Violence Detection using 3D Convolutional Neural Networks

Jiayi Su, Paris Her, Erik Clemens, Edwin Yaz, Susan Schneider
Marquette University
{jerry.su,paris.her,erik.clemens,edwin.yaz, susan.schneider}@marquette.edu

Henry Medeiros
University of Florida
hmedeiros@ufl.edu

Abstract

Accurate detection of abnormal behavior can help improve public safety. In this work, a 3D convolutional neural network (CNN) is implemented to detect violence captured by surveillance cameras. A comprehensive study of model hyper-parameter tuning is addressed to show competitive violence detection results using a general action recognition CNN without modifying the original architecture. Experimental results on three publicly available benchmark datasets show that the proposed method outperforms other sophisticated techniques designed specifically to detect violence in videos. Our analysis further indicates that reasonable network parameter adjustments can be an effective mechanism to guide the design of computer vision models in abnormal human behavior detection.

1. Introduction

Surveillance cameras have been widely deployed in many areas such as schools, shops, stadiums, and streets [39]. The main function of these cameras is to record evidence of abnormal human behaviors, such as acts of violence or theft. Usually, videos are transmitted to a local data visualization center and displayed to help identify conditions and necessary actions [26]. However, it is impractical for human operators to simultaneously monitor multiple surveillance videos, especially in large venues where hundreds or even thousands of cameras may be deployed. In practice, surveillance systems operators have to periodically switch surveillance feeds to monitor different locations. This inefficient switching reduces the probability that an operator is viewing the appropriate camera feed when an abnormal behavior occurs. On the other hand, automated vision-based abnormal event detection techniques can reduce the cognitive load imposed upon surveillance systems operators, the time to respond to an incident, and the probability that such incidents are overlooked altogether [23]. To achieve this goal, vision-based detection techniques can be deployed to help identify suspicious activities in surveillance video. Information from surveillance feed can be transmitted to corresponding operators for an immediate and effective response.

This work implements a vision-based detection technique to identify abnormal behaviors automatically from videos captured by surveillance cameras. Specifically, a 3D CNN that is designed for activity recognition is implemented and fine-tuned to detect violence in real time [8]. To achieve state-of-the-art (SOTA) detection accuracy, the model is pretrained for the more general task of activity recognition and then fine-tuned using a Bayesian hyper-parameter search method [14] for datasets that contain abnormal behaviors specifically. Ultimately, the fine-tuned model can be deployed to predict violence on a video sequence from the surveillance cameras, and the network predictions can be used to display focused, relevant video data for corresponding security staff.

In this work, one specific type of abnormal behavior is investigated: violence detection in videos. Commonly observed abnormal behaviors include one or more individuals pushing or kicking. Unlike image classification problems, temporal information plays an important role in activity detection [33], because most actions consist of different motions across consecutive frames, and different actions might appear very similar in a still image. For example, it may be hard to distinguish if a person is jogging or walking given a still image. Therefore, using temporal information is an effective strategy to produce accurate recognition results. Hence, networks that are able to exploit temporal information can generally provide more accurate action recognition results. One technique to employ temporal information for

978-1-6654-6382-9/22/$31.00 ©2022 IEEE

accurate action recognition consists of 3D convolutions [33]. Extended from 2D models, 3D kernels are designed to deal with information from both the spatial and temporal domains in the same manner. When the information from the temporal domain is considered, a 3D network is able to capture complex motion information compared to standard 2D networks. As a consequence, a variety of 3D networks have been designed solely for action recognition problems [5, 33, 34].

It has become increasingly popular to employ a common architecture to solve a variety of problems [38]. In this work, extensive experimental analysis of the modified network [8] is performed. Modified for violence detection in videos, experimental results on three violence detection datasets [2, 7, 9] demonstrate that our modified model outperforms most other violence detection methods with simple hyper-parameter adjustments. Our results also provide insights into the design of effective video classification models for future research.

2. Related Work

Most modern action recognition models are based on 3D convolutions. In [35], a 3D convolution is factorized into a combination of a 2D convolution in the spatial domain, followed by a 1D convolution in the temporal domain. A channel-separated convolutional network is proposed in [34] to reduce the computational cost of 3D convolutions. In [21], by shifting part of the channels along the temporal dimension, temporal information between neighboring frames is exchanged to obtain a higher accuracy. In [11], 3D kernels with multi-scale temporal length were introduced, with kernels scaled at different temporal length, the model obtained the ability of learning actions with both short- and long-duration, which brings more accurate recognition results. In [27], a two-stream 3D network is designed to recognize actions by fusing two identical networks at the output layer, where one stream is focused on extracting spatial information from video sequences and the other is designed for obtaining temporal information from the corresponding optical flow [4].

Many 3D networks have also been proposed specifically for violence detection. In [30], a 3D skeleton point cloud module is introduced to model the interactions between skeleton points and detect abnormal behaviors. Moreover, [15] proposed a Motion Saliency Map (MSM) module that highlights moving objects, as well as a Temporal Squeeze-and-Excitation (T-SE) block [10]. Along with that, [37] also utilize audio features to help recognize violence.

3. X3D Network for Violence Detection

In this work, the X3D network [8] is used to detect abnormal behaviors in videos. X3D is a 3D network designed using the Neural Architecture Search (NAS) technique [31].

Table 1. General X3D architecture [8]. $\{\gamma_\tau, \gamma_t, \gamma_s, \gamma_w, \gamma_b, \gamma_d\}$ are scaling factors described in Section 3. The dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes. Strides are denoted as {temporal stride, spatial stride^2}. The model can be expanded by using different scaling factors based on the needs of different applications.

stage	filters	output sizes $T\times S^2$
data layer	stride γ_τ, 1^2	$1\gamma_t\times(112\gamma_s)^2$
conv$_1$	1×3^2, 3×1, $24\gamma_w$	$1\gamma_t\times(56\gamma_s)^2$
res$_2$	$\begin{bmatrix} 1\times1^2, 24\gamma_b\gamma_w \\ 3\times3^2, 24\gamma_b\gamma_w \\ 1\times1^2, 24\gamma_w \end{bmatrix}\times\gamma_d$	$1\gamma_t\times(28\gamma_s)^2$
res$_3$	$\begin{bmatrix} 1\times1^2, 48\gamma_b\gamma_w \\ 3\times3^2, 48\gamma_b\gamma_w \\ 1\times1^2, 48\gamma_w \end{bmatrix}\times2\gamma_d$	$1\gamma_t\times(14\gamma_s)^2$
res$_4$	$\begin{bmatrix} 1\times1^2, 96\gamma_b\gamma_w \\ 3\times3^2, 96\gamma_b\gamma_w \\ 1\times1^2, 96\gamma_w \end{bmatrix}\times5\gamma_d$	$1\gamma_t\times(7\gamma_s)^2$
res$_5$	$\begin{bmatrix} 1\times1^2, 192\gamma_b\gamma_w \\ 3\times3^2, 192\gamma_b\gamma_w \\ 1\times1^2, 192\gamma_w \end{bmatrix}\times3\gamma_d$	$1\gamma_t\times(4\gamma_s)^2$
conv$_5$	1×1^2, $192\gamma_b\gamma_w$	$1\gamma_t\times(4\gamma_s)^2$
pool$_5$	$1\gamma_t\times(4\gamma_s)^2$	$1\times1\times1$
fc$_1$	1×1^2, 2048	$1\times1\times1$
fc$_2$	1×1^2, #classes	$1\times1\times1$

Similar to the strategy used in the design of EfficientNet [32], after obtaining the base model, X3D is scaled in multiple dimensions to improve its performance. Scaling aspects include temporal duration γ_t, frame rate γ_τ, spatial resolution γ_s, width γ_w, bottleneck width γ_b, and depth γ_d. By introducing these scaling factors, it allows designers to meet the desired balance in the computation/accuracy trade off based on real world applications. X3D shows competitive results on various datasets [16, 28] while maintaining a relatively low computation cost. To mitigate overfitting, a dropout layer is incorporated before the last fully connected layer [8]. Table 1 [8] shows the general architecture of the X3D network.

As shown in Table 1, the model contains two convolutional layers (conv$_1$, conv$_5$), four residual blocks (res$_2$ $\sim$ res$_5$), and two fully connected layers (fc$_1$, fc$_2$). The input video sequence is first sampled with a stride of γ_τ, and then the sampled video is sent to the model. Notice that to obtain as many temporal features as possible, the temporal duration γ_t remains the same for all stages, meaning there is no temporal down-sampling within the model.

3.1. Model Input Modifications

Since [18] demonstrated the effectiveness of random spatial cropping for data augmentation, it has been deployed in many settings, including X3D for video-based action recognition [8] and in [13, 22] for video violence detection. Such

augmentation can be appropriate in cases where context influences classification (for example, a “golf putting" video that is cropped to show only the golf course can still be classified correctly). However, random cropping is an inappropriate augmentation choice for violence detection in computer vision methods. In surveillance footage, relevant information may only occupy a small portion of the frame, including frame edges. Extracting a random crop may therefore completely remove violence information from a video clip. The data labels would then encourage the network to classify such a crop as violent, leading to the use of incorrect features for classification.

To preserve the benefits of spatial augmentation without cropping the input frames, we apply a custom augmentation technique we call *resizing within*. Given the model's frame input width w, height h, and pixel area $A = w \times h$, this technique uses interpolation, rather than cropping, to resize the original input video to a new width, height, and area $w' \times h' = A' \leq A$. Hyper-parameters control the scale, S', and aspect ratio, AR', of the rescaled frames, as seen in Eq. (1), where $U \sim (l, u)$ denotes a uniform distribution between the upper and lower bounds l and u.

$$\begin{aligned} S' &= A'/A, \\ AR' &= w'/h', \\ S' &\sim U\left(S'_L, 1.0\right), \\ AR' &\sim U\left(AR'_L, AR'_U\right). \end{aligned} \tag{1}$$

The rescaled video is randomly placed within the input frame and the remaining space $A - A'$ is zero-padded, as seen in Figure 1. While we use 240×320 input frames to match the common aspect ratio of $4:3$, some aspect ratio distortion does occur during validation and training. Sampling AR' randomly during training attempts to address this distortion, as well as distortions caused by camera placement.

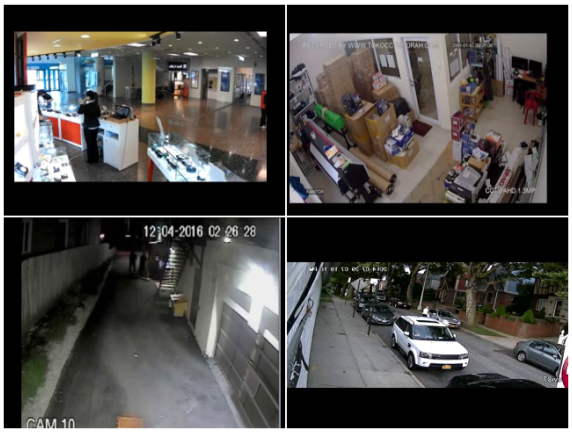


Figure 1. Training frames from [7] after resizing within.

The temporal sampling strategy in the X3D [8] architecture is modified to fit our data, where we make one prediction from one temporal window, and consequently reduce computation cost. For validation on datasets with a variable number of frames per clip (Surv/SCFD and ViolentFlows), we average the softmax predictions of 5 uniformly-spaced temporal sub-clips, which allows our pre-processing procedure to adapt the temporal spacing of the sub-clips to better fit videos of varying lengths. For the RWF-2000 dataset, which has a consistent number of frames per video, we use a single validation clip and tune the temporal subsampling rate τ as an additional hyper-parameter.

Table 2. Modified X3D-S architecture used in this work.

stage	filters	output sizes $T \times H \times W$
data layer	stride τ, 1^2	$13 \times w \times h$
conv_1	1×3^2, 3×1, 24	$13 \times \frac{w}{2} \times \frac{h}{2}$
res_2	$\begin{bmatrix} 1 \times 1^2, 54 \\ 3 \times 3^2, 54 \\ 1 \times 1^2, 24 \end{bmatrix} \times 3$	$13 \times \frac{w}{4} \times \frac{h}{4}$
res_3	$\begin{bmatrix} 1 \times 1^2, 108 \\ 3 \times 3^2, 108 \\ 1 \times 1^2, 48 \end{bmatrix} \times 5$	$13 \times \frac{w}{8} \times \frac{h}{8}$
res_4	$\begin{bmatrix} 1 \times 1^2, 216 \\ 3 \times 3^2, 216 \\ 1 \times 1^2, 96 \end{bmatrix} \times 11$	$13 \times \frac{w}{16} \times \frac{h}{16}$
res_5	$\begin{bmatrix} 1 \times 1^2, 432 \\ 3 \times 3^2, 432 \\ 1 \times 1^2, 192 \end{bmatrix} \times 7$	$13 \times \frac{w}{32} \times \frac{h}{32}$
conv_5	1×1^2, 432	$13 \times \frac{w}{32} \times \frac{h}{32}$
pool_5	$13 \times 5 \times 5$	$1 \times 1 \times 1$
fc_1	1×1^2, 2048	$1 \times 1 \times 1$
fc_2	1×1^2, #classes	$1 \times 1 \times 1$

In Table 2, the modified X3D model, specifically X3D-S, is shown. Different from the original model, the temporal stride τ is now considered as a tuneable parameter. Note that the number of input frames γ_t, width γ_w, bottleneck width γ_b, and depth γ_d remain the same. The original X3D-S architecture can be found in [8].

4. Experiments

To show that intuitive hyper-parameter tuning enables existing models to achieve competitive performance on new tasks, the modified model is first pre-trained on the Kinetics dataset [16]. It is then fine-tuned with various hyper-parameter and data augmentation configurations using a Bayesian based tuning method [14]. Experiments are performed using three widely used benchmark violence detection datasets: RWF-2000 [7], Surveillance Camera Fight Dataset (Surv/SCFD) [2], and ViolentFlows [9].

4.1. Datasets

RWF-2000 [7] is the largest public dataset of surveillance footage for violence detection. The dataset contains $2,000$ videos, half of which are labeled as violent and the others as non-violent. Each clip has a duration of 5 seconds and a frame rate of 30 fps, with varying resolutions. Data leakage prevention is done by assigning all the clips corresponding to a common video to the same partition.

SCFD [2] contains 300 surveillance videos with clip-level labels of fight or non-fight. The duration of each video is 2 seconds, with varying frame rates and resolutions. No public description is available for the training and testing splits. Given the small size of this dataset, we randomly divide the data into 5 folds, where each fold contains a balanced number of fight/non-fight clips. Since clips from the same video have similar features, we assign all the clips corresponding to the same video to the same fold. These folds are available for future researchers upon request. We evaluate our model using k-fold cross-validation. Hyper-parameter optimization is performed for each fold.

ViolentFlows [9] contains 246 videos of crowds with clip-level violent or non-violent labels. All videos in the dataset have 240×320 pixel resolution and range from 1.04 to 6.52 seconds. Again, given the limited amount of clips, we evaluate our model using k-fold cross validation. The folds are separated according to the partitions published in the original paper.

4.2. Experimental Procedure

We use the Bayesian method HyperOpt [14] to determine the optimal hyper-parameter configuration for each dataset. With this algorithm, the first $k_1 = 32$ hyper-parameter combinations are randomly sampled from the search space. Following this warm-up period, the algorithm uses Tree Parzen Estimators [3] to suggest future hyper-parameter combinations. We iterate through an additional $k_2 = 40$ combinations for a total of $k = 72$ hyper-parameter combinations. We apply the following data augmentation strategies: random temporal sampling, resizing within (Section 3.1), brightness jitter, and horizontal flips. Table 3 shows the hyper-parameter search space used in our experiments. Values of 0.5625, 0.75, 1.3333, and 1.7777 for AR'_L and AR'_U correspond to the common video aspect ratios of $9:16$, $3:4$, $4:3$, and $16:9$, respectively. When τ is tuned as an additional hyper-parameter for the RWF-2000 dataset, we use a minimum τ of 9 to ensure that, if violence only occurs for a short duration, it is likely to be included in our sub-sampled frames. During the random hyper-parameter sampling period, learning rate is sampled such that $\log_{10}(LR) \sim U(-7, -2)$. For all other hyper-parameters, values are chosen from a discrete uniform distribution over the specified set.

Table 3. Hyper-parameter search space.

Hyper-parameter	Value Range/Set
Learning Rate (LR)	$[1e-7,\ 1e-2]$
Step (n)	$\{10,\ 15, \ldots 35,\ 40\}$
Dropout Rate (DR)	$\{0,\ 0.1, \ldots 0.6,\ 0.7\}$
Brightness Jitter (BR)	$\{0,\ 0.1, \ldots 0.5,\ 0.6\}$
Scale Bound (S'_L)	$\{0.2,\ 0.25, \ldots 0.95,\ 1.0\}$
AR'_L	$\{0.5625,\ 0.75,\ 1.0\}$
AR'_U	$\{1.0,\ 1.3333,\ 1.7777\}$
τ (RWF-2000 Only)	$\{9, 10, 11\}$

For each hyper-parameter combination, the corresponding model is trained for 60 epochs. We use the Adam optimizer [17] and a multi-step learning rate scheduling policy that reduces the learning rate by a factor of 10 every n epochs. We limit the results shown in this paper to the X3D-XS and X3D-S models, which are computationally lighter than X3D-M and X3D-L while showing comparable performance.

After training models on 72 initial hyper-parameter combinations, we analyze the search space using functional ANOVA (fANOVA) [12]. Specifically, we fit eight random forests to empirically quantify the importance of each hyper-parameter for a given dataset. For each random forest, the p hyper-parameters are ranked from most (1) to least (p) important. Ranks are summed across all eight random forests to identify the three least important hyper-parameters. These hyper-parameters are then eliminated from the search space by fixing them to the values that gave the highest mean validation accuracy during the 72 initial trials. Following the reduction in search space dimensionality, a finer HyperOpt search is performed using another 112 hyper-parameter combinations.

4.3. Implementation

Our work is implemented in PyTorch. We use the HyperOpt [14] algorithm as implemented in the Tune library [20] to determine the optimal hyper-parameters. The models are trained and evaluated on one of Marquette University's high-performance computing cluster nodes, with 8 models simultaneously trained across 8 NVIDIA Tesla V100 GPUs and two Intel Cascade Lake 18-core 2.6 GHz processors. For the X3D-S model with a batch size of 16, the average training and validation time was 229 minutes per model on the RWF-2000 dataset.

5. Benchmark Results

Video level violence classification accuracy for each dataset are shown in Table 4. Bolded values indicate state-of-the-art performance. As hyper-parameter optimization was performed individually for each dataset, further details are given in Subsections 5.1- 5.3.

Table 4. Accuracy of several state-of-the-art violence detection models, as reported by their respective authors. Bolded results are the highest performance for the given dataset.

Method	RWF	SCFD	VF
FightCNN, Bi-LSTM, Attn. [2]	-	72.0	-
ViT-Large-16 [1]	-	76.6	-
X3D Transfer Learning [24]	84.8	-	-
Flow-Gated Net [7]	87.3	-	88.9
VDstr [6]	93.8	-	90.6
SPIL [30]	89.3	-	94.5
ECA-Two Cascade TSM [19]	89.3	-	98.0
SepConvLSTM [13]	89.8	-	-
MSM + T-SE [15]	92.0	**92.0**	98.0
RCNN + Darknet + LSTM [36]	-	74.0	98.2
VGG-16 + ConvLSTM [22]	92.4	-	**98.4**
Modified model	**94.0**	88.7	98.0

5.1. RWF-2000 Results

Our preliminary experiments on RWF-2000 showed better performance when using a single clip for evaluation and higher τ values. A higher τ allows the model to examine a longer portion of the video. Thus, we use single-clip evaluation and analyze the impact of τ on model performance.

After the initial 72 trials with X3D-S, the fANOVA analysis found that learning rate, S'_L, and step had the strongest influence on training and validation accuracy, while AR'_L, AR'_U, and τ had the least importance. After the subsequent 112 trials in the reduced hyper-parameter search space, we found the highest validation accuracy with $AR'_L = 1.0$, $AR'_U = 1.3333$, $BR = 0.3$, $DR = 0.1$, $LR = 1.996 \times 10^{-4}$, $S'_L = 1.0$, $n = 25$, $\tau = 11$. Our model achieves a 94% classification score, which surpasses any previous methods as mentioned in Table 4.

Figure 3 illustrates several predictions of our method on the RWF-2000 dataset. In row (a), the video clip is labeled "violent", and as we can see there is a confrontation between individuals in the scene. However, that dispute occurs for a few frames at the end of the clip causing our method to miss-classify the clip as "non violent". Even when a different clip from the same video is evaluated, if confrontation occurs throughout the scene, such as illustrated in row (b), our model is able to correctly identify that this is indeed a violent video segment. For row (c), although the label is "non violent", our modified model predicts it as violent. This may be caused by the individual in the scene performing a seemingly aggressive action, i.e. "punching the air". Another non-violent clip from the same video but with the same individual staying still leads our model to correctly predict it as non violent.

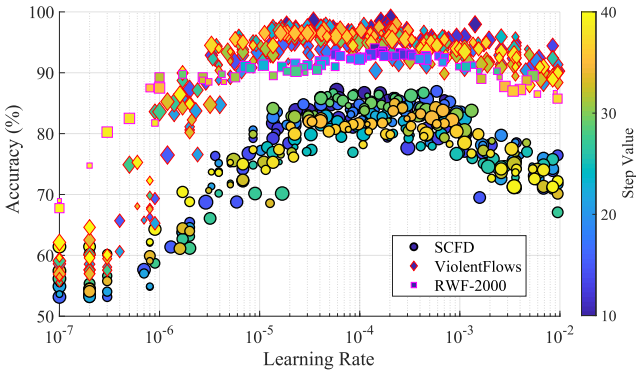


Figure 2. Accuracy as a function of the most relevant hyper-parameters: learning rate, step (color), and S'_L (size).

To further investigate these detection results, we apply a three-dimensional extension of Grad-CAM [25] for visualizing and analyzing our detection results. Grad-CAM is usually employed to analyze features on which the network focuses. This analysis is done using a weighted sum of the activation maps at layer conv_5 from the model in Table 2. The weights are computed using the gradient of each activation at pool_5 with respect to the selected class. The weighted sum activation map is then expanded using interpolation and overlaid frame-wise on the input video following the approach of [29], which extended [40] to three dimensions.

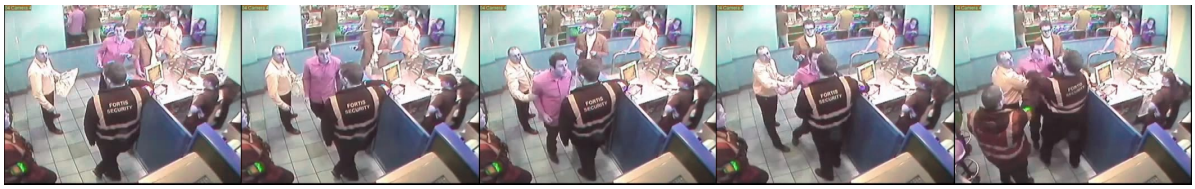
Figures 4 and 5 show the Grad-CAM results related to the clips shown in Figures 3 (a) and (b). In Figure 4, it can be seen that since the confrontation only happens at the end of the video sequence, the network fails to subtract the relevant features (highlighted areas), and this results in the misclassification of this clip. On the other hand, Figure 5 shows when the confrontation is throughout this clip, the network is able to highlight the correct features where violence happens, and in this case, the violence is correctly detected.

5.2. SCFD Results

We use X3D-XS as the model of choice given the small amount of training data. As the shortest video in Surv/SCFD is 20 frames long, we use X3D-XS with $\tau = 5$ and sample the same 4 frames per sub-clip as [8]. The first 72 trials found AR'_L, AR'_U, and n to be the three least important hyper-parameters. We achieved the best performance with $AR'_L = 1.0$, $AR'_U = 1.7777$, $BR = 0.4$, $DR = 0.3$, $LR = 1.77 \times 10^{-4}$, $S'_L = 0.6$, $n = 35$. Our method achieves 88.7%, which is ranked the second best as shown in Table 4.

5.3. ViolentFlows Results

Here we use X3D-XS again but with $\tau = 6$ because the shortest video consists of 24 frames. The 72 initial trials showed that learning rate, BR, and S'_L are the most important hyper-parameters, while AR'_L, AR'_U, and DR were the three least important hyper-parameters. We achieved the best performance with $AR'_L = 1.0$, $AR'_U = 1.7777$, $BR = 0.4$, $DR = 0.5$, $LR = 1.754 \times 10^{-4}$, $S'_L = 0.95$, $n = 10$. As



(a) Label: Violent, Pred: Non Violent

(b) Label: Violent, Pred: Violent

(c) Label: Non Violent, Pred: Violent

(d) Label: Non Violent, Pred: Non Violent

Figure 3. Test set example clips from the RWF-2000 dataset containing combinations of correct and incorrect predictions. Each row refers to sample frames from an individual clip. Examples (a) and (b) are different clips from the same video, likewise with (c) and (d).

for the violence classification accuracy, our method obtains 98%, which is only 0.4% behind the SOTA [22] as shown in Table 4. It is important to note that the high saturation in classification accuracy makes it difficult to significantly increase performance in this dataset.

5.4. Ablation Results

Our results show the varying impacts of hyper-parameters on accuracy depending on the evaluation dataset. Figure 2 shows that learning rates between 10^{-3} and 10^{-5} provide the best performance across all three benchmarks. More importantly, optimal performance on the ViolentFlows and SCFD dataset is achieved when the learning rate is reduced more frequently, as shown by the step parameter. Higher S'_L values provide the highest accuracy, and AR'_L and AR'_U are consistently among the least important hyper-parameters.

6. CONCLUSION

In this work, the X3D model is modified to address the violence classification problem. Compared to the previous implementation on violence detection with the vanilla X3D [24], our method improves upon those accuracy results by nearly 8% on the RWF-2000 dataset. The method also achieves competitive results compared to other methods as shown in Table 4. Specifically, our method achieves 94% accuracy, which represents the SOTA for RWF-2000. In terms of results on SCFD dataset, our method ranked the second best as shown in Table 4. Moreover, results on VF dataset are only 0.4% of accuracy behind the SOTA. Future work will include an in-depth investigation on the influence of spatial cropping, “resize within", and other data augmentation techniques. In addition, further research will investigate the use of systematic hyper-parameter scaling on other architectures and violence detection datasets.

7. Acknowledgement

This work was funded by the U.S. National Science Foundation (NSF) award CNS-1952102 “SCC-PG: Safety and Security of College Campuses and their Adjacent Communities", U.S. Department of Education GAANN P200A180003,

Figure 4. GradCAM results corresponding to Figure 3 (a). Focus on incorrect features results in the miss-classification of this clip.

Figure 5. GradCAM results corresponding to Figure 3 (b). Focus on the region where violence takes place results in a correct prediction.

and partially funded by NSF award CNS-1828649 “MRI: Acquisition of iMARC: High Performance Computing for STEM Research and Education in Southeast Wisconsin"

References

[1] Ş. Aktı, F. Ofli, M. Imran, and H. K. Ekenel. Fight detection from still images in the wild. In *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2022.

[2] Ş. Aktı, G. A. Tataroğlu, and H. K. Ekenel. Vision-based fight detection from surveillance cameras. In *9th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2019.

[3] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36, 2004.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] M. Chelali, C. Kurtz, and N. Vincent. Violence detection from video under 2D spatio-temporal representations. In *IEEE International Conference on Image Processing*, pages 2593–2597, 2021.

[7] M. Cheng, K. Cai, and M. Li. Rwf-2000: An open large scale video database for violence detection. In *25th International Conference on Pattern Recognition*, pages 4183–4190, 2021.

[8] C. Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, 2020.

[9] T. Hassner, Y. Itcher, and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[10] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[11] N. Hussein, E. Gavves, and A. W. Smeulders. Timeception for complex action recognition. In *IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 254–263, 2019.

[12] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *31st International Conference on Machine Learning*, volume 32, pages 754–762, 2014.

[13] Z. Islam, M. Rukonuzzaman, R. Ahmed, M. H. Kabir, and M. Farazi. Efficient two-stream network for violence detection using separable convolutional LSTM. In *International Joint Conference on Neural Networks*, pages 1–8, 2021.

[14] James Bergstra, Dan Yamins, and David D. Cox. Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. In *12th Python in Science Conference*, pages 13 – 19, 2013.

[15] M.-s. Kang, R.-H. Park, and H.-M. Park. Efficient spatio-temporal modeling methods for real-time violence recognition. *IEEE Access*, 2021.

[16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

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

[19] Q. Liang, Y. Li, B. Chen, and Y. Kaikai. Violence behavior recognition of two-cascade temporal shift module with attention mechanism. *Journal of Electronic Imaging*, 30(4):1 – 13, 2021.

[20] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[21] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[22] I. Mugunga, J. Dong, E. Rigall, S. Guo, A. H. Madessa, and H. S. Nawaz. A frame-based feature model for violence detection from surveillance cameras using ConvLSTM network. In *6th International Conference on Image, Vision and Computing (ICIVC)*, pages 55–60, 2021.

[23] H. Nallaivarothayan, C. Fookes, S. Denman, and S. Sridharan. An mrf based abnormal event detection approach using motion and appearance features. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 343–348, 2014.

[24] F. Santos, D. Durães, F. S. Marcondes, S. Lange, J. Machado, and P. Novais. Efficient violence detection using transfer learning. In *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good.*, pages 65–75, 2021.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[26] Z. Shao, J. Cai, and Z. Wang. Smart monitoring cameras driven intelligent processing to big surveillance video data. *IEEE Transactions on Big Data*, 4(1):105–116, 2018.

[27] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[28] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman. A short note on the kinetics-700-2020 human action dataset. *CoRR*, abs/2010.10864, 2020.

[29] A. Stergiou, G. Kapidis, G. Kalliatakis, C. Chrysoulas, R. Veltkamp, and R. Poppe. Saliency tubes: Visual explanations for spatio-temporal convolutions. In *IEEE International Conference on Image Processing*, pages 1830–1834, 2019.

[30] Y. Su, G. Lin, J. Zhu, and Q. Wu. Human interaction learning on 3D skeleton point clouds for video violence recognition. In *European Conference on Computer Vision*, pages 74–90, 2020.

[31] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[32] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

[33] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision*, 2015.

[34] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019.

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

[36] F. U. M. Ullah, M. Obaidat, K. Muhammad, A. Ullah, S. Baik, F. Cuzzolin, J. Rodrigues, and V. Albuquerque. An intelligent system for complex violence pattern analysis and detection. *International Journal of Intelligent Systems*, 07 2021.

[37] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *European Conference on Computer Vision*, pages 322–339, 2020.

[38] H.-J. Ye, D.-W. Zhou, L. Hong, Z. Li, X.-S. Wei, and D.-C. Zhan. Contextualizing multiple tasks via learning to decompose. *arXiv preprint arXiv:2106.08112*, 2021.

[39] J. Yu, H. Chen, K. Wu, Z. Cai, and J. Cui. A distributed storage system for robust, privacy-preserving surveillance cameras. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 1195–1196, 2020.

[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.