Automatic Violence Detection through Skeletal-Estimation Based Deep Learning and Video Data Augmentation

Kevin Wedage
University of Alberta
Faculty of Engineering
kwedage@ualberta.ca

Amr Marey
University of Alberta
Faculty of Engineering
amarey@ualberta.ca

Abstract

*Automatic violence detection (AVD) through video surveillance cameras is a vital endeavor to ensure the safety of the public. By automatically detecting violence in video surveillance, this allows emergency services to be deployed to the scene sooner. However, most of the current research literature on AVD does not investigate the robustness of their model under various noise distortions that may be present in the surveillance video stream. We explore the effect of these distortions on different state-of-the-art (SOTA) AVD deep learning (DL) models and show that their performance degrades as the video stream gets distorted. The authors propose a new AVD DL model that revolves around video data augmentation to ensure high performance even under various video stream noise distortions. Achieving an average improved accuracy of **9.83%** on varying amounts of salt and pepper noise, and **10.35%** on Gaussian noise with a standard deviation greater than 4.*

1. Introduction

Video surveillance systems are widespread across public places to survey events and human action. They are used in streets, transportation facilities, airports, stadiums, etc. to record and monitor for the presence of any abnormality, particularly the presence of unexpected human violence [**?**]. With the abundance of video surveillance systems, the need for responsive detection of abnormal and violent scenarios is vital to ensure that immediate counteractions can be taken. By quickly detecting when human violence has occurred, this allows for expedited emergency service responses and in turn the improved safety and general well-being of the parties involved. However, video surveillance systems typically generate a large amount of live data and it can be challenging for the human camera operator to manually detect for the rare occurrences of violence. This issue is especially prominent when the human operator is required to monitor multiple live surveillance cameras simultaneously. Hence, there is a strong demand for automatic violence detection (AVD) mechanisms to increase the effectiveness of video surveillance systems [**?**].

The research field of AVD is considered a subfield of the human activity recognition (HAR) field, which aims to detect different human activities from different video data. The prior research literature on HAR focused on using consecutive video frames to detect and track the body parts of the persons in the video [**?**]. [**?**] utilizes spatio-temporal descriptors to model each video as a bag-of-features for a support vector machine (SVM) to classify the video as violent or non-violent. [**?**] and [**?**] present the usage of the histogram of oriented gradients (HOG) and the histogram of optical flow (HOF) for body-part tracking. Unfortunately, these methods tend to suffer when the video is recorded under poor lighting conditions or in the event of occlusion [**?**]. There is some research literature that proposes the usage of depth sensors to resolve this issue [**? ? ?**]. However, depth sensors tend to be noisy and would substantially increase the financial costs of the video surveillance systems [**?**].

Over the past few years, there has been a trend in applying deep learning (DL) for AVD. DL allows for many advantages such as automated feature extraction, scalability in model complexity, and robustness to new data [**?**]. Convolutional neural networks (CNN) have become widespread in the field of computer vision due to their strong performance in image-related tasks. CNNs are a popular field of research in the current literature due to the growth of computing power and the presence of big-data. They have been extensively utilized for HAR as they are robust to changing lighting conditions, variations in video background, occlusions, etc [**?**]. Another prominent DL architecture is the long-short term memory (LSTM) neural network. LSTMs have shown strong performance with tasks involving time-series data, which makes them suitable for HAR [**?**]. [**?**] introduced the ConvLSTM, which is a DL architecture that aims to combine the spatial processing capabilities of the

CNN and time-series processing capabilities of the LSTM, making them perfect for video classification.

The most prominent difficulty in using DL models is the requirement of having a large amount of (typically) labeled data. The cost of gathering this data is often non-trivial and can be a combination of time costs, economic costs, and legal costs. Fortunately, data augmentation (DA) can mitigate this challenge by generating a larger dataset from a smaller dataset. One can apply color or geometric transformations to generate new data. Recently, data randomization and simulation-based methods have been proposed for video DA [**?**].

This paper aims to explore the effects of geometric and noise-injection DA transformations on SOTA AVD models. The baseline SOTA AVD model is taken from [**?**] due to its computational efficiency and competitive performance. The following are the contributions of this work:

1. The proposal of applying general DA transformations, specifically noise-injection DA transformations, to increase the robustness of AVD models under various distortions.
2. The assessment of when SOTA AVD models can no longer perform sufficiently due to the degradation of video quality.
3. The comparison of the baseline DL AVD model with the DA DL AVD model across different artificially corrupted datasets to show the generalizability of DA DL model in comparison to the baseline model.

The rest of this paper is divided as follows. The related work is discussed in section 2. The DA methods considered in this paper are presented in section 3. The baseline model architecture is presented in section 4. The experiments and results are shown in section 5. Section 6 presents some future work to consider. Finally, the paper is concluded in section 7. The authors show their individual contributions and the code used for this report in section 8.

2. Related Works

2.1. Recent DL-based AVD Models

There are a lot of papers that discuss applying DL for AVD in various ways. [**?**] presents a method that encodes the difference between consecutive frames using a ConvLSTM architecture. This model was shown to perform better compared to a model trained on raw image frames. [**?**] took a model revolving around XceptionNet [**?**] to derive features from the video stream and applied a bidirectional LSTM to explore the features in the temporal space. [**?**] presented a three-stream CNN for AVD. The three streams took the RGB image, optical flow, and person-to-person acceleration as inputs. It was hypothesized that the person-to-person acceleration stream would extract the high-energy segments present in violent videos. [**?**] utilizes a two-stream ConvLSTM structure. The first stream is designed to take the RGB images as input and extract the human poses present in the images and the second stream consists of the change detection pipeline. The change detection pipeline is intended to use the inter-frame information to help determine whether violence is present in the video.

2.2. Data Augmentation

DA is a powerful tool to enhance DL models. It can increase the size of the training dataset and generalize the input data to decrease the chances of over-fitting. Furthermore, it can also be utilized to test the model under perturbations that were not present in the original training or test data [**?**]. DA can be divided into several classes [**?**]. A few of them are introduced in this section.

One can augment video data by applying basic geometric transformations. Some examples of simple geometric transforms include scaling, translating, rotating, and flipping the image [**?**]. [**?**] explores the use of noise-injection into images to generate more robust models. Noise-injection of a given type of noise, allows the model to be more robust against that same form of noise. Furthermore, noise-injection allows neural networks to learn better by occluding irrelevant features. Intuitively, by injecting certain forms of noise, and keeping the ground truth label the same, the model is encouraged to discard the changes introduced by the noise and focus on underlying input features that remain consistent. [**?**] utilizes temporal data augmentation by iteratively temporally cropping frames from the original video sequence. One can also apply image-mixing techniques, such as those presented in [**?**], to mix different images from the same dataset. [**?**] presents a data augmentation method by warping certain image frames through optical flow fields. This paper will focus primarily on the basic transformation DA class.

There has been recent literature on using generative adversarial networks (GANs) to augment video datasets [**?**]. For example, CrowdGAN [**?**] is a GAN that synthesizes crowd videos from a few initial context frames. CrowdGAN has two modules. The first module is responsible for predicting the following frame while the other module predicts the optical flow map to warp the starting frame. The two modules are fused together to generate the next frame. This process is iterated by using the following output frames as inputs.

Graphic cards and real-time rendering systems are continuously being improved each year. One can use a graphic and a physics engine such as Unity [**?**] or Unreal Engine [**?**] to produce synthetic videos that are of high-quality. These engines also come with a programmable user-interface allowing developers to generate augmented video datasets with them. This has been used for training robotic reinforcement learning algorithms in simulation as

(a) $f(m, n, t)$

(b) $\bar{f}(m, n, t)$

Figure 1. The effect of horizontal flipping.

training these algorithms in the real-world can be dangerous [**?**].

3. Common DA Geometric Transformations

This section presents the DA geometric transformations considered in this paper.

3.1. Horizontal Flipping

This data augmentation technique revolves around reflecting the video over the vertical axis. Assume that the video resolution is of size $M \times N$. Hence

$$\bar{f}(m, n, t) = f(m, N - n, t).$$

An example of horizontal video flipping is presented in Figure 1.

3.2. Scalar Video Multiplication

This data augmentation technique modifies the brightness of the training video such that it is either brighter or dimmer compared to the original video. This is done by multiplying the original video signal by a positive scalar α. Hence

$$\bar{f}(m, n, t) = \begin{cases} 255 & \text{if } \alpha f(m, n, t) > 255 \\ \alpha f(m, n, t) & \text{if } \alpha f(m, n, t) < 255 \end{cases}.$$

If $\alpha > 1$ then the augmented video becomes brighter than the original video. If $\alpha < 1$ then the augmented video becomes dimmer than the original video. One must ensure that $\alpha \approx 1$ as if α is too large $(\alpha >> 1)$, then the video would be a white screen; if α is too small $(\alpha << 1)$, then the video would be a black screen. In these scenarios, the augmented data would lead to detrimental model performance as the data no longer contains relevant information to predict the corresponding ground truth label. Some examples of scalar video multiplications are presented in Figure 2 .

(a) $\alpha = 0.7$

(b) $\alpha = 1.$

(c) $\alpha = 1.3$

Figure 2. The effect of scalar video multiplication on the original video.

3.3. Additive Gaussian Noise

Additive Gaussian noise consists of additive noise that is sampled from the normal distribution $\mathcal{N}(0, \sigma)$. Hence, the noisy video can be described as

$$\bar{f}(m, n, t) = f(m, n, t) + \mathcal{N}(0, \sigma).$$

Where $\mathcal{N}(0, \sigma)$ originates from the Gaussian probability density function (PDF).

$$\mathcal{N}(x, 0 : \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{x-\mu}{\sigma})^2}.$$

Where μ is the expected value of the PDF (it is assumed that $\mu = 0$), and σ is the standard deviation of the Gaussian distribution. As one increases the value of σ, the augmented video becomes blurrier as shown in Figure 3.

Gaussian noise can occur naturally in videos that take place under poor lighting conditions, or due to video compression, and video broadcasting.

3.4. Salt and Pepper Noise

Salt and pepper noise consists of massive noise disruptions on random pixels located throughout the image such that these pixels become either entirely white or black. A pixel has a chance to be distorted with probability ϵ. Hence the noisy video can be described as:

$$\bar{f}(m, n, t) = \begin{cases} 0 & \text{with probability } \epsilon \\ f(m, n, t) & \text{with probability } 1 - 2\epsilon \\ 255 & \text{with probability } \epsilon \end{cases}.$$

Figure 3. The effect of Gaussian noise on a video frame.

If $\bar{f}(m,n,t) = 0$ then the associated pixel is black. Else if $\bar{f}(m,n,t) = 255$ then the associated pixel is white. An example of salt and pepper noise is presented in Figure 4.

One can quantify the amount of salt and pepper noise present by the ratio R. Where

$$R = \frac{K}{MN}.$$

Where K is the number of pixels in the photos that are uncorrupted. As R increases to one, the image becomes less noisy. Salt and pepper noise arises when transmitting images over noisy links.

Figure 4. The effect of salt and pepper noise on a video frame.

4. Baseline Model Architecture

The DL model that is utilized in this work is first introduced in [**?**]. The architecture is showcased in Figure 5. The architecture consists of two distinct pipelines, the RGB Pipeline and the Change Detection Pipeline. The RGB Pipeline is intended to extract intra-frame information. Whereas, the Change Detection Pipeline is intended to extract inter-frame information. Both pipelines take in the raw videos as input. The RGB Pipeline utilizes an external model, in this case, Openpose [**?**], to extract the basic skeleton of the humans detected in the frame. The non-human skeleton pixels are set to black to reduce computation and facilitate learning. An example of pose estimation using Openpose [**?**] is presented in Figure 6.

The Change Detection Pipeline is a secondary pipeline that takes in the raw input videos, and attempts to extract information between the frames. There are several different methods to determine the change between consecutive frames, but similar to [**?**], the element-wise difference between frames is computed. An example of this is showcased in Figure 7.

Replicating the results in [**?**], the two distinct pipelines are combined in a manner to temporally aggregate the sequences of frames. A 2-dimensional convolution layer is first applied to the output of the RGB layer. This consists of 9 filters, and a 3x3 kernel, and uses the ReLU activation function. Following this, batch normalization is utilized. In the Change Detection Pipeline, consecutive frames are subtracted together, then passed through a batch normalization layer, and finally through a 2-dimensional ConvLSTM layer, which consists of 9 filters, a 3x3 kernel, and uses the tanh activation function. The two pipelines are fused using point-wise addition. The fused feature map can be described as

$$F_c = \text{BatchNorm}(F_{RGB}) \oplus \tanh(F_{CD}).$$

Where F_c is the fused feature map, F_{RGB} is the feature map of the RGB pipeline, and F_{CD} is the feature map of the change detection pipeline.

The combined output F_c is passed through another ConvLSTM layer with 32 filters and a 3x3 kernel. The output of this is passed through a depth-wise 2-dimensional convolutional layer with a 3x3 kernel, and depth multiplier of 2, using ReLU activation function. The output is average using a 2-dimensional global averaging pooling layer. Finally, this output passes through 3 dense layers. The first with 128 units, then 16, and finally 1. The output of the last dense unit provides the binary classification. The above configured

5. Experiment and Results

The task of an AVD model is a binary classification problem, whereby the input is a video clip, and the output is either 1 or 0, where 1 indicates that the video contains human violence and 0 otherwise. In Table 1, the main datasets that have been used in prior literature for AVD are outlined. The authors primarily focused on using the RWF-2000 [**?**] dataset. The reason being that the RWF-2000 dataset best aligns with the intended use case presented in this paper, as it contains videos obtained primarily from surveillance systems that are labeled with either the class of Fight or Non-Fight. In addition, the dataset consists of 2000 videos, which exceeds most other AVD datasets.

5.1. Performance on RWF-2000

A baseline model is first trained following the procedure and training procedures discussed in [**?**]. This involves running Openpose [**?**] on the *avi* formatted videos of the

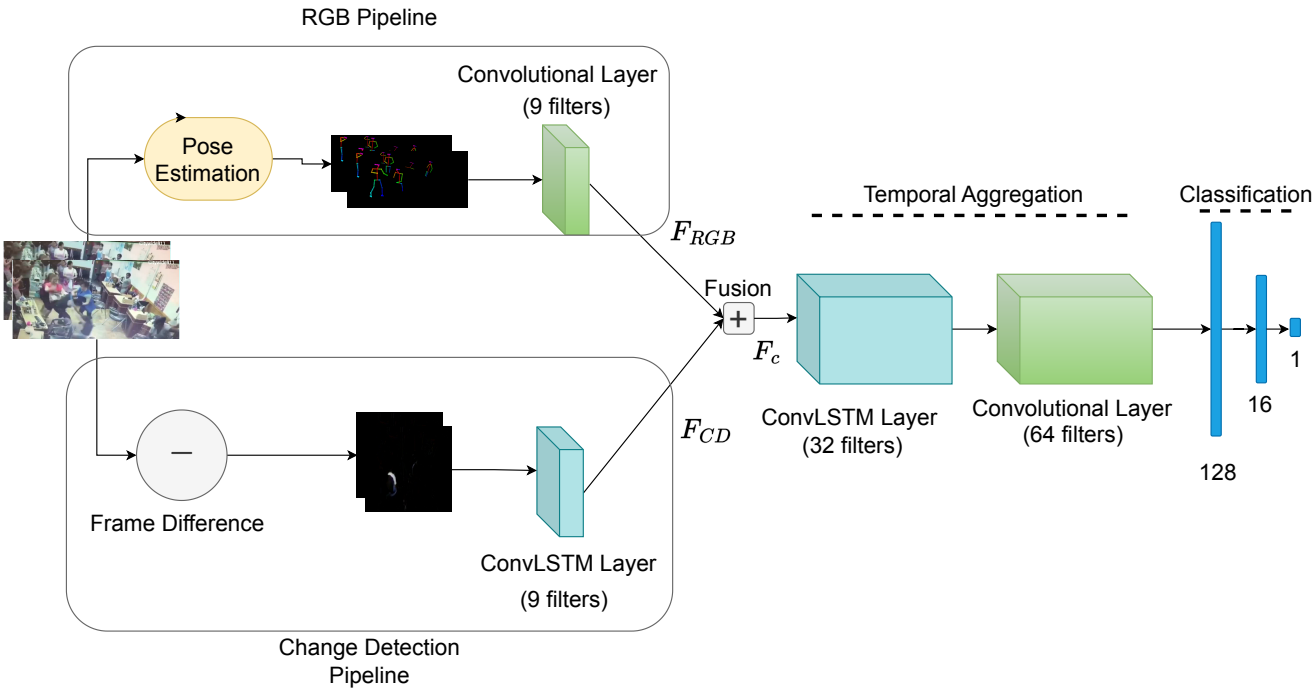
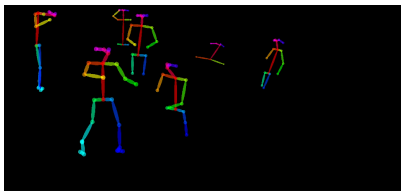


Figure 5. Architecture of the DL model utilized in this paper that was replicated from [**?**].

Dataset	Number of Videos	Basic Remark
RWF-2000 [**?**]	2000	Consists of violent/non-violent videos taken from surveillance cameras.
RLVS-2000 [**?**]	2000	Consists of violent/non-violent videos collected from YouTube.
Hockey Fights [**?**]	1000	Dataset consisting violent/non-violent scenarios occurring in hockey games.
Movie Fights [**?**]	200	Dataset consisting violent/non-violent scenes taken from movies.
Crowd Fights [**?**]	246	Dataset consisting violent/non-violent situations occurring in public crowds

Table 1. Datasets considered in this paper.

(a) Original image of several people.



(b) Openpose example of several people, with background removed.

Figure 6. Comparison between original and Openpose images.

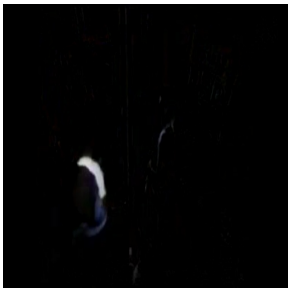
RWF-2000 dataset. It is important to note that the RWF-2000 dataset contains only two splits, named *training* and *validation*. As outlined in the RWF-2000 repository, the authors renamed the *validation* split to *test* and created a validation split from a random sample of the original training split. This random sample contained 100 examples of violent and non-violent videos each, totaling 200 videos which corresponds to 10% of the entire dataset.

The baseline model was implemented as outlined in section 4. The baseline model was trained for 30 epochs on our specific training split and reported the training and validation loss and accuracy in Figure 8. The Adam [**?**] optimizer was used with a learning rate of 0.001, a β_1 value of 0.9, a β_2 value of 0.999, and an ϵ of 1e-7.

For our DA DL model, the training procedure above were repeated, which resulted in the training and validation loss shown in Figure 9. The only difference is that the DA variation of the RWF-2000 training set were utilized. This consisted of horizontal flipping, adding varying amounts of

(a) Example frame 1 of a video.

(b) Example frame 2 of a video.



(c) Element-wise difference between frame 2 and frame 1.

Figure 7. Change detection example computation.

Gaussian, salt and pepper, and multiplicative noise to each input video. From the original 1400 training videos, 5600 extra DA videos were obtained. The authors refrained from data augmenting the validation split used to determine the best model checkpoint. This is to ensure that regardless of the changes made, the model should still optimize performance towards the original dataset.

For both models, after the initial 30 epochs of training, last checkpoint was used and retrained for an additional 20 epochs as suggested from [**?**]. For the last 20 epochs, the checkpoint of the model that resulted in the best validation accuracy was saved. For the last 20 epochs of both models, the non-data-augmented training and validation splits were used.

The final reported accuracy was obtained by evaluating both models on the test split which resulted in an accuracy of **87.00%** for the baseline and **87.25%** for the DA variation. Note that the decreased accuracy obtained on the baseline, in comparison to the literature [**?**], is potentially a result of using a separate validation and test split, in comparison to the literature which used the same for both.

The robustness of the baseline model is investigated further by evaluating it on varying degrees of artificially corrupted test splits.

5.2. Gaussian Corrupted Test Set

By injecting varying amounts of Gaussian noise into the test split, starting at no noise, then injecting Gaussian noise with a $\sigma = 5$ up to $\sigma = 18$, one can determine how well the baseline model will do under Gaussian noise. In Figure 10, the performance of the Baseline DL model and the DA DL model (Ours) were compared. The Baseline and DA DL models both perform equally at no noise and up to around $\sigma = 4$. However, beyond this point, as the standard deviation of the noise increases, the DA DL model consistently outperforms the Baseline by a significant amount, and on average is **10.35%** higher accuracy.

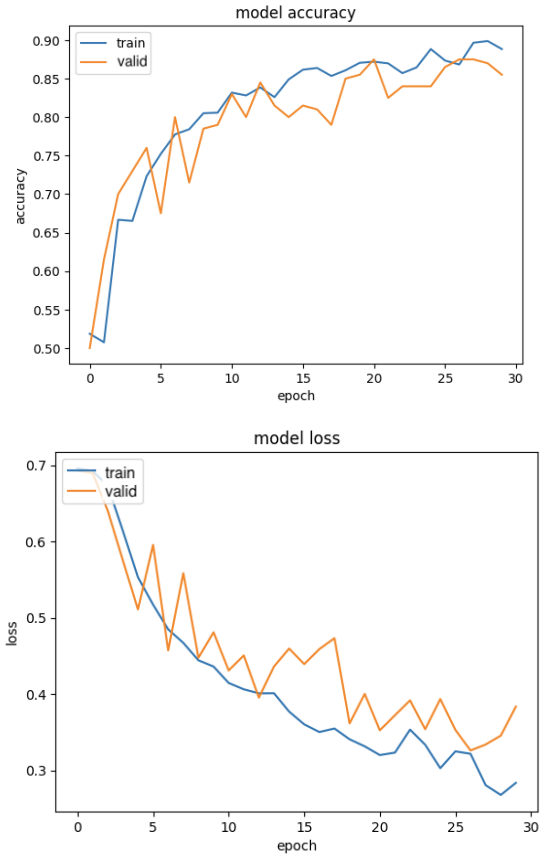


Figure 8. Baseline model trained for 30 epochs on the RWF-2000 dataset. Note *valid* refers to the validation split.

An interesting observation, when the σ gets very large ($\sigma \geq 18$) the videos become unrecognizable to a human observer. Consequentially, the resulting prediction of either model tends towards predicting non-violence. However, since the dataset is equally split between violence and non-violence, the model's accuracy is lower bound by 50%.

5.3. Salt and Pepper Corrupted Test Set

Similarly, by injecting varying amounts of salt and pepper noise into the test split, starting with a $R = 0.05$ up to $R = 0.55$, one can determine how well the baseline model will do under salt and pepper noise. In Figure 11, the performance of the Baseline DL model and the DA DL model (Ours) are compared. Interestingly, at $R = 0.05$ which is

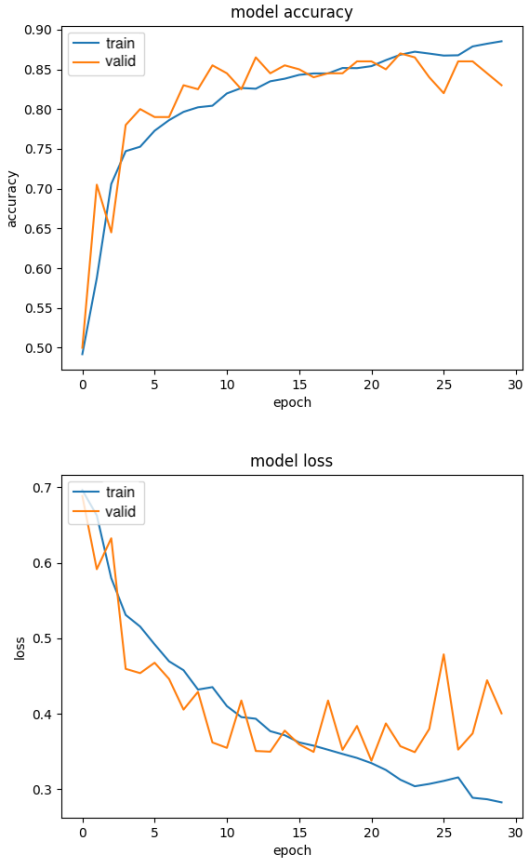


Figure 9. DA model trained for 30 epochs on our DA RWF-2000 dataset. Note *valid* refers to the validation split.

the largest amount of salt and pepper noise that was experimented with, the results of both models are fairly similar sitting around 71%. However, as the amount of noise decreases, there again begins to be a gap between the Baseline and DA DL model. Whereby, on average the DA DL model has an accuracy **9.83%** higher than the baseline.

6. Future Work

This paper showcases the importance of using DA to increase the robustness of models, and specifically AVD models. For future work the authors intend to explore additional forms of DA through the application of GANs or physics engine simulators.

Furthermore, as the amount of noise, used in the DA, increases, the RGB Pipeline suffers significantly. This is primarily due to the inability of Openpose [**?**] to detect human skeletons at such a high degree of video corruption. Specifically, this occurs when there is large amounts of Gaussian noise in the video. The authors suggest exploring training Openpose [**?**] with our AVD model in an end-to-end format with DA videos.

7. Conclusion

There is a high demand for automatic violence detection (AVD) algorithms due to the substantial amount of violence occurring in public settings. AVD algorithms allow for expedited detection of violence as compared to having a human video surveillance camera operator. This paper aims to investigate video data augmentation (DA) for automatic violence detection models (AVD). The use of DA prevents the AVD model from over-fitting to training data and increase the model robustness to different perturbations. The authors specifically focus on DA that revolve around basic geometric transformations such as flipping the video, scaling the brightness of the video, injecting different forms of noise (such as salt and pepper noise and Gaussian noise). It has been showed that the DA AVD model out-performed the non-data-augmented AVD model under common distortions typically present in video surveillance systems.

8. Contributions and Implementation Details

Kevin Wedage primarily focused on training the various models, running the experiments, and gathering the results. This included obtaining the Openpose input videos and configuring the training scripts for each experiment. The experiments were run using a rented GPU service provided from JarvisLabs.ai, and primarily used 4xA5000 GPUs. Each GPU has 24 GB of VRAM.

Amr Marey focused on doing the literature survey on automatic violence detection (AVD) models and video data augmentation (DA) techniques. In the earlier weeks , he would present different AVD models and DA techniques in group meetings and discuss them with Kevin. The advantages and disadvantages were discussed. He also focused on augmenting the videos and developing the procedures to augment those videos. Some of the augmentations techniques that were experimented did not yield fruitful results hence they were discarded but they were still an important part of our research process.

A link to the code is available at the following GitHub repository kdwedage/Automatic-Violence-Detection-with-Data-Augmentation.

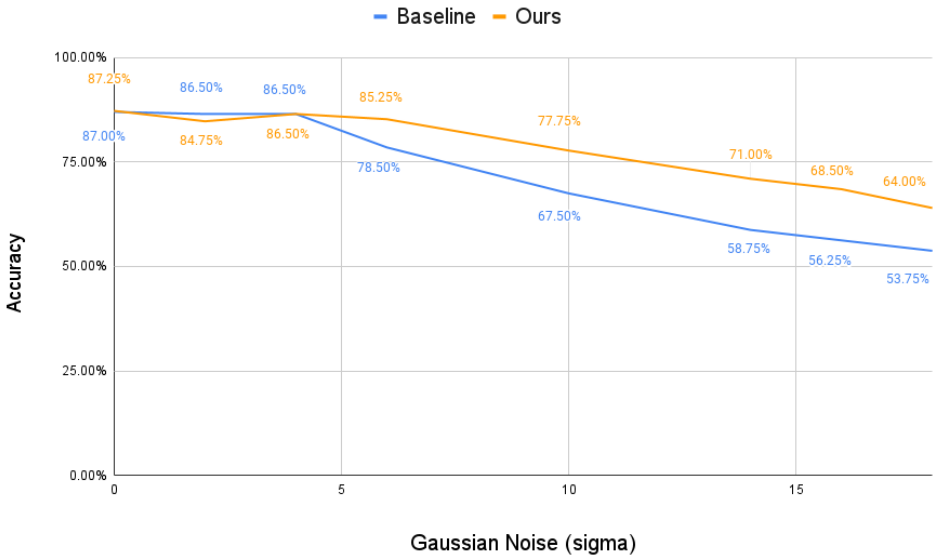


Figure 10. Baseline DL model compared with the DA DL model (Ours) on varying degrees of Gaussian noise for the test split of RWF-2000.

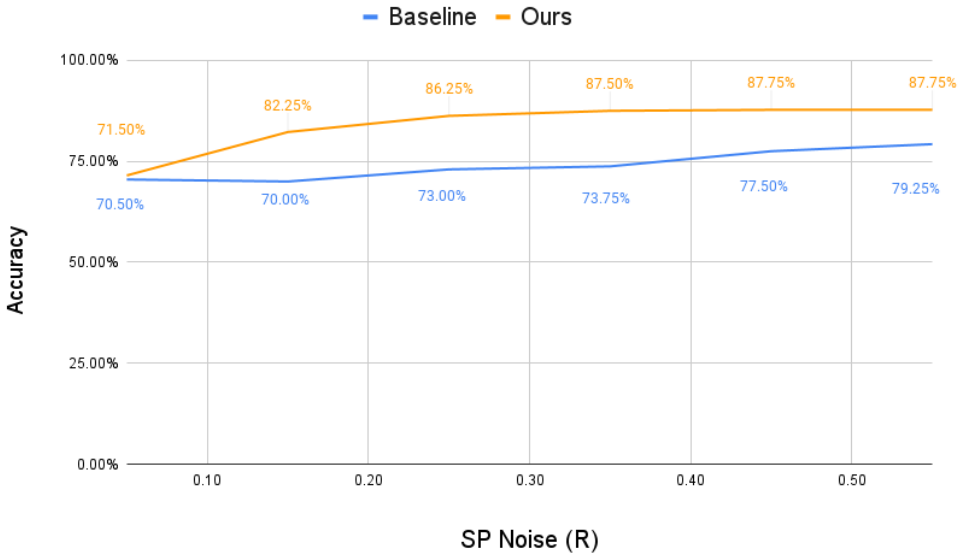


Figure 11. Baseline DL model compared with the DA DL model (Ours) on varying degrees of salt and pepper noise for the test split of RWF-2000.