A Two-Stream Approach to Anomaly Detection in Surveillance Videos

Anantapadmanaabha Prasannakumar
Department of Electrical and Computer Engineering
University of Central Florida
anantalp@knights.ucf.edu

Abstract

Surveillance cameras can be found everywhere in today's society, and their use is continuously growing. With that, the need for quality review of these videos has increased. However, manual review of every video is almost impossible and an automated solution has become more attractive. So, we're proposing a solution to this by finding anomalies in surveillance videos using deep learning. Our technique utilizes MIL (Multiple Instance Learning) with a two-stream network. One stream consists of RGB (Red Green Blue) videos and the other stream utilizes background reduction and optical flow. Our results show the use of a Two-Stream method has slightly improved anomaly detection as well as greatly reducing false flags when compared to existing methods.

1. Introduction:

The need to review security camera footage has become a necessity in order maintain the security of an establishment or public space. The existing solutions that are in place either utilize basic motion detection, or having a physical person watching a collection of screens. There's several problems with these approaches. For instance, with only motion capture, there's a large amount of false flags since in general motion by itself does not indicate that an anomaly is happening. Additionally, a person watching a screen is not only very expensive, but also they aren't guaranteed to actually see an anomaly.

The existing techniques for solving this problem in the deep learning space also have their problems. The main problem is that datasets for strongly labeled data are very small, and don't cover a large amount of anomalies. Additionally, some techniques do not account for the vast majority of security footage very rarely every having an anomaly.

Anomalies in of themselves are very difficult to define. They essentially are anything that's different from normal, which can fall into an endless number of categories. Therefore attempting to train on all individual anomalies individually by type is nearly impossible.

Our technique expands upon the work done in Sultani, Waqas, *et al* [1] which aimed to solve these problems. In particular, our work lowers the number of false flags from [1] by nearly 10%.

The dataset created by [1] utilizes weakly labeled data at the video level. So, a video will be labeled as an anomaly as long as an anomaly happens at any point in the video. It will also be labeled as normal if no anomaly happen at all in the video. This is largely beneficial since the dataset is very large in size compared to existing strongly-labeled data.

Our approach proposes to detect anomalies by using a two stream network. Our first stream makes use of the approach introduced by Sultani, Waqas, et al [1]. In [1] they tried to learn anomalies using a deep multiple instance ranking framework. This approach specifically uses Multiple Instance Learning (MIL) as introduced by Bergeron, Charles, et al. [8] MIL is a technique which has a bag of weakly labeled data where not all of them is the correct answer and over time elevates the correct data to the top, while lowering the incorrect data.

In addition to the RGB network, we introduced a parallel network that would be trained with flow video data. The videos in this network are converted using a Gaussian Mixture and Dynamic Flow method adapted from Fan, Yaxiang, et al. [10].

Once the videos are converted, we then make use of the multiple instance learning approach. The reason we decided to use such a network is that even though anomalies widely vary and are inherently complex, we believe a good majority of them involve a change in motion. The RGB stream would extract and train features from all of the data within a video. The flow stream would solely focus on motion capturing. Since even the large dataset introduced by [1] has a large amount of anomalies that occur, we argue that the data

from motion capture will still play a vital role until the day that an all-encompassing dataset is created.

2. Related Work:

Anomaly detection has become a very important problem to solve. So, a lot of research has gone into finding an applicable solution. Various attempts have been made to detect specific anomalies such as accidents, violence, and aggression. Also, various video surveillance applications were proposed to detect aggression [11, 12, 13, 14, 15]. In particular, Datta et al. [13] makes use of motion and limb orientation of people to detect aggression. Similar efforts to detect violence were made by Kooij et al [15], Gao et al [16], and Mohammadi et al [17].

Apart from detecting violence, efforts have been made by researchers to track the motion of an object and any deviation from the proposed path is considered as an anomaly [18, 19]. However, due to the difficulty in finding a reliable track to test the model, researchers have used global motion patterns to [20, 21] develop a model. These models tend to detect the least probable of all patterns as an anomaly. Researchers [22, 23] have also considered the sparse representation and dictionary learning approaches to solve many problems. A deep learning based autoencoders were employed by [24, 25] to learn the normal events and this was employed in the reconstruction of the anomaly events. Sultani, Waqas, et al proposed to consider both normal and anomaly events for their research, by making use of weakly labeled videos. This study forms the basis for our research.

During the initial stages of our study, we made observation of two stream networks. Ye, Hao, et al proposed a 2 network system for video classification using a CNN. His work was mainly based on [4]. They conducted extensive experiments on evaluating network structure and learning parameters. They also tested network fusion that add up results from multiple networks for the classification process. Further, Tran, An, and Loong-Fah Cheong [5]. They also did a similar work by considering a two-stream network architecture. They tried to feed RGB frames and optical flow streams as data to the CNN.

We explored Multiple Instance Learning (MIL) by Bergeron, Charles, et al [8]. This method tries to take in all the instances coming for a label into a bag. It would take the highest score in each of these bag to train the model, by back propagating the loss function by making use of highest score in each bag.

Waqas, et al [1] proposed an anomaly detection method that would differentiate anomaly video from normal video by making use of Multiple Instance learning [8]. They proposed bags for normal and anomalous videos, and that would be further used in

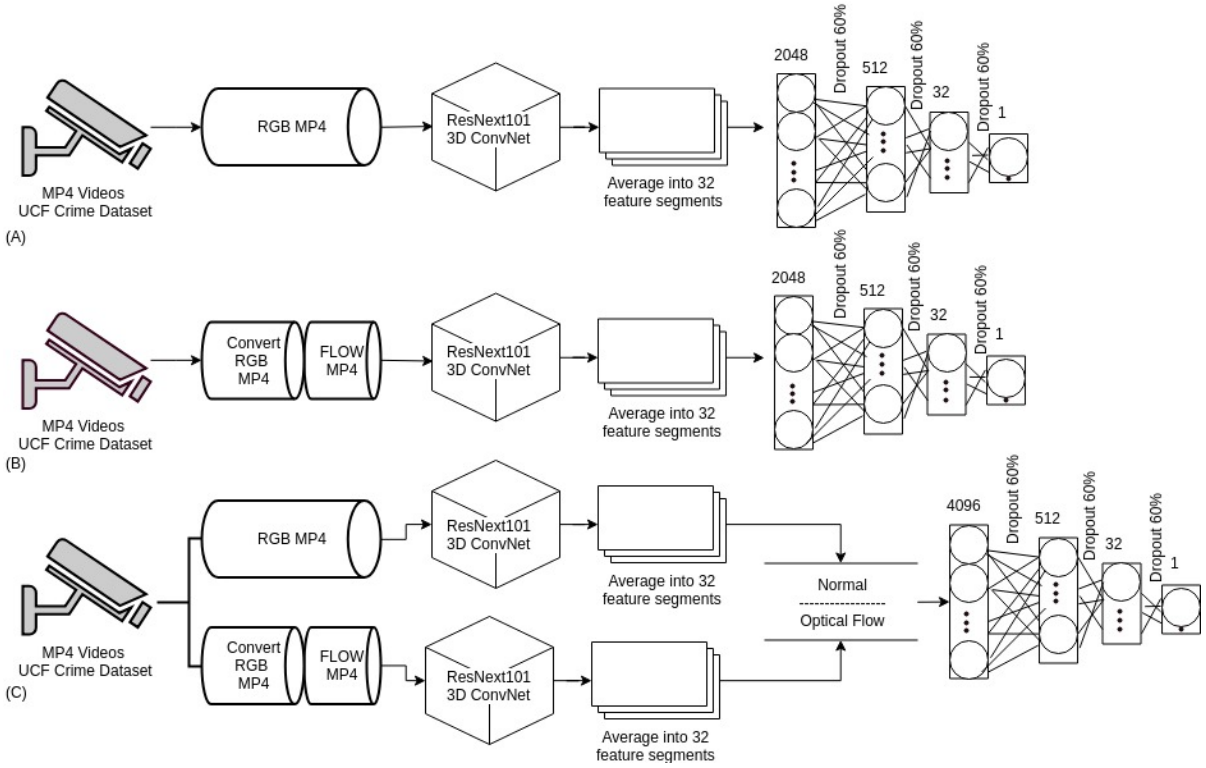


Figure 1: (A) RGB Network (B) Flow Network (C) Combined Network

calculating the loss function during training as discussed above. Further, we observed the method proposed by Fan, Yaxiang, et al.[10] , wherein the videos would be converted using Gaussian Mixture and Dynamic Flow.

We propose a method making use of observed two-stream network architecture and the networks

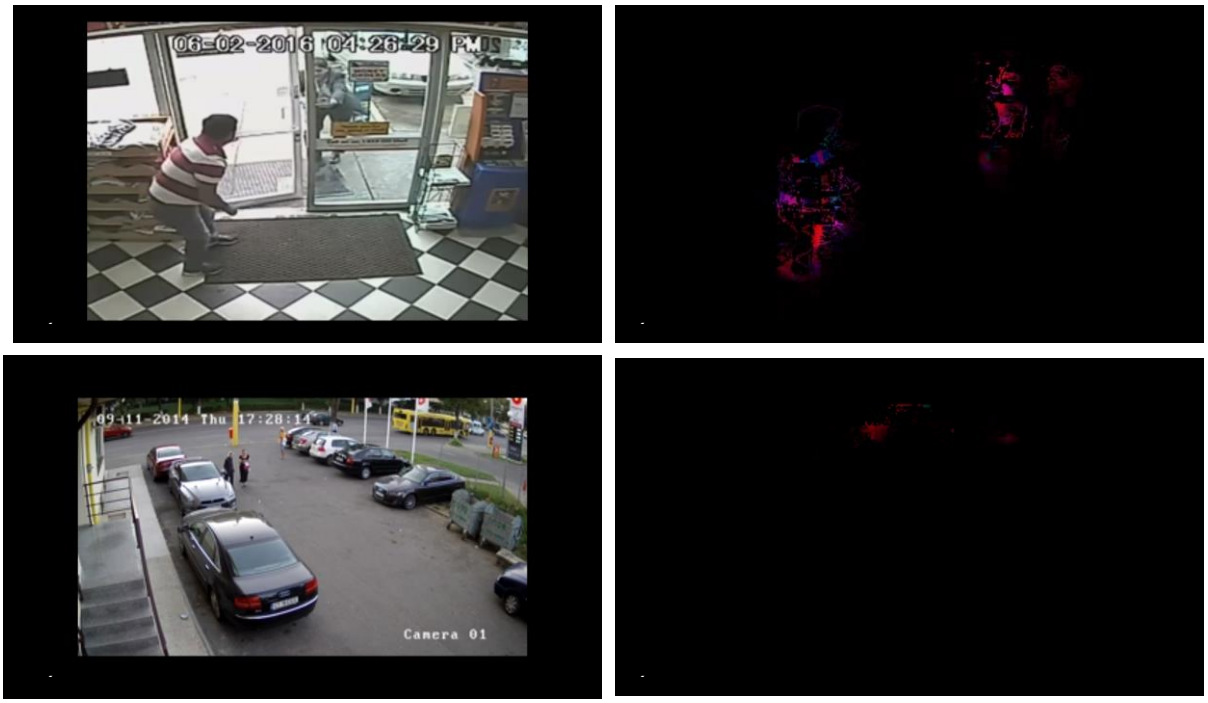


Figure 2: (A) Frame from an RGB anomaly video, (B) Flow converted anomaly frame.

proposed by Waqas, et al. and Fan, Yaxiang, et al. to create an anomaly detection system. We make use of original network by [1] and optical flow network proposed by [10] and this is passed to a convolutional neural network. We then segment the extracted features into 32 segments and average out the features for each segment. We then pass it to a fully connected layer, and then make use of multiple instance learning to calculate the loss function.

3. Approach Method:

The RGB network is very similar to the approach as described in [1]. We use this network [Figure 1 A] as a baseline to compare the results of the Flow network and the Combined network. The first step of this network is to have both the anomaly and normal videos input into the 3D ConvNet for feature extraction. The features output from the ConvNet are then separated and averaged into 32 segments. From there the segments are sent to the fully connected network. The fully connected network is where MIL comes into play. The segments of the video which have the highest anomaly score are used to back propagate the network. Essentially, that means the two highest segments of each video will be labeled to train to have a 1. The same is done for normal videos, but in reverse, where the two highest segments will be labeled to train to have a 0. This trains the network so the potential normal segments in an anomaly labeled video will not be wrongly labeled as anomalies. Additionally, makes it so the segments in the normal videos that are closest to anomalies will be correctly labeled as normal, and not throw false flags.

The Flow Network [Figure 1 B] starts by converting all the videos in the dataset. The videos first have the background removed from each frame, then have dynamic flow applied. This creates a video where only has motion in the foreground is applied. The flow images look to be very sparse [Figure 2 B, D] where only the important motion is recorded. Following converting the videos, this network follows the exact same pattern as the RGB network.

The Combined Network makes use of parts from both the RGB network and the Flow Network. This is our proposed two-stream network. This network has an RGB stream and a Flow stream. If you look at [Figure 1 C] you'll notice that the RGB stream is the same as the RGB network until right before the fully connected layer. The same can be said about the flow stream with the flow network. From there the features from the RGB stream is stacked on top of the features from the Flow stream. This allows the fully connected layer to have context of both the image and the motion at the same time. Even though the 3D ConvNet extracts flow data, the flow is a bit abstracted since it also contains spatial and image data. Having

additional pure flow data will help the network make more accurate decisions. Which can be seen from our results.

In order to overcome this difficulty, we continued with the use of UCF-Crime dataset [29]. It consists of long surveillance videos with 13 realistic anomalies. To ensure the authenticity of the datasets, creators of

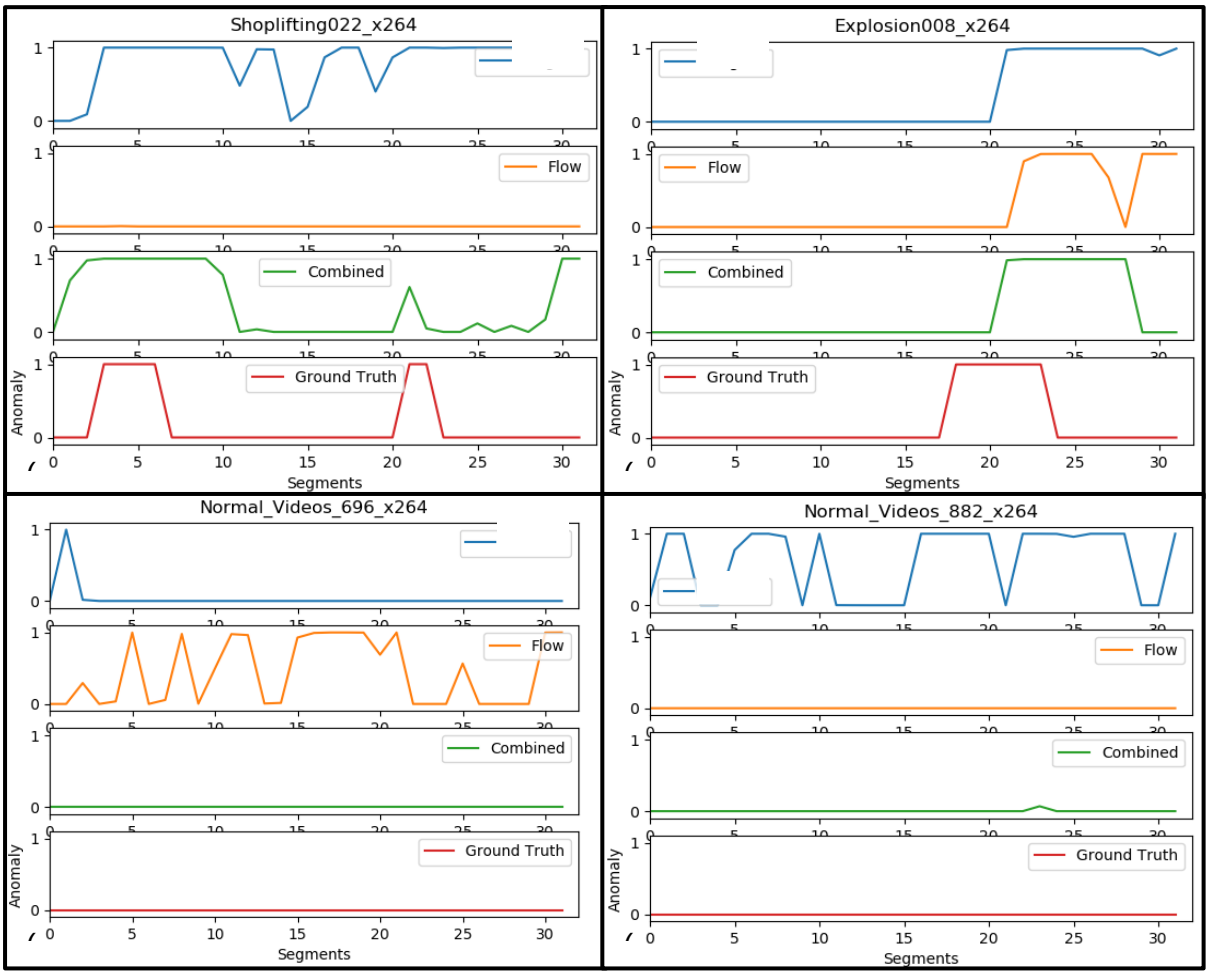


Figure 3: Results comparing the different networks to the ground truth. Notice how in (A)

4. Datasets:

Like the problems encountered by the original paper [1], we noticed similar problems while using UMN datasets [24], UCSD Ped1, and Ped2 datasets [25], Avenue dataset[26], Subway Exit and Subway abnormalities datasets [27], and BOSS dataset [28]. The UMN dataset has five kinds of staged videos. This dataset basically contains people initially walking and then suddenly running in a different direction. The UCSD Ped1 consists of 70 surveillance videos and Ped2 datasets consist of 28 surveillance videos. But these videos are captured only at one location, and are simple and not realistic. Avenue dataset, which has 37 videos, has more anomalies but is taken at a single location. Subway Exit and Subway abnormalities have single dataset each, and has very simple actions. These datasets are small in number, not realistic and variation in abnormalities.

this dataset trained 10 annotators. Videos with ambiguous anomalies were discarded. The dataset has 1900 videos. Of these 1900, the training set would have 800 normal videos and 810 anomalous videos. For the testing set, 150 would be normal and 140 would be anomalous videos. For our anomaly detection method, we need only video-level labels for training the model. For testing the model, we would have to know the start and end frames of the anomalous event.

5. Experiments:

5.1 Implementation Details

Obtaining Dynamic Flow:

Dynamic Flow was obtained by first combining a background reduction algorithm and an optical flow algorithm, then taking the max value for each pixel in each frame over the following 15 frames.

For the background reduction algorithm we used a combination of the algorithms explained in Zivkovic, Z. *et al* [29] and [30]. In tandem with the background algorithm, we applied the optical flow algorithm explained in Farnebäck, Gunnar *et al* [31].

Once both the background reduction and optical flow were combined for each frame, we normalized the flow over the next 15 frames. Which means, we took each pixel from each frame and took the max value from the following 15 frames (half of a second). This helped to give additional context for motion, since on occasion the subtle movements would not persist from frame to frame.

Convolutional Layer:

For feature extraction, we used ResNeXt, a deep 101-layer residual network proposed by Xie *et al.* [35]. ResNeXt was used to extract 2048 features from every 16 frames of each RGB video in the UCF-Crime dataset. We also extracted 2048 features for each 16 frames of the Flow videos. We fuse the feature outputs of both the RGB and Flow stream by concatenating the features into a single 4096 feature vector for each 16 frames.

Segmentation of Features:

The segmentation of features exists in order to have all videos represented in the same shape for the fully connected layer. In order to do this, the features extracted from the convolutional layer are separated into 32 segments. Within each one of those segments all the features are averaged into a single array. So, for all videos the end product after this stage is a 2D array of size 32x2048.

RGB Stream:

This stream uses only the RGB video data that was output from the segmentation of features layer.

Flow Stream:

This stream uses only the Flow video data that was output from the segmentation of features layer.

Combined Stream:

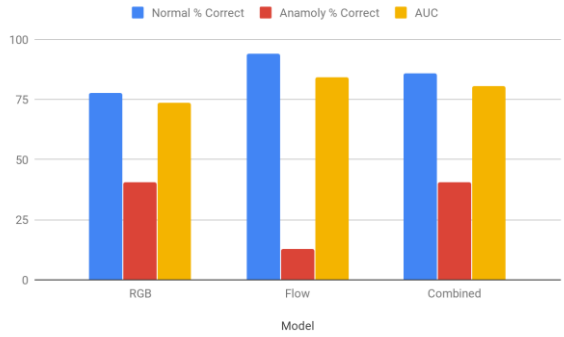
This stream concatenates the video data from the RGB videos on top of the video data from the Flow videos before feeding into the fully connected layer.

Fully Connected Layer:

A 3 layer convolutional network with input dimension of 2048 for the RGB stream and Flow stream, and input size of 4096 for the combined stream, dropout rate of 60%, and activation relu for the first layer. The second layer has an input dimension of 512 with dropout rate of 60%. The third layer has an input dimension of 32 and sigmoid activation. So, the final output should be a value between 0 and 1 where 0 would be normal and 1 would be an anomaly.

Multiple Instance Learning:

This was implemented as a modified loss function in the fully connected layer. It was implemented in the same way as it was in [1]. Essentially, Only the two highest scoring anomaly segment or normal score segment for each video would be used.



5.2 Evaluation Metric

We included metrics for each of the three networks separately. The RGB network can be directly compared to the network in [1]. We used AUC as our evaluation metric. In addition we also included the average percent correct for both normal and anomaly segments.

The reason why we included the two additional metrics was because AUC by itself is misleading. AUC takes into account both normal segments and anomaly segments, however, there are a significantly more normal segments than there are anomaly segments, so the AUC gets skewed towards the normal segments.

5.3 Experimental Results

For our experiments, we divided the UCF-Crime dataset into a testing and training set, using the same split used in [1]. To evaluate the RGB Network, Flow Network, and Combined Network, we trained and tested each model using this split.

Method	Normal % Correct	Anomal % Correct	AUC
RGB	77.9	40.5	73.5
Flow	**94.0**	12.7	**84.3**
Combined	85.6	**41.5**	80.4

Table 1 shows the results for each of our models. The table is broken into three parts: Normal Videos Results, Abnormal Video Result, and Combined Results. We choose to show the individual results for both Normal and Abnormal videos instead of just the Combined results, because we believe that the

combined results do not accurately depict the successes and failures of each model.

When focusing on just anomaly detection, the Two-Stream model and the RGB model both greatly outperformed the Flow model. The Combined model slightly outperformed the RGB model, but the difference is so small that we cannot say this is significant.

The Combined model does greatly outperform the RGB model when comparing their accuracy for identifying Normal segments. The Combined model has almost a 10% increase in accuracy for Normal video segments.

An interesting find during our experimentation was that the Flow-Stream model had a significantly higher accuracy for identifying Normal video segments than both the RGB Network and Combined network. We believe the reason for this is that the flow model frequently predicted video segments to be Normal causing the Normal accuracy to be so high, and the Abnormal accuracy to be so low.

5.4 Comparison with the State-of-the-art

Method	AUC
Hasan *et al.* [2]	50.6
Lu *et al.* [3]	65.51
Sultani *et al.* [1]	75.41
Proposed RGB	73.52
Proposed Combined	**80.34**

To compare our proposed method to other state-of-the-art methods. We use the results reported in [1], since these results are also from the UCF-Crime dataset. Table 2 shows the AUC results for the methods proposed in [1], [2], and [3], along with our own proposed methods. The authors of the previous research did not supply results for Abnormal and Normal videos individually, so here we consider the AUC for the whole test set. Our proposed Two-Stream model significantly outperformed the other methods.

8. Conclusion:

In this paper, we propose a Two-Stream approach to anomaly detection. We create an optical flow version of each video in the UCF-Crime dataset by removing the background from each frame and calculating the motion across multiple frames. Using a Two-Stream pipeline, we extract 2048 features, per each 16 frames, from both the RGB and Flow frames. Each video is segmented into 32 temporal segments in which we average the feature vectors together. To fuse the outputs of both streams, we concatenate the features of each segment so that a segment now consists of 4096 features (2048 from the RGB stream and 2048 from the Flow stream).

For our experimentation, we trained three models: a RGB only model, a Flow only model, and the Two-Stream Combined, RGB and Flow, model. The experimental results show that our proposed Two-Stream model outperforms both Single-Stream models. We also compare our results to the results of recent state-of-the-art methods that were also tested on the UCF-Crime dataset. The Two-Stream model was found to have a higher AUC than the other state-of-the-art methods discussed in [1].

We believe that there are many opportunities for future work. A Long Short-Term Memory (LSTM) network could be used since Abnormal and Normal frames could be viewed as having a temporal relationship. Using a LSTM may prevent the case in which during an anomaly, our model predicts the beginning and end as anomalous, but misses part of the middle. We also believe that other methods could be employed for both the flow and feature extraction. FlowNet 2.0 [33], a recent state-of-the-art Convolutional Neural Network for optical flow extraction, may create better optical flow data for our Flow stream. An alternative feature extraction method also could be employed. For example, the Two-Stream I3D model [34] has recently shown improved results on many activity recognition datasets. This method may extract more meaningful features that better represent the video segments.

References

[1] Sultani, Waqas, et al. “Real-World Anomaly Detection in Surveillance Videos.” Arxiv, 31 Mar. 2018, arxiv.org/pdf/1801.04264.pdf.

[2] Hasan, Mahmudul, et al. “Learning Temporal Regularity in Video Sequences.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 15 Apr. 2016, doi:10.1109/cvpr.2016.86.

[3] Lu, Cewu, et al. “Abnormal Event Detection at 150 FPS in MATLAB.” 2013 IEEE International Conference on Computer Vision, 8 Dec. 2013, doi:10.1109/iccv.2013.338.

[4] Ye, Hao, et al. “Evaluating Two-Stream CNN for Video Classification.” Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR 15, 2015, doi:10.1145/2671188.2749406.

[5] Tran, An, and Loong-Fah Cheong. “Two-Stream Flow-Guided Convolutional Attention Networks for Action Recognition.” 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 12 Nov. 2017, doi:10.1109/iccvw.2017.368.

[6] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, 2014.

[7] Soomro, Khurram, et al. “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild.”

ArXiv:1212.0402v1, 3 Dec. 2012, arxiv.org/pdf/1212.0402.pdf.

[8] Bergeron, Charles, et al. “Multiple Instance Ranking.” ICML '08 Proceedings of the 25th International Conference on Machine Learning , 9 July 2008, citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.8314&rep=rep1&type=pdf.

[9] hx173149. “hx173149/C3D-Tensorflow.” GitHub, 9 July 2018, github.com/hx173149/C3D-tensorflow.

[10] Fan, Yaxiang, et al. “Video Anomaly Detection and Localization via Gaussian Mixture Fully Convolutional Variational Autoencoder.” Arxiv, 29 May 2018, arxiv.org/abs/1805.11223.

[11] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu. Violence detection using oriented violent flows. Image and Vision Computing, 2016.

[12] J. Kooij, M. Liem, J. Krijnders, T. Andringa, and D. Gavrila. Multi-modal human aggression detection. Computer Vision and Image Understanding, 2016.

[13] A. Datta, M. Shah, and N. Da Vitoria Lobo. Person-on-person violence detection in video data. In ICPR, 2002.

[14] S. Mohammadi, A. Perina, H. Kiani, and M. Vittorio. Angry crowds: Detecting violent events in videos. In ECCV, 2016.

[15] J. Kooij, M. Liem, J. Krijnders, T. Andringa, and D. Gavrila. Multi-modal human aggression detection. Computer Vision and Image Understanding, 2016.

[16] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu. Violence detection using oriented violent flows. Image and Vision Computing, 2016.

[17] S. Mohammadi, A. Perina, H. Kiani, and M. Vittorio. Angry crowds: Detecting violent events in videos. In ECCV, 2016.

[18] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In ICCV, 2013.

[19] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In CVPR, 2011.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015. [37] J. Wang, Y. Song, T. L

[22] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016.

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

[24] Unusual crowd activity dataset of university of minnesota. http://mha.cs.umn.edu/movies/crowdactivity-all.avi

[25] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. TPAMI, 2014.

[26] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In ICCV, 2013

[27] Unusual crowd activity dataset of university of minnesota. In http://mha.cs.umn.edu/movies/crowdactivity-all.avi.

[28]http://www.multitel.be/image/researchdevelopment/research-projects/boss.php.

[29] Zivkovic, Z. “Improved Adaptive Gaussian Mixture Model for Background Subtraction.” Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., 2004, doi:10.1109/icpr.2004.1333992. Zivkovic, Zoran, and Ferdinand Van Der Heijden.

[30] “Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction.” Pattern Recognition Letters, vol. 27, no. 7, 2006, pp. 773–780., doi:10.1016/j.patrec.2005.11.005.

[31] Farnebäck, Gunnar. “Two-Frame Motion Estimation Based on Polynomial Expansion.” Image Analysis Lecture Notes in Computer Science, 2003, pp. 363–370., doi:10.1007/3-540-45103-x_50.

[32] Hara, Kensho, et al. “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?” ArXiv, 2 Apr. 2018, arxiv.org/pdf/1711.09577.pdf.

[33] Ilg, E., et al. (2017). FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.179

[34] Carreira, J., & Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.502

[35] Xie, S., et al. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.634

[36] Link to the code to reproduce results of the paper. https://github.com/BrandonRoyalUCF/DeepLearningAnomalyDetection