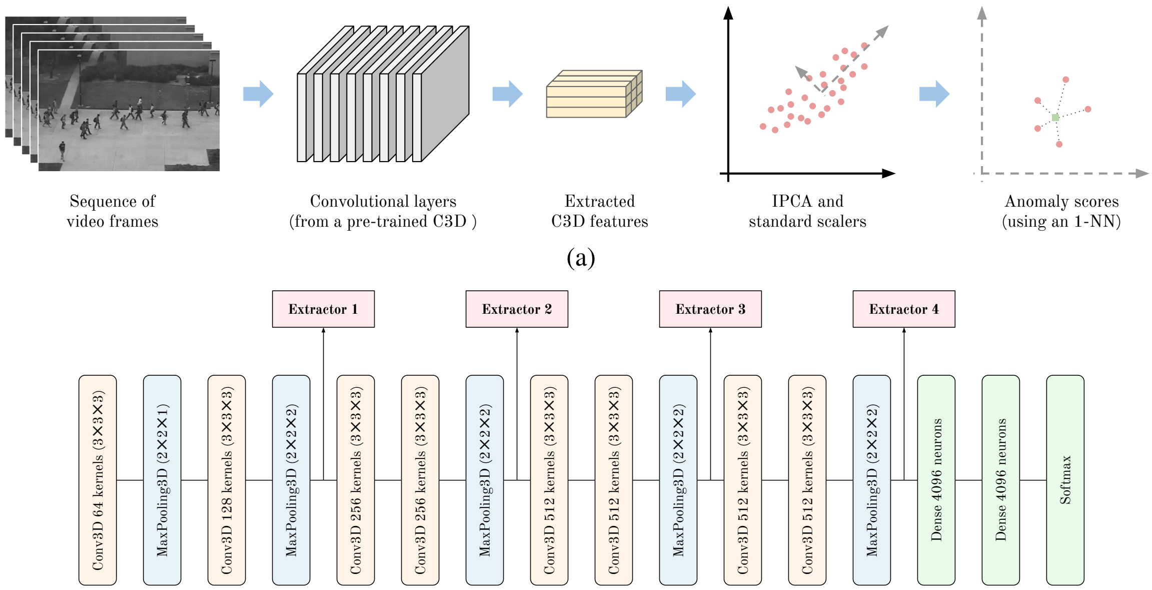
We present novel empirical evidence on this application including that: (i) extracting features from bottom layers of the network, i.e. closer to the input layer, improves the anomaly detection, while reducing the processing demands to generate features; (ii) reduction of video resolution produces a small loss in anomaly detection performance, while making the feature extraction considerably faster; (iii) a domain-specific auto-encoder (which only uses 3D convolutions) is capable of obtaining similar performance when compared to the C3D features, while have a much lower computational complexity. Figure 1 illustrates both approaches using heatmaps of the local features obtained by C3D and AE-3D for anomaly detection, as detailed in the next sections.

2 RELATED WORK

Noticeably, convolutional neural networks have turned into the main framework to solve various computer vision problems (Ponti et al., 2017b) (Goodfellow et al., 2016). One of the main reasons for that is the ability for transfer learning, in which a model that was trained in some dataset/domain is used on a new and, sometimes rather different, domain (Kornblith et al., 2019) (Dos Santos et al., 2020) . In such case, the pre-trained network is leveraged as a feature extractor (Razavian et al., 2014) or as weight initialization in attempt to improve the starting point for training on a target application (Kornblith et al., 2019; Yosinski et al., 2014).

One of the applications that has benefited from the usage of transfer learning is the detection of anomalous events in surveillance. For instance, in (Ravanbakhsh et al., 2018), the image region description generated by a pre-trained AlexNet (Krizhevsky et al., 2012) is analyzed over time to detect unusual behaviors. In (Sultani et al., 2018), the authors used a pre-trained C3D model as proposed by (Tran et al., 2015) (originally trained for action recognition) as a feature extractor to detect unusual behavior on surveillance videos. Nonetheless, in their study, the model was used in classification setup what means that samples from the anomalous class were used during training. In (Nazare et al., 2018b), the authors compared the features generated by several CNNs trained on the ImageNet dataset (Deng et al., 2009) to detect anomalies in security videos, while (dos Santos et al., 2019) evaluated the ability of 2D CNNs to generalize across different video surveillance datasets. The previous studies extracted features from each individual frame and did not use any kind of temporal tracking, their model only use the instant appearance to detect anomalies.

Although those studies achieved good results with regards to the detection of abnormal behavior in security videos – as pointed out previously – they may not be suited for situations in which the amount of processing power is limited. Considering this gap, in this paper, we investigate three ways of reducing the amount of computation needed to leverage 3D convolution in surveillance applications. The first approach, which is based on the findings of (Yosinski et al., 2014), is to extract feature from convolutional layers that are closer to the input of the network. By doing so, we are able to reduce the processing demands involved in extracting features (given that we have fewer layers to execute a forwarding pass) and we may also get better features (as pointed out in (Yosinski et al., 2014)). The second approach com-


Sequence of video frames
Convolutional layers (from a pre-trained C3D)
Extracted C3D features
IPCA and standard scalers
Anomaly scores (using an 1-NN)
Extractor 1
Extractor 2
Extractor 3
Extractor 4
Conv3D 64 kernels (3×3×3)
MaxPooling3D (2×2×1)
Conv3D 128 kernels (3×3×3)
MaxPooling3D (2×2×2)
Conv3D 256 kernels (3×3×3)
Conv3D 256 kernels (3×3×3)
MaxPooling3D (2×2×2)
Conv3D 512 kernels (3×3×3)
Conv3D 512 kernels (3×3×3)
MaxPooling3D (2×2×2)
Conv3D 512 kernels (3×3×3)
Conv3D 512 kernels (3×3×3)
MaxPooling3D (2×2×2)
Dense 4096 neurons
Dense 4096 neurons
Softmax

(a)

(b)

Figure 2: Experimental setup diagram for C3D feature extraction.

prises the reduction of the frame sizes before passing them through the network, therefore reducing the amount of pixels to be processed and, consequently, the computation time required to generate features. The third approach builds up a domain-specific autoencoder based only on 3D convolutions and use frame its reconstruction error as an anomaly score.

3 EXPERIMENTAL SETUP

3.1 C3D Features

We start our experiments by using the experimental setup presented in Figure 2 to better understand the video anomaly detection of features extracted from a pre-trained C3D network. On one hand, regarding video anomaly detection (see Figure 2 (a)), uses ideas presented in (Kornblith et al., 2019) but for the video-surveillance scenario . On the other hand, concerning the study of possible features to be extracted from the pre-trained C3D model, our study is based on the findings in (Yosinski et al., 2014).

In the **first step** of our setup, we extract features from video segments of 16 consecutive frames by performing a forward pass such video segments through convolutional layers of a pre-trained C3D model. As a result of this process, we obtain tensors that contain the descriptions of $32 \times 32 \times 16$ regions of the video segment, which is used to detect anomalies in video regions/frames.

Given the four different C3D layers that we are going to use to extract features from $32 \times 32 \times 16$ video regions, we need to interpreted the output tensor differently for each case (layer). For example, when extracting features from the last convolutional layer of the C3D model (Extractor 4 in Figure 2 (b)), we get an $M \times N \times 512$ tensor in which we have a 512-dimensional feature vector to describe $M \times N$ video regions of size $32 \times 32 \times 16$ pixels. On the other hand, when carrying out this process for Extractor 3, each $32 \times 32 \times 16$ region is going to generate a $2 \times 2 \times 2 \times 512$ descriptor. In Table 1, we present the number of feature generate by the four different extractors when describing a video region with $32 \times 32 \times 16$ pixels. Since Extractors 1, 2 and 3 generate a large number of features, we applied a 3D pooling on their spatial dimension in order to reduce them to one and speedup the anomaly detection process (see the third column of Table 1).

Table 1: Number of features generated by each extractor when describing a $32 \times 32 \times 16$ video region (before and after pooling).

Extractor	**# of raw features**	**# of features after pooling**
1	$8 \times 8 \times 8 \times 128 = 65536$	128
2	$4 \times 4 \times 4 \times 256 = 16384$	256
3	$2 \times 2 \times 2 \times 512 = 4096$	512
4	$1 \times 1 \times 1 \times 512 = 512$	512

In the **second step** of our setup, we reduce the number of features using the Incremental PCA

(IPCA) model (Ross et al., 2008). Additionally, we apply a standard scaler before the IPCA transformation and another one after it. By applying these preprocessing steps, we reduce the number of features and make them follow a mean equals to zero and a standard deviation equals to one. This helps the approximate nearest neighbor method that we employed to detect anomalies (Muja and Lowe, 2014) in order to achieve faster inference time and give the same importance to all features.

Lastly, in the **third step** of our setup, we use our nearest neighbor model to compute the Euclidean distance of each new video region to the closest sample in the training data. Such distance is considered as an anomaly scores, where greater distance values indicate a greater chance of being an anomaly.

The setup described above allow us to compare features generated by four different C3D extractors with respect to their suitability as features for video anomaly detection scenarios and their computational requirements. Our experiments cover the hyperparameter variations present in in Table 2 that were selected to explore both the anomaly detection results and the computational cost of the pre-trained C3D model. Finally, we compare the results of all these experiments using the frame-level anomaly detection results (AUC and EER) on the Ped1 and the Ped2 datasets.

Table 2: Values tested for each hyperparameter.

Parameter	Values
Pooling type	average, max
Frame resolution	192×128, 384×256
IPCA output dimensions	8, 16, 32, 64, 128

3.2 Auto-encoder

Next, based on the good video anomaly detection results of the auto-encoders present in (Xu et al., 2015), we designed a fully 3D-convolutional auto-encoder architecture to model normal behavior in surveillance applications. Our network architecture, which is depicted in Figure 3, uses: $3 \times 3 \times 3$ kernels in all convolutional layers, `ReLU` activation function in all layers, $2 \times 2 \times 2$ size for both the max-pooling and upsampling operations, Batch Normalization in every layer, Gaussian noise to corrupt the input video segment and MSE as the loss function.

Such model is trained to reconstruct $20 \times 20 \times 8$ video segments extracted from training videos containing only normal behaviors. During the anomaly detection phase, the trained auto-encoder is used to reconstruct video segments and the reconstruction error of each pixel is used as pixel-level anomaly score.

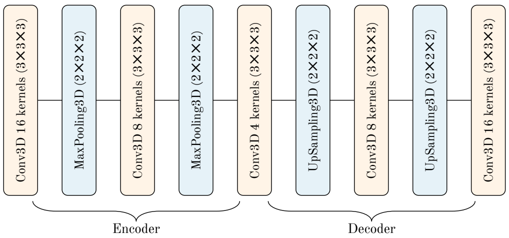


Figure 3: The auto-encoder architecture proposed.

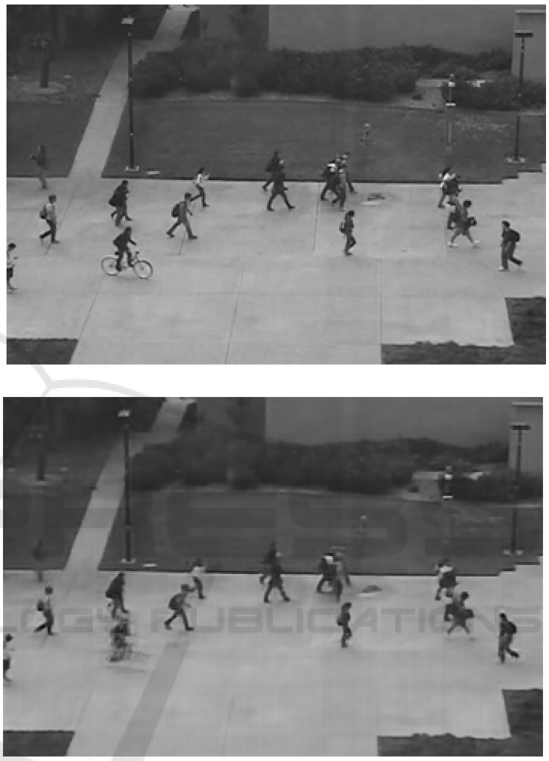


Figure 4: Frame reconstructed by our 3D auto-encoder architecture. The first image shows the original frame, while the second presents its reconstructed version.

To illustrate the usefulness of such reconstruction error, Figure 4 presents such process on a video frame containing a single anomaly. Inspecting the resulting image, we can notice that the anomaly (a biker) was poorly reconstructed, while the remaining of the frame is rather similar to the original one.

3.3 Surveillance Video Datasets

Our experiments are based on the two UCSD datasets – Ped1 and Ped2 – which were both obtained from security footage from the University of California San Diego (Mahadevan et al., 2010; Li et al., 2014). This means that, differently from other surveillance sets, these datasets were obtained from real situations and, therefore, are of great value to estimate the detection performance of automated surveillance systems. Those datasets remain important benchmarks, since they describe a similar problems (detecting anomalies in a pedestrian environment), but having different resolution, perspective, and types of anomalies. In particular Ped1 is more challenging while Ped2 is more well-behaved. In Table 3 and Figure 5, we present, respectively, some characteristics and some frames of these two datasets.

Table 3: Dataset characteristics.

Characteristic	**Dataset**	
	Ped1	**Ped2**
# training videos	34 videos	16 videos
# test videos	36 videos	12 videos
resolution	238 × 158	360 × 240
fps	10 fps	10 fps

3.4 Reproducibility Remarks

In order to extract features from the security videos using a pre-trained C3D model and create our auto-encoder model, we have used the `Keras` (Chollet et al., 2015) library. With regards to the training the IPCA model to reduce the number of features, we have employed the implementation available on the `Scikit-learn` (Pedregosa et al., 2011) library. Finally, to estimate the distance of each new sample to the training set, we have used the approximate nearest neighbor method implemented on the `FLANN` (Muja and Lowe,) library. In order to make it easier to reproduce our results, our source code was made publicly available[1].

4 RESULTS AND DISCUSSION

4.1 Pre-trained C3D Network

Based on our C3D feature extraction setup, we experimented with different parameters. In our first experiment, we tried to determine which configuration – regarding frame resolution, pooling method and number of dimensions after the IPCA transformation – provides the best results on the Ped1 and Ped2 datasets. The results of our C3D experiments are presented in Table 4 and indicate that features from Extractor 2 obtained overall better results. This is interesting mainly due to the following reasons:

1. As the original C3D training dataset and our target datasets (*Ped1* and *Ped2*) are different, the more generic features from Extractor 2 end up being more suitable for the anomaly detection task; and
2. When comparing to Extractors 3 and 4, Extractor 2 requires a forward pass throughout less C3D layers and, consequently, less processing power to generate its features.

Regarding the usage of different pooling methods, we noticed that combining an average pooling with the C3D usually leads to better anomaly detection results. Such phenomenon is particularly evident when we compared the AUC values obtained by the two pooling methods using some boxplots (see Figure 6). Also, in most cases, keeping a higher number of IPCA dimensions (i.e. 64 or 128) generates better anomaly detection results.

4.2 3D Auto-encoder

Next, we tested our auto-encoder architecture with and without background subtraction. The results obtained are shown in Table 5 and indicate that background subtraction does not makes a substantial difference on the anomaly detection performance. When comparing against the C3D results, we can see that the auto-encoder performed better on *Ped1* (while lacking behind regarding state-of-the-art results) and worse on *Ped2* (still being comparable to the best results for that dataset). Nonetheless, as shown in Table 6, it is important to notice that our auto-encoder uses far less parameters than the C3D feature extractors.

4.3 Discussion

We emphasize our aim is not to present SOTA results, but investigate how far can we get with 3D-convolution-based deep networks, that can be leveraged as backbones or components of more sophisticated methods. When comparing C3D and AE-3D results to classic and state-of-the-art methods on both USCD datasets (Ped1 and Ped2) as in Table 7, we notice that both C3D and AE3D methods for learning representations achieved remarkable results on Ped2, specially regarding AUC comparisons even when compared to recent work (Park et al., 2020). On the other hand, on the Ped1 experiments, our results are far from the state-of-the-art and only comparable to classic approaches (specially the ones from the models based on C3D features).

Another interesting observation is that, for Ped1 even CNN-2D features (Nazare et al., 2018b) were better than C3D pre-trained features, showing the positional learning of 3D convolutions from previous

[1] Our source code is available at https://github.com/tiagosn/c3d_features_anomaly_detection.

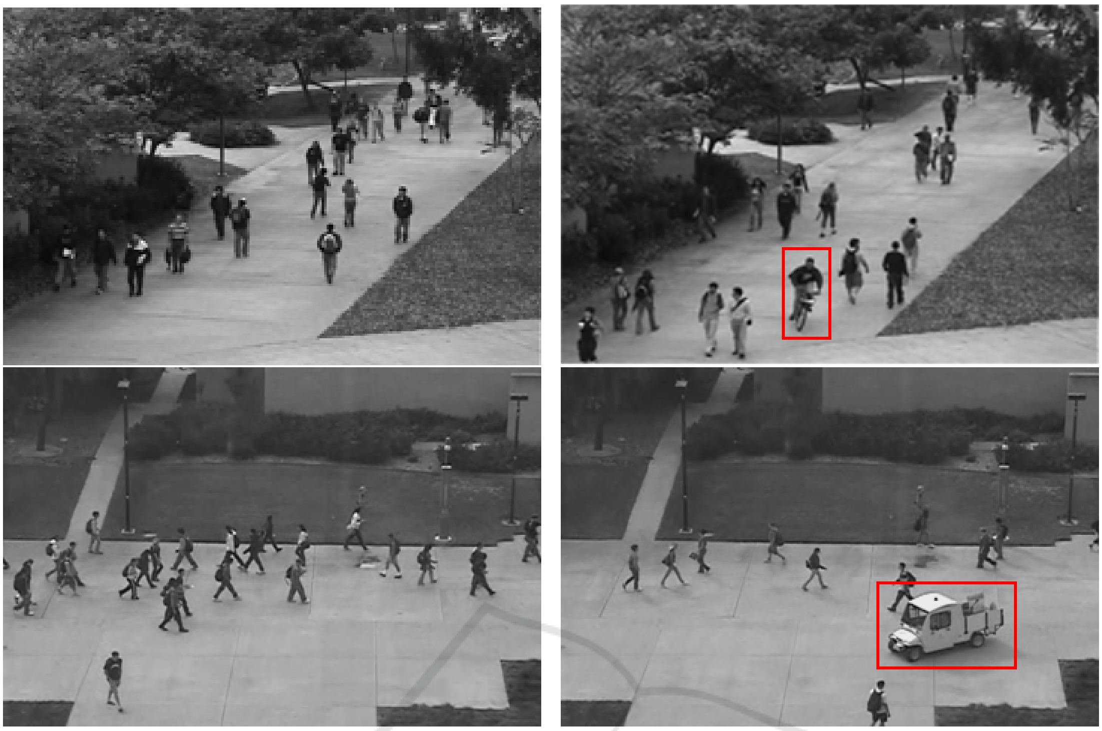


Figure 5: Some examples of frames from the two security video datasets. In the first row, we have frames from the Ped1 dataset, while the second shows frames from the Ped2 dataset. The anomalies are highlighted by red boxes.

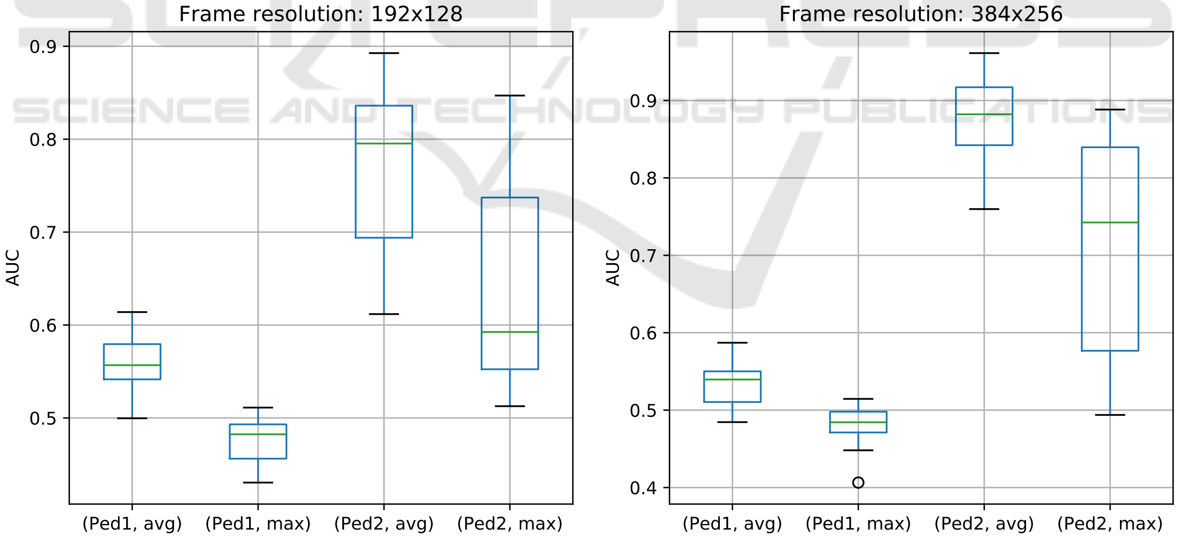


Figure 6: Pooling method comparison. These results show that just by using average pooling instead of max pooling as the last layer of the feature extractor, we can greatly improve anomaly detection results.

tasks may not be adequate when there is significant relative distance between objects and camera.

This comparison also clearly indicates where the learning process must become task-oriented: in a more well behaved scenario such as Ped2, the use of well selected features from a pre-trained network can achieve results close to the state-of-the-art. On the other hand, Ped1 frames has variation on both speed and scale features of objects on the scene, which makes it harder to flag anomalies correctly. Thus, it is paramount to design systems that address such scenarios and this is why region-based methods works better. However, by learning dataset-specific models, and compensating too much for regions of the frame,

Table 4: Results obtained from the frame-level detection on the Ped1 and the Ped2 datasets. This results where obtained using several different hyperparameter configurations (feature extractor, pooling method, number of IPCA features and frame resolution). Please notice that the greater the AUC is, the better it is; and the lower the EER is, the better it is.

Extractor	**Pooling**	**IPCA dims.**	**Ped1**				**Ped2**			
			192 × 168		**384 × 256**		**192 × 168**		**384 × 256**	
			AUC	**EER**	**AUC**	**EER**	**AUC**	**EER**	**AUC**	**EER**
1	avg	8	**61.39%**	**42.21%**	54.88%	48.06%	79.53%	28.29%	88.17%	20.14%
		16	56.33%	44.08%	53.96%	47.03%	74.80%	31.80%	90.03%	17.18%
		32	56.83%	45.41%	55.03%	45.95%	70.20%	33.77%	88.96%	19.47%
		64	54.21%	47.58%	54.80%	45.28%	82.96%	25.57%	92.65%	15.41%
		128	49.96%	51.14%	49.70%	52.03%	84.11%	21.48%	94.63%	11.96%
	max	8	46.66%	52.88%	46.61%	51.99%	54.07%	42.95%	49.38%	50.43%
		16	43.04%	54.73%	47.61%	51.91%	51.27%	45.62%	50.48%	48.95%
		32	44.76%	54.23%	48.43%	52.22%	53.24%	46.52%	52.70%	49.51%
		64	44.41%	53.69%	48.43%	51.02%	59.25%	42.90%	57.66%	45.90%
		128	45.32%	52.34%	48.95%	52.60%	56.41%	43.82%	57.69%	44.35%
2	avg	8	57.94%	44.46%	**58.71%**	44.44%	68.58%	37.05%	77.18%	30.48%
		16	57.95%	43.54%	53.58%	47.56%	68.10%	36.07%	83.45%	25.53%
		32	59.44%	43.52%	55.00%	45.77%	70.52%	35.96%	90.77%	15.99%
		64	58.65%	43.68%	56.60%	46.33%	**89.26%**	20.22%	94.61%	11.80%
		128	54.33%	46.05%	57.02%	**44.42%**	88.44%	**19.40%**	**96.12%**	**11.48%**
	max	8	49.46%	50.47%	49.95%	49.41%	57.57%	45.08%	67.03%	37.40%
		16	48.44%	51.75%	49.72%	49.84%	53.27%	45.87%	79.46%	26.70%
		32	48.24%	50.87%	49.94%	49.28%	58.03%	41.84%	82.26%	24.52%
		64	48.10%	50.47%	49.86%	51.06%	77.71%	29.74%	85.66%	22.87%
		128	48.69%	50.51%	51.45%	48.91%	84.70%	23.28%	88.83%	19.07%
3	avg	8	53.35%	47.10%	50.45%	49.34%	61.18%	44.79%	75.97%	30.99%
		16	54.10%	46.34%	48.45%	51.16%	65.81%	40.80%	77.57%	30.93%
		32	52.09%	48.41%	50.37%	50.06%	80.25%	26.04%	85.00%	22.61%
		64	55.57%	46.89%	51.65%	48.90%	83.10%	23.27%	87.85%	18.36%
		128	55.68%	46.40%	51.94%	48.84%	85.04%	22.62%	88.22%	16.72%
	max	8	49.85%	49.47%	46.33%	52.78%	61.74%	43.23%	74.24%	29.02%
		16	49.16%	50.01%	40.64%	56.32%	61.29%	46.56%	72.60%	33.11%
		32	45.92%	53.10%	44.81%	53.72%	70.57%	32.56%	79.89%	25.05%
		64	49.83%	51.08%	48.12%	51.28%	76.87%	30.47%	86.95%	18.76%
		128	51.11%	48.90%	48.52%	50.36%	82.35%	28.20%	87.25%	17.18%
4	-	8	48.98%	50.78%	50.49%	50.25%	76.47%	30.07%	79.17%	24.32%
		16	49.85%	49.87%	46.04%	53.53%	77.41%	27.54%	84.98%	18.84%
		32	48.02%	50.95%	48.86%	51.92%	81.76%	23.20%	85.69%	17.43%
		64	48.53%	51.03%	46.55%	53.61%	72.93%	30.11%	87.93%	16.07%
		128	48.38%	51.40%	48.48%	51.66%	70.67%	32.39%	87.02%	16.11%

Table 5: Frame-level anomaly detection results for the proposed auto-encoder.

Method	**Ped1**		**Pe2**	
	AUC	**EER**	**AUC**	**EER**
3D-auto-encoder (with bg subtraction)	64.8%	39.7%	91.0%	15.0%
3D-auto-encoder (without bg subtraction)	67.7%	38.7%	90.0%	19.0%

the best methods on Ped1 end up obtaining worse performance on Ped2 such as for (Xu et al., 2015; Ravanbakhsh et al., 2017; Ravanbakhsh et al., 2018).

We believe that the main reason for this low performance are due to changes in space perspective (object sizes change according to the image region that they are located), something that our approach does not take into account.

Table 6: Number of parameters of CNN in our experiments.

Model	**# of trainable parameters**
AE-3D	9,364
Extractor 1	226,560
Extractor 2	2,881,280
Extractor 3	13,499,136
Extractor 4	27,655,936

5 CONCLUSION

The usage of 3D convolutions as a way to learn or obtain representations to detect anomalies in security videos is still a matter of investigation. A pre-trained C3D model, especially when exploring different lay-

Table 7: Frame-level anomaly detection comparison on both UCSD datasets (Ped1 and Ped2). The lower the EER is, the better it is; and greater the AUC is, the better it is.

Method	**Dataset**			
	Ped1		**Ped2**	
	AUC	**EER**	**AUC**	**EER**
LMH (Adam et al., 2008)	63.4%	38.9%	58.1%	45.8%
MPPCA (Kim and Grauman, 2009)	59.0%	40.0%	69.3%	30.0%
Social force (Mehran et al., 2009)	67.5%	31.0%	55.6%	42.0%
Sparse reconstruction (Cong et al., 2011)	-	19.0%	-	-
LSA (Saligrama and Chen, 2012)	92.7%	16.0%	-	-
Sparse combination (Lu et al., 2013)	91.8%	15.0%	-	-
MDT (Li et al., 2014)	81.8%	25.0%	82.9%	25.0%
LNND (Hu et al., 2014)	-	27.9%	-	23.7%
Motion influence map (Lee et al., 2015)	-	24.1%	-	**9.8%**
Composition pattern (Li et al., 2015)	-	21.0%	-	20.0%
HOFM (Colque et al., 2015)	71.5%	33.3%	89.9%	19.0%
AMDN (Xu et al., 2015)	92.1%	16.0%	90.8%	17.0%
Flow decomposition (Ponti et al., 2017a)	-	-	-	31.7%
Adversarial discriminator (Ravanbakhsh et al., 2017)	**96.8%**	**7.0%**	95.5%	11.0%
Plug-and-play CNN (Ravanbakhsh et al., 2018)	95.7%	8.0%	88.4%	18.0%
CNN-2D features (Nazare et al., 2018b)	64.1%	40.4%	88.9%	19.6%
Siamese (Ramachandra et al., 2020)	86.0%	23.3%	94.0%	14.1%
Memory auto-encoder (Park et al., 2020)	-	-	**97.0%**	-
C3D (best results)	61.4%	42.2%	96.1%	11.5%
AE-3D (best results)	67.7%	38.7%	91.0%	15.0%

ers, shows discriminative capacities for scenes with low variation in speed and scale along the frames. A 3D auto-encoder, on the other hand, is also capable of learning spatial-temporal features from normal data and produce remarkable results when considering the low number of parameters of such model.

Regarding the C3D features, we considered four different layers as feature extractors, and concluded that features from early layers (layers that are closer to the network input) are better for such task, specially when combined with average pooling layers. Such results where matched by a 3D auto-encoder that has a significantly lower number of parameters.

Those conclusions are enriched when we compare the C3D and AE-3D with results from the literature. Well-behaved scenarios, such as Ped2 dataset, benefit from 3D convolutions and need less effort to be solved. However, the varying distance between camera and objects of Ped1 makes it less a generic problem in terms of positional-based feature learning via 3D convolutions. Therefore, we advocate that 3D-convolution-based methods are adequate for video anomaly detection, however one should combine other mechanisms to compensate for changes in scale and speed along the frame. With this, future work may improve the design of systems without trial-and-error and better coupling the power of deep 3D convolutional networks and other strategies.

Future studies may include C3D models previously trained on grayscale datasets since, according to (Nazare et al., 2018a), this can make a considerable difference in the overall results; use alternative CNN architectures designed for videos, such as the ones presented in (Srivastava et al., 2015; Tran et al., 2018); consider multi-resolution video processing, which has been successfully used by (Xu et al., 2015), to better deal with changes in space perspective.

ACKNOWLEDGMENT

This work was supported by FAPESP (grants #2015/04883-0, and #2018/22482-0), CNPq (grants #303886/2020-0 and #304266/2020-5) and partially supported by CEPID-CeMEAI (FAPESP grant #2013/07375-0).

REFERENCES

Adam, A., Rivlin, E., Shimshoni, I., and Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. Pattern Anal. Mach. Intell.*

Chollet, F. et al. (2015). Keras. https://keras.io.

Colque, R. M., Caetano, C., de Melo, V. H. C., Chavez, G. C., and Schwartz, W. R. (2018). Novel anomalous event detection based on human-object interactions. In *VISIGRAPP (5: VISAPP)*, pages 293–300.

Colque, R. V. H. M., Caetano, C., and Schwartz, W. R. (2015). Histograms of optical flow orientation and magnitude to detect anomalous events in videos. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*.

Cong, Y., Yuan, J., and Liu, J. (2011). Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*.

Dee, H. M. and Velastin, S. A. (2008). How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

dos Santos, F. P., Ribeiro, L. S., and Ponti, M. A. (2019). Generalization of feature embeddings transferred from different video anomaly detection domains. *Journal of Visual Communication and Image Representation*, 60:407–416.

Dos Santos, F. P., Zor, C., Kittler, J., and Ponti, M. A. (2020). Learning image features with fewer labels using a semi-supervised deep convolutional network. *Neural Networks*.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Haering, N., Venetianer, P. L., and Lipton, A. (2008). The evolution of video surveillance: an overview. *Machine Vision and Applications*.

Hu, X., Hu, S., Zhang, X., Zhang, H., and Luo, L. (2014). Anomaly detection based on local nearest neighbor distance descriptor in crowded scenes. *The Scientific World Journal*, 2014.

Kim, J. and Grauman, K. (2009). Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.

Kornblith, S., Shlens, J., and Le, Q. V. (2019). Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*.

Lee, D.-g., Member, S., Suk, H.-i., and Park, S.-k. (2015). Motion Influence Map for Unusual Human Activity Detection and Localization in Crowded Scenes. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8215.

Li, N., Wu, X., Xu, D., Guo, H., and Feng, W. (2015). Spatio-temporal context analysis within video volumes for anomalous-event detection and localization. *Neurocomputing*.

Li, W., Mahadevan, V., and Vasconcelos, N. (2014). Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lu, C., Shi, J., and Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13.

Mabrouk, A. B. and Zagrouba, E. (2018). Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*, 91:480–491.

Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly detection in crowded scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1975–1981.

Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *CVPR*.

Muja, M. and Lowe, D. G. FLANN: Fast Library for Approximate Nearest Neighbors.

Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.

Muthusenthil, B. and Kim, H. S. (2018). CCTV Surveillance System, attacks and design goals". *International Journal of Electrical and Computer Engineering*, 8(4):2072.

Nazare, T. S., de Mello, R. F., da Costa, G. B. P., and Ponti, M. A. (2018a). Color quantization in transfer learning and noisy scenarios: An empirical analysis using convolutional networks. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*.

Nazare, T. S., de Mello, R. F., and Ponti, M. A. (2018b). Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos? *arXiv preprint arXiv:1811.08495*.

Park, H., Noh, J., and Ham, B. (2020). Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ponti, M., Nazare, T. S., and Kittler, J. (2017a). Optical-flow features empirical mode decomposition for motion anomaly detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Ponti, M. A., Ribeiro, L. S. F., Nazare, T. S., Bui, T., and Collomosse, J. (2017b). Everything you wanted to know about deep learning for computer vision but were afraid to ask. In *2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*.

Ramachandra, B., Jones, M., and Vatsavai, R. (2020). Learning a distance function with a siamese network to localize anomalies in videos. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2598–2607.

Ravanbakhsh, M., Nabi, M., Mousavi, H., Sangineto, E., and Sebe, N. (2018). Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Ravanbakhsh, M., Sangineto, E., Nabi, M., and Sebe, N. (2017). Training adversarial discriminators for cross-channel abnormal event detection in crowds. *CoRR*, abs/1706.07680.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 512–519, Washington, DC, USA. IEEE Computer Society.

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *Int. J. Comput. Vision*, 77(1-3):125–141.

Saligrama, V. and Chen, Z. (2012). Video anomaly detection based on local statistical aggregates. In *CVPR*. IEEE Computer Society.

Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR.

Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. *CoRR*, abs/1801.04264.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.