Computational Intelligence in Electrical Engineering
Vol. 14, No. 4, 2024
pp. 17-26
Research Paper

A study of the effect of deep cascade network on anomaly detection using optical flow as a high-level additional signal

Maedeh Bahrami[1], Majid Pourahmadi[*2], Abbas Vafaei[3], Mohammad Reza Shayesteh[4]

[1] Department of electrical engineering, Yazd Branch, Islamic Azad University, Yazd, Iran
maedehbahrami1983@gmail.com
[2] Department of electrical engineering, Yazd Branch, Islamic Azad University, Yazd, Iran
pourahmadi@iauyazd.ac.ir
[3] Faculty of Computer Engineering, Isfahan University, Isfahan, Iran
abbas_vafaei@eng.ui.ac.ir
[4] Department of electrical engineering, Yazd Branch, Islamic Azad University, Yazd, Iran
shayesteh@iauyazd.ac.ir

Abstract :
Video anomaly detection by reconstruction is a challenging task. One of its challenges is related to the volume of input data frames needed to be processed to detect anomalies. The challenge usually manifests itself as increased training and especially testing time. The proposed architecture boosts performance while maintaining the same test time as our previously introduced AnoDetNet architecture. The proposed architecture is a cascaded framework that is a succession of reconstruction and an auxiliary network. Upon training, the auxiliary network acts as guidance through the use of combined loss. The combined training of the networks results in a performance increase compared with the reconstruction case alone. Considering that the auxiliary network's results are not used in the test phase, the overall anomaly detection test time does not change compared with the non-cascaded architecture. Two possible auxiliary networks, namely edge detection and optical flow estimation are studied. The proposed architecture results in state-of-the-art results on the Ped2 and Avenue datasets.

Keywords: Video anomaly detection, Deep learning, Cascade network, Neural network, Autoencoder, optical flow.

1. Introduction

Video anomaly detection is the detection of non-normal behaviors in videos. The definition of non-normality in our case is tightly coupled with the definition of normal behaviors in videos which are mainly contextual. For example, in machine learning, datasets are used to train algorithms; and each dataset has a different definition of normality. One may consider walking in every direction normal, while another might consider it an anomaly. Therefore, video anomaly detection can be better described as a model that learns the normal behavior, and considers all non-normal behavior as an anomaly [1]. On the other hand, the broad definition of normality in videos makes video anomaly detection a challenging task.

With the advent of deep learning, many researchers apply it to image, video, and sound-based applications. Currently, many image and video applications such as human pose estimation

Submission date:14, 09, 2022
Acceptance date: 04, 04, 2023
Corresponding author: Majid Pourahmadi , Islamic Azad University, Yazd branch - Iran

https://doi.org/10.22108/ISEE.2023.135667.1596

[2], action recognition [3], and image description [4] among others rely on and sometimes depend entirely on deep neural networks. At the same time, some of these applications have reached better-than-human performances [5]. Historically, in general machine learning, the detection pipeline of many applications consisted of four parts, namely preprocessing, feature extraction, classification/regression, and postprocessing. An example would be Amraee et al. [6] which use HOG[1] and optical flow at the feature extraction stage and a simple rule-based classifier to detect anomalies in image patches. In recent years, deep neural networks have aggregated the feature extraction and detection stages. Essentially, the extraction stage has become a learnable process and has benefited the most in deep neural networks [7].

Similar to other applications which have benefited from strong feature representation of deep networks, video anomaly detection has also achieved increased performances with deep networks. Deep networks are used in two well-known approaches to detect anomalies.

The first approach uses deep networks to directly detect anomalies. In this approach, the network's input is a set of raw frames or feature descriptors and the output is a binary mask (or a multilevel mask where each pixel is indexed with an anomaly probability) that predicts the presence and location of anomalies. For example, Sabokrou et al. [8] use a cascade of two convolutional classifiers to detect anomalies directly. In another example, Gunale and Mukherji [9] process spatiotemporal features extracted by HOME FAST[2] by a deep network to achieve the masks.

The second approach is anomaly detection by reconstruction. In this approach, a frame is reconstructed based on a sequence of input frames (sequence count can be as low as one). In the next step, anomalies are detected from the reconstructed frame. This approach uses an implicit assumption, supported by the fact that the training is only done with non-anomalies. It assumes that in the output frame, anomalies are poorly reconstructed. Therefore, there is a contrast between the successfully reconstructed normal pixels and partially reconstructed (or destructed) pixels. This approach has been used extensively in the deep video anomaly detection field such as [10-12].

Regardless of the approach, additional input signals can be used to guide the process of training the network. The nature of these signals is either high-level signals such as optical flow [13] or low-level signals such as object edges [10], [11]. More than anything else, the nature of these signals, shows the general perspective of the researcher and the top-down or bottom-up approach to infusing information. Especially in anomaly detection by reconstruction, the low-level signals such as edges, aim to improve the reconstruction in the periphery of objects. In other words, the edges correlate to object boundaries and the additional signal would help to instill this information into the network. On the other hand, high-level signals, such as semantic segmentation and optical flow, directly infuse high-level signals as most of these, correlate with object structures in images. For example, [13] has employed a cascade architecture to boost detection performance. The cascaded architecture is entirely trained. In addition to the reconstruction error, the optical flow error is also used to detect anomalies. Another example would be [14] that have used optical flow estimation task to boost their results. While their overall architecture is different, the technique to boost performance is the same as the current paper. The current paper studies the effect of general overhead networks more deeply.

From another perspective, these additional signals are either used as input or as output. From an empirical viewpoint, these two categories both infuse additional knowledge into the network. Additional inputs must be present at both stages of training and testing and are fairly self-explanatory [11].

Additional outputs guide the gradient descent navigation and change the loss function. These additional outputs are the results of another network or sub-network. These can be considered a subtle approach to using additional signals [13]. Furthermore, if the additional outputs are not used in prediction, they can be omitted from the test stage. Therefore, they do not add to the processing power needed to run a neural network graph.

In this paper, a cascaded architecture is introduced for anomaly detection. This architecture is a linear combination of AnoDetNet [15] and an auxiliary network. AnoDetNet is an anomaly detection pipeline that detects anomalies with a rate of 2.8 frames per second [15]. Optical flow and edge detection are chosen as the high-level signals. This network cascade is trained on a data set. At the test stage, only the part of the network graph that corresponds to AnoDetNet is

applied and the overhead task is not used at all. Therefore, at the test phase, the processing power needed to detect sample anomalies is the same as AnoDetNet and the anomaly detection rate remains the same. In other words, the overhead network only contributes to the loss of signal. To the best of our knowledge, this is the first time that pre-trained auxiliary networks are used only to guide training in the video anomaly detection application. There are three contributions to this paper:

- A study on the effectiveness of auxiliary networks. To this end, two auxiliary networks are compared.
- A study on the effectiveness of multi-head networks and their comparison to auxiliary networks.
- The balance between the main and auxiliary network by the loss function is studied.

The rest of the paper is structured as follows. Section 2 presents the proposed approach, section 3 introduces the results and finally, section 4 concludes the paper.

2. Proposed Approach

The general prediction pipeline can be seen in Fig. 1. The predictor is composed of two main parts, namely the deep reconstruction network and the regularity thresholder. The network is tasked with reconstructing a frame, based on a sequence of frames, whereas the regularity thresholder computes the regularity score and then thresholds the score to get a binary mask (which can then be compared with ground truth).

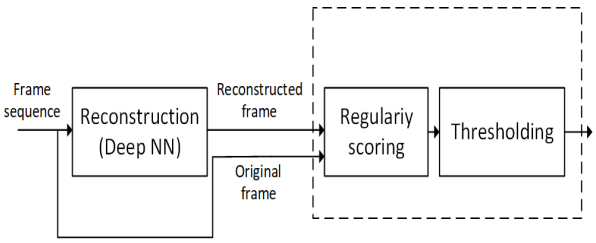

Fig. 1. Overview of the proposed approach

The reconstruction network block in Fig. 1, is implemented by three methods. Further, in the results section, these three methods are compared. The first method uses a multi-task decoder, which replaces the original AnoDetNet decoder and estimates optical flow, and does frame reconstruction simultaneously. The combination is trained end to end. The second method applies a separate pre-trained optical flow estimation network on the output of AnoDetNet. Similarly, the third method applies a separate pre-trained edge detection network on the output of AnoDetNet. The second and third methods are cascaded networks. These three methods are further discussed in the following sections.

2.1. Multi-task decoder

The AnoDetNet's [15] reconstruction network is a convolutional autoencoder that consists of a convolutional spatial encoder and a convolutional spatiotemporal decoder. Contrary to AnoDetNet, the proposed multi-task decoder is a multi-output decoder that generates an RGB reconstruction frame and its corresponding optical flow magnitude simultaneously. The proposed decoder replaces the original decoder of AnoDetNet. (“Fig 2 shows the proposed multi-task decoder. The autoencoder is trained end-to-end”)

2.2. Cascaded networks

The cascaded network is the combination of AnoDetNet and an auxiliary network. In this paper, two pre-trained deep neural networks are used as the auxiliary network. The first one is an optical flow (OF from this point onwards) estimator and the second one is an edge detection network.

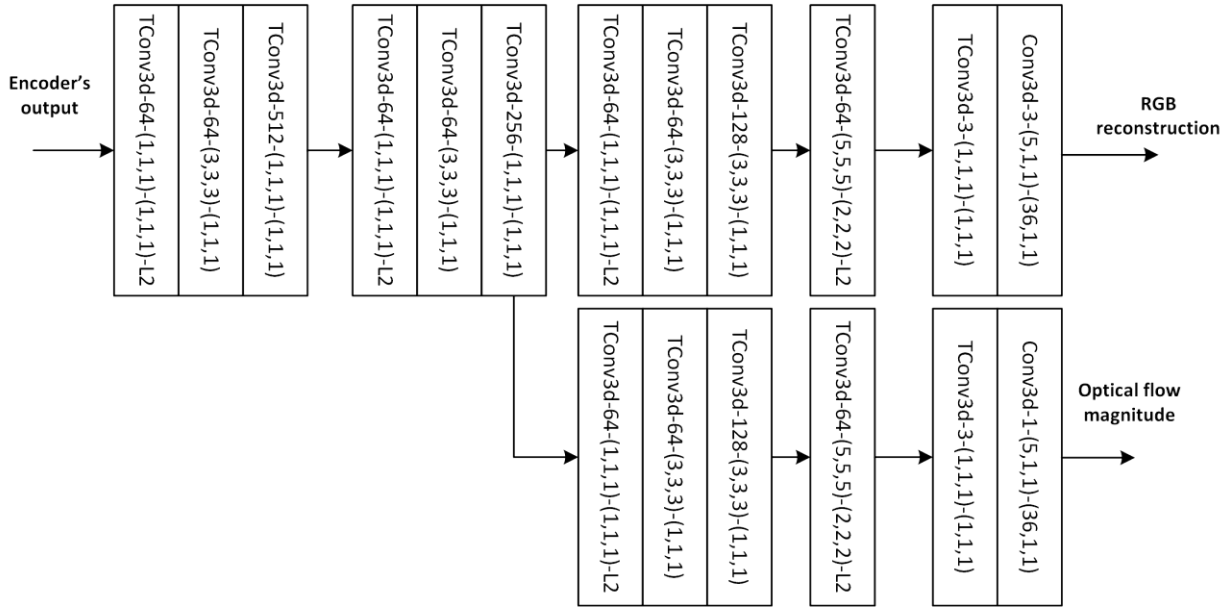

Fig. 2. Multi-task decoder which estimates optical flow magnitude in addition to reconstructing the RGB frame.

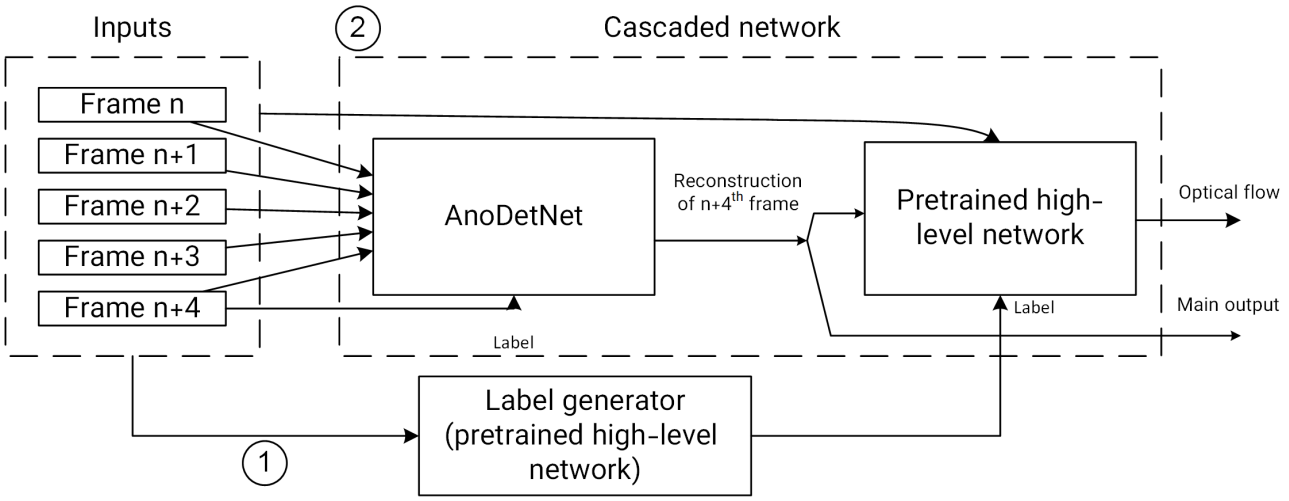

Fig. 3. The process of training the cascaded network. Numbers show the order of operations.

In the cascaded network, the effect of the auxiliary network is limited to the training stage. Fig. shows the process of training the cascaded network.

In the first step, the input frames are used by the pre-trained auxiliary network to generate the labels. For the case of optical flow, where the two frames are usually consecutive, the authors have chosen the first and fifth frames as the label generator's input. This is due to the fact that the movements between consecutive frames are too small to produce noticeable optical flows (this is depicted in Fig.). For the case of edge detection, the last input frame is chosen for label generation.

In the second step, the cascade network is trained with the loss function shown in (1). The loss function is the linear combination of each network's individual loss.

$$\begin{aligned} cascade\ loss &= \alpha(BCE_{AnoDetNet}) + (1-\alpha)(MSE_{Aux}) \\ &= \alpha(-\frac{1}{N}\sum_{i=1}^{N} Y_i \log(p(\hat{Y_i})) + (1-Y_i)\log(1-p(\hat{Y_i}))) \\ &+(1-\alpha)(\frac{1}{N}\sum_{i=1}^{N}(\hat{Y_i}-Y_i)^2) \end{aligned} \quad (1)$$

In (1), BCE is binary cross entropy and MSE is a mean squared error, $\hat{Y_i}$ is the prediction of ith pixel and Y_i is the label for ith pixel, N is the total number of pixels, and α controls the balance

between AnoDetNet and auxiliary networks effectiveness.

In the test stage of the cascaded network, the auxiliary network is not used. The test network is shown in Fig. . In practice, many graph platforms such as Tensorflow [16] only compute the part of the graph that its output is required and do not run the rest of the graph. This fact is used in this research to maintain the test duration. In fact, testing a single sample takes 35 ms for the Ped2 dataset on an NVIDIA 1080 GPU which is the same as the original AnoDetNet implementation.

The choice of optical flow and edge detection as the auxiliary network is rooted in practicality and the ability to estimate and build labels for them. For example, by using semantic segmentation as the high-level signal generator, one has to manually prepare the generator's labels whereas many anomaly detection datasets do not contain the necessary input data for label generation. In contrast, when using optical flow estimation as the auxiliary signal generator, a pre-trained OF estimator and two input frames can be used to generate the labels. For the choice of the overhead signal generator, PWCNet [17] (optical flow) and DexiNed [18] (edge detection) have been chosen.

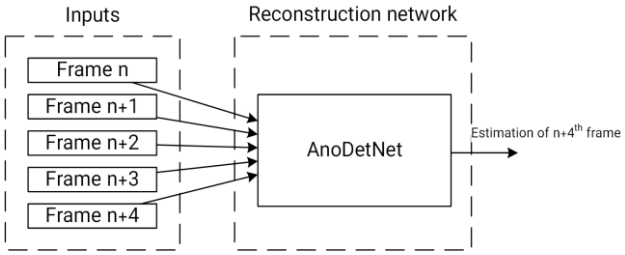

Fig. 4. The auxiliary network is omitted in the test phase.

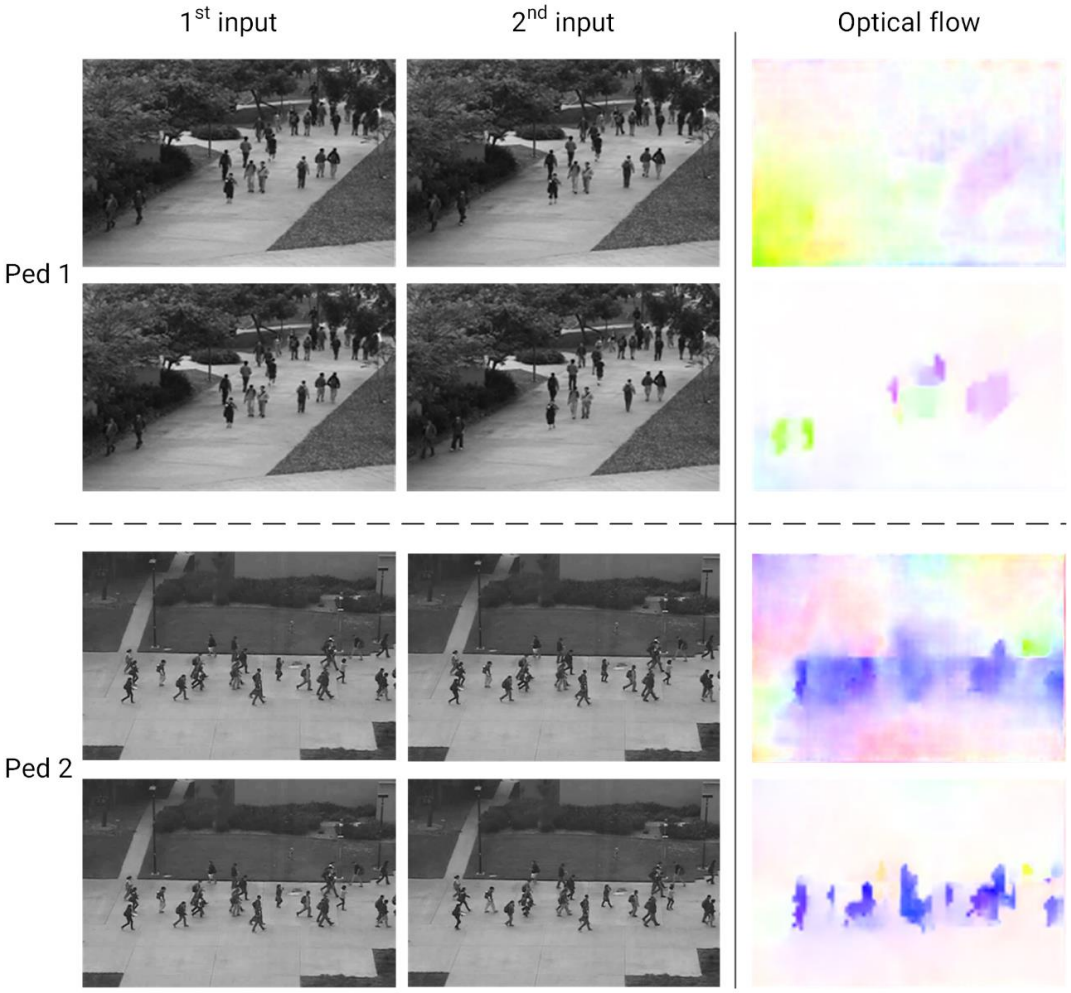

Fig. 5. The output of PWCNet with pre-trained weights on UCSD data set samples. Row 1 and 3 samples are from consecutive frames whereas row 2 and 4's inputs are 3 frames apart.

3. Results

In this section, first, the metrics and the dataset are introduced and then the results will be presented and discussed.

The metrics in this section will either be presented as frame-level metrics or pixel-level metrics. The frame-level metrics are calculated on the $SSIM^3$ matrix. The pixel-level metrics require an extra step which is calculating the SSIM score (each pixel in the SSIM matrix is considered a score). Afterward, a threshold is used to get the final binary mask. This mask shows the localization of anomalies and therefore all pixel-level metrics are based on the anomaly localization mask. The scoring and thresholding regime is the same as [15].

Two metrics are prevalently used in the literature, namely receiver operating characteristic's area under the curve which from here onwards will be called AUC, and equal error rate, EER.

Many classification tasks output a score as the final result. Therefore, there is a thresholding step that is inherent to the process which builds the final classification result. The score can usually be perceived as the probability or the confidence of the algorithm in its output. The threshold can be as simple as 0.5 (in the case of [0,1] output) or an optimized threshold to gain the best balance between the true positive rate and false negative rate (shown in (2) and (3)). The receiver operating characteristics is a diagram that plots the true positive rate, TPR against the false positive rate, FPR, and its area under the curve. AUC is naturally between 0 to 1. What makes AUC unique is the fact that to draw the ROC, all possible thresholds must be used and therefore, it is independent of the used threshold. This makes it a perfect candidate metric.

$$\text{TPR}=\frac{\text{Number of true positive frames}}{\text{Number of positive frames}} \quad (2)$$

$$\text{FPR}=\frac{\text{Number of false positive frames}}{\text{Number of negative frames}} \quad (3)$$

The equal error rate is the value that would make the false positive rate, FPR, and false negative rate, FNR equal.

There are several existing video anomaly detection data sets normally used in the anomaly detection field. In this research, three of the most prevalent ones are applied, namely the UCSD pedestrian1, pedestrain2 [19], and CUHK avenue [20]. The Ped1 dataset frames have an image resolution of 238*158 pixels.

This dataset contains 34 training video clips and 36 test video clips. Among the test video clips, there are 10 video clips containing anomalies localized and the rest does not have any anomaly localizations. The Ped2 data set frames have an image resolution of 360*240. In all of the following experiments, the images are resized to 256*256. Before the images are resized, the top 100 rows are cropped and left out of the image. This region does not change significantly between frames and does not contain any object of interest, therefore is of no benefit to anomaly detection. To the best of our knowledge, [21] has been the first to crop the Ped2 frames in this manner. The Avenue data set frames have an image resolution of 640*360 pixels. In experiments with this data set, width, and height have been reduced to 1/4 of their original values to make the training time feasible.

The first experiment is the sensitivity test of loss function weight α. α is the weight of AnoDetNet's loss function and (1-α) is the weight of the auxiliary network. The best parameters' values are found using a thorough search. Table.) shows the sensitivity test. The test is performed on UCSD Ped1, Ped2, and Avenue data sets and the parameter value of best performance is carried forward to the next experiments.

Table. 1. Performance comparison using different loss weights.

		$\alpha = 0.9$	$\alpha = 0.8$	$\alpha = 0.7$	$\alpha = 0.6$
Ped1	AUC	86.3%	88.4%	85.5%	80.2%
	EER	16.6%	14.7%	15.9%	18.4%
Ped2	AUC	97.6%	98.1%	96.9%	92.4%
	EER	8.1%	7.2%	9.8%	12%
Avenue	AUC	89.9%	91.4%	91.5%	89.5%
	EER	16.0%	13.5%	13.3%	16.3%

From Table.) it is evident that $\alpha = 0.8$ yields the best results for the UCSD dataset and $\alpha = 0.7$ yields the best results for the Avenue dataset. In other words, to make the cascaded network's loss effective at guiding the original network, the auxiliary network's losing weight must be much

lower than the original network's weight.

In the second experiment, the proposed architectures are tested on UCSD Ped1, Ped2, and Avenue data sets. Table.) shows the results.

Table.) shows that apart from the frame-level Ped1 results, the proposed approach demonstrates state-of-the-art results on the remaining data sets. It is worth mentioning that, the algorithm's test time has not changed; therefore, the algorithm can process 2.8 frames per second on Ped1 and Ped2 datasets and 11.2 frames on the Avenue dataset. Furthermore, in comparison to the AnoDetNet which is the simple (non-cascaded) version of the framework, there is a 0.6% and 4.3% performance increase on Ped2 and Avenue datasets respectively.

Table (2) also compares the cascaded network and multi-task decoder. It is worth emphasizing the similarities and dissimilarities between these two approaches. Both of these approaches estimate optical flow in addition to RGB image reconstruction and their output is redundant at the test phase. Theoretically, the two approaches would both help the proposed method through the guidance of an additional output's loss. On the other hand, the method of optical flow estimation is different in the two approaches. The multi-task decoder is trained whereas the PWCNet is pre-trained and their network architectures are different. At the same time, the two approaches generate the same conceptual outputs, namely RGB reconstruction and OF estimation. The results of Table (2) demonstrate the ineffectiveness of the multi-task decoder approach.

This is mainly due to the training of the second network. In other words, in the multi-task decoder, the second network competes with the original reconstruction one and the loss through that network creates a distraction for the overall approach. The cascaded network uses a pre-trained PWCNet, therefore the propagated loss is accurate from the start and the problem does not manifest itself.

The comparison of various types of auxiliary networks in Table (2) yields an interesting result. By comparing the edge detection and optical flow estimation networks, the OF estimation seems to be the clear winner. This effect is even more pronounced on the Avenue dataset. This can be attributed to the high-level nature of OF estimation, whereas edge detection is a basic task in computer vision. Another effective factor might be that the input of the edge detection network (output of AnoDetNet) lacks sharp edges. Therefore, there is a wide gap between the label and the output in the training stage, and crossing the gap would guide the overall network away from the primary goal which is the RGB reconstruction. Yet another possible factor is the exact nature of anomalies in the dataset and their potential to generally benefit from more motion-based training signals. For example, the Avenue dataset contains anomalies in which the subjects move in the wrong direction. Therefore, considering the nature of the Avenue dataset, the auxiliary network signal helps the dataset more than the Ped2 dataset.

Fig. and Fig show the visual detection results on a sample of Ped1, Ped2, and Avenue datasets

Table. 2. Performance comparison of proposed architecture on UCSD Ped1, Ped2, and CUHK Avenue datasets along with results of other studies. N/A values were not provided by their respective author/authors. MTD is a multi-task decoder whereas "CasOF" is a cascaded network using optical flow and "CasED" is the cascaded network using edge detection.

	Ped1				Ped2				Avenue	
	Frame level		Pixel level		Frame level		Pixel level		Frame level	
	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER
ConvLSTM-AE [22]	75.5	N/A	N/A	N/A	88.1	N/A	N/A	N/A	77.0	N/A
Future frame prediction [23]	83.1	N/A	N/A	N/A	95.4	N/A	N/A	N/A	85.1	N/A
Optical flow GAN [24]	97.4	**8**	70.3	35	93.5	14	N/A	N/A	N/A	N/A
Appearance-Motion Correspondence [10]	N/A	N/A	N/A	N/A	96.2	N/A	N/A	N/A	86.9	N/A
Cascade reconstruction [13]	82.6	N/A	N/A	N/A	97.7	N/A	N/A	N/A	88.9	N/A
Robust multipath frame prediction [25]	83.4	N/A	N/A	N/A	96.3	N/A	N/A	N/A	88.3	N/A
Integrated prediction and reconstruction [14]	84.7	N/A	N/A	N/A	96.3	N/A	N/A	N/A	85.1	N/A
AnoDetNet [15]	N/A	N/A	N/A	N/A	97.5	8.1	94.7	12.7	87.2	19.9
Our method (MTD)	67.1	28.0	68.5	27.7	74.6	21	70.1	23.7	66.9	27.4
Our method (CasED)	86.2	17.3	82.9	20.3	96.2	8.8	92.8	13.0	89.6	16.1
Our method (CasOF)	88.4	14.7	86.1	18	98.1	7.2	95.0	11.1	91.5	13.3

4. Conclusion

In this paper, a technique for cascading multiple networks for anomaly detection was introduced. The proposed approach combines an auxiliary network with the main network and uses the backpropagated signal to train the main neural network. Therefore, this backpropagated signal acts as an additional training guidance used by the main network.

As it was shown, the best results are obtained when the main network has a much bigger loss of weight compared to the auxiliary network (in this application a ratio of 4 to 1). Furthermore, compared with the main network alone, the cascaded architecture shows a 0.6% and 4.2% performance increase on Ped2 and Avenue datasets respectively. In addition, the results need no additional processing requirement; therefore, the testing time of a single sample is the same as the main network.

Two auxiliary networks are studied in this paper, namely edge detection and optical flow estimation. The optical flow estimation performed better than edge detection. This is mainly due to the inherent nature of the signal and its suitability for anomaly detection. Furthermore, the same network outputs (RGB reconstruction and auxiliary result) were achieved through the use of a multi-head network. The results and their comparison to the cascade architecture show that the multi-head approach performs poorly.

Various studies can be followed up in the future. For example, whether the amount of performance increase of GAN frameworks compared with a normal deep network when the applied autoencoder is the same remains an open question. Furthermore, the architecture of the reconstruction network has always been under various modifications and enhancements and it may have chances to achieve better performance to be researched in future work

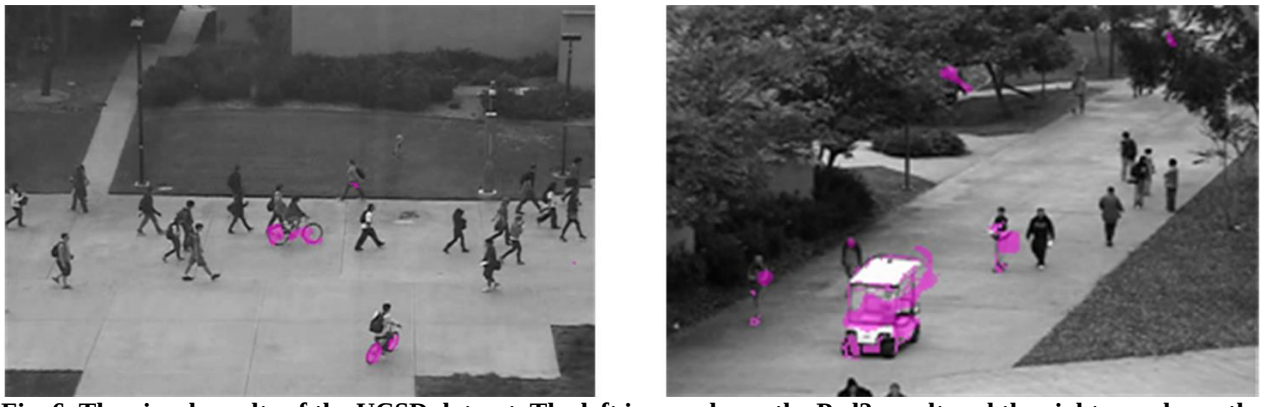

Fig. 6. The visual results of the UCSD dataset. The left image shows the Ped2 result and the right one shows the result of Ped1. The pink pixels are the binary anomaly mask that is overlaid on the original frame.

Fig7. The visual result of the CUHK Avenue dataset. Top left: original frame, Top right: reconstructed frame, Bottom left: The label mask, Bottom right: predicted binary mask.

References

[1] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," pp. 1–50, 2019.

[2] M. Ben Gamra and M. A. Akhloufi, "A review of deep learning techniques for 2D and 3D human pose estimation," *Image and Vision Computing*, Vol. 114, p. 104282, 2021.

[3] H. B. Zhang *et al.*, "A comprehensive survey of vision-based human action recognition methods," *Sensors (Switzerland)*, Vol. 19, No. 5, pp. 1–20, 2019, doi: 10.3390/s19051005.

[4] M. D. Zakir Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys*, Vol. 51, No. 6, pp. 1–36, 2019, doi: 10.1145/3295748.

[5] T. Orosz *et al.*, "Evaluating human versus machine learning performance in a legal tech problem," *Applied Sciences*, Vol. 12, No. 1, 2022, doi: 10.3390/app12010297.

[6] S. Amraee, A. Vafaei, K. Jamshidi, and P. Adibi, "Anomaly detection and localization in crowded scenes using connected component analysis," *Multimedia Tools and Applications*, Vol. 77, No. 12, pp. 14767–14782, 2018, doi: 10.1007/s11042-017-5061-7.

[7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.

[8] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Transactions on Image Processing*, Vol. 26, No. 4, pp. 1992–2004, 2017, doi: 10.1109/TIP.2017.2670780.

[9] K. G. Gunale and P. Mukherji, "Deep learning with a spatiotemporal descriptor of appearance and motion estimation for video anomaly detection," *Journal of Imaging*, Vol. 4, No. 6, 2018, doi: 10.3390/jimaging4060079.

[10] T. N. Nguyen and J. Meunier, "Anomaly detection in video sequence with appearance-motion correspondence," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1273–1283. doi: 10.1109/ICCV.2019.00136.

[11] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, Vol. 105, pp. 13–22, 2018, doi: 10.1016/j.patrec.2017.07.016.

[12] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 1896–1904. doi: 10.1109/WACV.2019.00206.

[13] Y. Zhong, X. Chen, J. Jiang, and F. Ren, "A cascade reconstruction model with generalization ability evaluation for anomaly detection in videos," *Pattern Recognition*, Vol. 122, p. 108336, 2022, doi: 10.1016/j.patcog.2021.108336.

[14] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognition Letters*, Vol. 129, pp. 123–130, 2020, doi: 10.1016/j.patrec.2019.11.024.

[15] M. Bahrami, M. Pourahmadi, A. Vafaei, and M. R. Shayesteh, "A comparative study between single and multi-frame anomaly detection and localization in recorded video streams," *Journal of Visual Communication and Image Representation*, Vol. 79, p. 103232, 2021.

[16] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *Network: Computation in Neural Systems*, Vol. 16, No. 2–3, pp. 121–138, 2005, doi: 10.1080/09548980500300507.

[17] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943. doi: 10.1109/CVPR.2018.00931.

[18] X. S. Poma, A. Sappa, P. Humanante, and A. Arbarinia, "Dense Extreme Inception Network for Edge Detection," in *IEEE Winter Conference on Applications of Computer*

Vision, 2020, pp. 1912–1921. doi: 10.1109/WACV45572.2020.9093290.

[19]V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition*, 2010, pp. 1975–1981. doi: 10.1109/CVPR.2010.5539872.

[20]C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2720–2727. doi: 10.1109/ICCV.2013.338.

[21]M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially Learned One-Class Classifier for Novelty Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388. doi: 10.1109/CVPR.2018.00356.

[22]W. Luo, W. Liu, and S. Gao, "Remembering history with convolutional LSTM for anomaly detection," in *Proceedings - IEEE International Conference on Multimedia and Expo*, 2017, pp. 439–444. doi: 10.1109/ICME.2017.8019325.

[23]W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection - A new baseline," in *Computer Vision and Pattern Recognition*, 2018, pp. 6536–6545.

[24]M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *International Conference on Image Processing*, 2017, pp. 1577–1581. doi: 10.1109/ICIP.2017.8296547.

[25]X. Wang *et al.*, "Robust Unsupervised Video Anomaly Detection by Multipath Frame Prediction," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 6, pp. 2301–2312, 2022, doi: 10.1109/TNNLS.2021.3083152.

[1] Histogram of oriented gradients

[2] Histogram of Orientation, Magnitude, and Entropy with Fast Accelerated Segment Test

[3] Structural similarity