Revue d'Intelligence Artificielle
Vol. 37, No. 5, October, 2023, pp. 1221-1228
Journal homepage: http://iieta.org/journals/ria

Dynamic Adaptation of Activation Function to Fine Tune Video ResNet for Fight or Non-Fight Classification

Atif Faridi[1], Farheen Siddiqui[1], Durgesh Nandan[2], Md Tabrez Nafis[1*], Mohd Abdul Ahad[1]

[1] Department of Computer Science & Engineering, Jamia Hamdard, Delhi 110060, India
[2] Department of Electronics & Telecommunication, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India

Corresponding Author Email: tabrez.nafis@gmail.com

(This article is part of the Special Issue **the Impact of AI on Decision-Making**)

https://doi.org/10.18280/ria.370514

Received: 25 June 2023
Revised: 25 July 2023
Accepted: 31 July 2023
Available online: 31 October 2023

Keywords:
fight detection, 3DCNN, Video ResNet, incident detection, fine tuning, 2Plus1D CNN

ABSTRACT

The task of designing and training a 3D convolutional neural network (CNN) from scratch poses significant complexity, necessitating high levels of expertise to achieve a performance that rivals the state-of-the-art. To circumvent this, fine-tuning of neural networks has emerged as a formidable approach. This study focuses on the utilization of Video ResNet, a state-of-the-art architecture known for its proficiency in capturing spatiotemporal patterns from video data. A novel approach is proposed for the fine-tuning of the 3D CNN model (Video ResNet) that involves altering activation functions over epochs while maintaining the network weights and biases consistent. This dynamic approach was assessed under various hyperparameters, yielding encouraging results. Contrary to most studies that employ down-sampling of the temporal sequence to minimize memory requirements, this study introduces a sliding window-based approach to evade down-sampling and prevent potential information loss. The proposed methodology yielded an accuracy of 87.25% in the fight/non-fight classification on the RWF-2000 dataset, marginally surpassing the performance of the state-of-the-art model. The proposed method not only facilitates the development of a real-time video incident detection model but also addresses the issue of overfitting during training through the incorporation of adaptive dynamic activation functions. This study thus contributes to the ongoing advancements in the field of neural network fine-tuning and video data classification.

1. INTRODUCTION

Nowadays, the cost of CCTV video surveillance systems has been reduced significantly due to which it is frequently being deployed at many places like smart cities, hospitals, schools, restaurants, stadiums, shopping malls and theaters etc. The worldwide utilization of CCTV surveillance systems in public and private places has enabled researchers to analyze a huge volume of data to ensure automatic monitoring. In order to maintain a good and peaceful environment around the surveillance area, abnormal activities like fighting must be detected and reported in real time. Automatic detection of fights for rapid actions is very significant and can efficiently assist the concerned departments.

3D convolutional neural networks (3D CNNs) [1] represent a type of neural network that conducts convolutions in three dimensions across the spatial and temporal dimensions of a video volume. This network architecture excels at discerning patterns within spatiotemporal data. Given that videos inherently encompass spatiotemporal characteristics, the application of 3D CNNs proves particularly apt for effectively capturing and analyzing such data.

After Alexnet [2], deep neural network-based learning has provided a great motivation in the field of still-image understanding. With a regular advancement driven by profound design and innovations like spatial filters [3], multi-scale convolutions [4], skip connections [5], residual learning [6], dropout layer [7] etc. has established the deep neural network-based image understanding. Fine tuning a benchmarked deep neural network architecture [8] has given a big boost to the field of computer vision and this technology is helping to solve many real-life problems like face recognition [9], MRI image classification [10], malign cell detection and segmentation [11, 12], object segmentation [11] etc. However, video understanding has not yet taken the Alexnet momentum. The reason is modeling of spatiotemporal data is a big challenge. While some deep neural networks like I3D [13] and Video ResNets [14] match the state-of-the-art results on action recognition datasets like UCF-101 [15], HMDB51 [15], Kinetics and Sports-1M [15].

The temporal sequence length of fight/non fight cannot be determined because of variability and non-deterministic nature of fight duration. In some cases, fights might happen slowly however in some cases it might be fast [16]. So, it is very important to examine various sequence lengths and find out the most appropriate sequence length for an action. In our experimental setup we have analyzed different sequence lengths. In this paper, we will fine tune the pretrained Video ResNet network by dynamically adapting the activation functions after some random epochs during the model fitting for the automatic fight detection on recently published benchmark fight/non-fight dataset RWF-2000 [17] which is somehow different as compared to traditional model training in which activation function remains constant during whole training process, however we will fine tune the temporal sequence length of video frames and learning rate using traditional technique. The rest of the manuscript is organized as follows:

Section 2 discusses the related works, highlighting the state-of-the-art in the domain of violence detection. Section 3 provides the background and the discussion about the various existing machine learning approaches. Section 4 provides the proposed approach and discusses the methodology and algorithm used. Section 5 discusses the result and provides the future scope.

2. RELATED WORKS

A number of open-sourced video dataset has been published for automatic fight/non-fight detection from video like hockey dataset, UCF-101 dataset [15] etc. But none of them have sufficient data for video analysis on fight/non-fight. Cheng et al. [17] has published RWF-2000 (RealWorldFight) that has 1000 fight and 1000 non-fight video clips which are divided into two mutually exclusive train and validation sets to avoid data leakage. Ming Cheng et al. has also proposed a state-of-the-art technique for automatic fight/non-fight detection. They have proposed a five-block model in which two blocks run independently to extract spatiotemporal features. In the first block, there are four 3D convolutional layers connected back-to-back which take an RGB image as input. Second block is similar to the first block except it takes optical flow channels in place of the RGB image. In the third block, RGB spatiotemporal features are fused with optical flow features. The fourth block is a merging block, it merges outputs from the previous block and performs 3D convolution. The last block contains fully connected layers. The accuracy of this method is 87.25% on the validation dataset. Since the number of frames in 5 seconds clip with frames rate 30 is 5×30=150, Cheng et al. [17] have decided to down sample the frame length of 150 to avoid memory and computational overflow. The study [17] have skipped frames at regular intervals to down sample frame sequence of 150 to frame sequence of 64.

Islam et al. [18] have proposed a two-stream neural network for automatic violence detection from video using separable convolutional neural network. They have proposed a separable convolutional LSTM layer which is a reconstruction of ConvLSTM, in this setting each gate of ConvLSTM is replaced by a depthwise separable convolutional layer. In the first stream, background suppressed frames are passed to the CNN layer and SepConvLSTM layer. In the second stream, frame differences are passed to the CNN layer followed by SepConvLSTM layer. These two streams are connected with a fusion block that performs the fusion operation on the two stream features. The CNN layer here is typically a pre-trained truncated MobileNet. In model training, Islam et al. [18] have suggested uniform sampling to reduce 150 frame sequence length to 32 frame sequence length. The sampling of higher temporal sequence to lower temporal sequence length helps in reducing hardware resource requirement for the training as well as inference.

Ullah et al. [19] have proposed an AI assisted IoT based video surveillance system to detect violence for Industrial IoT. The study [19] focused on light weighted model so that the model can be deployed on edge devices. Ullah et al. have proposed object detection-based technique to identify the objects that can be used in violence they call them the suspicious objects and the model detects the human presence in the frame with suspicious objects. If humans and suspicious objects coincide in a frame then an alert is generated and sent to the corresponding alert sensor. The authors [19] have selected YoloV3 as the base model for object detection and a subset of ImageNet dataset is used to fine tune the YoloV3 model on the subset dataset. This model didn't consider the temporal sequence for activity detection. Since the list is too long, we are taking a quick walk through of some recent violence detection techniques.

The authors [19] proposed an AI based violence detection system for the constrained IoT devices. They claimed to improve the accuracy by 3.9% as compared with the existing state-of-the-art detection methods. The authors [20] conducted a literature review of the existing violence detection methods. They primarily focused on the deep sequence learning approaches. The authors [18] proposed a two-stream deep learning architecture using "Separable Convolutional LSTM (SepConvLSTM)" and pre-trained MobileNet for taking background suppressed frames as inputs and processing them afterwards. The authors [21] proposed to improve the accuracy of the existing violence detection approaches by introducing a new feature called "Histogram of Optical flow Magnitude and Orientation (HOMO)". The feature is used to calculate the optical flow between the frames. In the study [22], the authors proposed an approach for violence detection in a real-time scenario of a football stadium. They have used the BiLSTM mechanism for training the model and claimed to achieve an accuracy of 94.5% on hockey dataset. The authors [23] proposed a deep learning approach for detecting abnormal user behavior in the input video streams. They claim to achieve a detection accuracy of more than 95%.

In almost all studies, authors have considered a complete video clip which is of duration 5 seconds as an input to the model.

3. BACKGROUND

This section provides a background of the approaches used in the proposed mechanism. We have used the RWF-2000 dataset [17] of a large video database having 2000 video clips of 5-seconds each at 30 fps. This is a complete balanced dataset containing 1000 videos from fight class and 1000 video for non-fight class. Further, the videos are divided into two sets: the train set and validation set in the ratio of 80-20.

3.1 Activation functions

Activation functions can show complex connections that don't follow a straight line [24]. It can learn not only from tabular data, but also from data about images and speech [24]. For biases and weights in a deep neural network to work in a complex way, they need to have functions. With the activation function, back propagation is now possible because the mistake values from the gradients can be used to change the weights and biases. These functions are monotonic, which means that the model's error surface will always be convex.

3.1.1 Rectified Linear Unit (ReLU)

ReLU is an activation function introduced by He et al. [6], which has strong biological and mathematical underpinning.

From a biological perspective, ReLU's behavior is loosely inspired by the firing behavior of biological neurons. In real neurons, there exists a threshold below which the neuron remains inactive, and beyond which it becomes active and produces an output signal. ReLU mimics this concept by outputting zero for all negative inputs and the input value itself for all positive inputs. This "rectification" process, where negative values are set to zero, reflects the idea that neurons tend to be inactive until a certain level of excitation is reached. Figure 1 displays the plot of the Rectified Linear Unit (ReLU) function.

$$ReLU\left(x\right)=\begin{cases} x, & x\geq 0 \\ 0, otherwise & \end{cases}$$

3.1.2 Leaky ReLU

The Leaky ReLU Activation Function (LeakyReLU) is very similar to the ReLU Activation Function with one change. Instead of sending negative values to zero, a very small slope parameter is used which incorporates some information from negative values. This activation function was first introduced by Maas et al [25]. Figure 1 shows the graph for Leaky ReLU function.

3.1.3 Exponential Linear Unit (ELU)

An ELU activation layer performs the identity operation on positive inputs and an exponential nonlinearity on negative inputs. Figure 2 presents a graphical representation that elucidates the Exponential Linear Unit (ELU) function and its corresponding derivative.

$$ELU\left(x\right)=\begin{cases} x, & x>0 \\ a*(\exp(x)-1), x\leq 0 & \end{cases}$$

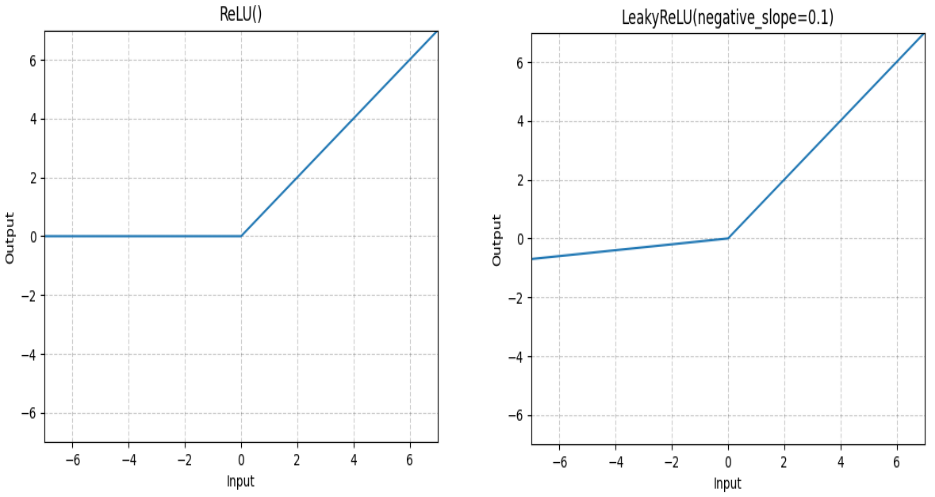

Figure 1. Relu vs Leaky ReLU activation function [26]

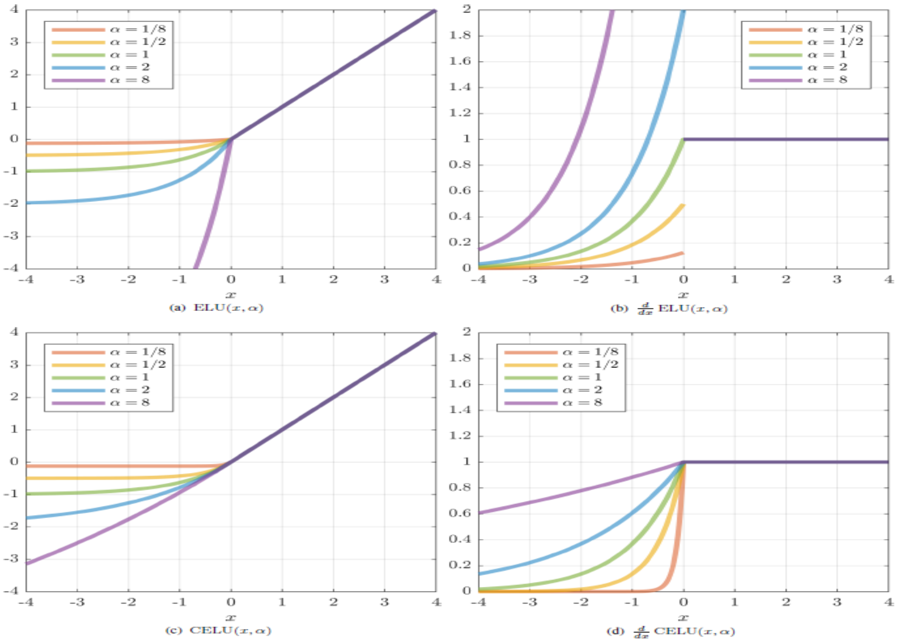

Figure 2. ELU vs CELU activation function [26]

3.1.4 Continuously Differentiable Exponential Unit (CELU)

When alpha is not 1, ELU is not differentiable for x=0. “CELU”, is simply the ELU where the activation for negative values has been modified to ensure that the derivative at x = 0 for all values of α is 1 [27]. Figure 2 presents a visual representation of the CELU function and its derivative, providing a graphical illustration of their characteristics.

$$CELU\left(x\right)=\begin{cases} x, & x>0 \\ a*(\exp(x/a)-1), & x\le 0 \end{cases}$$

3.2 BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) is a type of recurrent neural network architecture that incorporates both forward and backward information flows to capture contextual dependencies in sequential data. BiLSTM networks are well-suited for tasks involving sequences, as they enable the model to consider both past and future information when making predictions. This bidirectional nature enhances the network's ability to understand the context and relationships within the input sequence, making it particularly effective for tasks like natural language processing, speech recognition, and time series analysis. Using Bidirectional Long Short-Term Memory (BiLSTM) networks for processing video data can be challenging due to the multi-dimensional nature of videos, leading to issues in reshaping data, high computational demands, difficulty in capturing long-range dependencies, limited temporal depth, and the complexity of relationships between visual content and temporal aspects. Specialized architectures like 3D CNNs, temporal convolutions, and hybrid models are often preferred for effective video data processing.

3.3 2Plus1D ResNet (Video ResNet)

2Plus1D ResNet [28] is a spatiotemporal 3D convolutional neural network that clearly separates a 3D convolution into a 2D spatial convolution and a 1D temporal convolution that are done one after the other. Figure 3 provides a visual representation of the operation of a 2-plus-1-dimensional convolutional neural network (CNN). Here, the question is: what's the point of breaking things down? First, there are two major benefits: an extra nonlinear correction between these two operations, which essentially doubles the number of nonlinearities compared to a network using full 3D convolutions for the same number of parameters and lets the model learn more complex functions. The other possible benefit is that the decomposition makes it easier to optimize. In practice, this means that both the training loss and the validation loss are smaller.

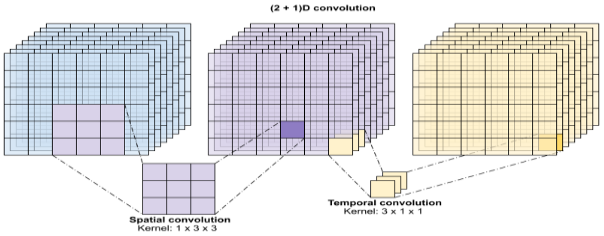

Figure 3. 2Plus1D convolution [29]

3.4 Fine tuning neural network model

Fine-tuning is an example of how transfer learning can be used. Fine-tuning is a process that takes a model that has already been trained for one task and changes it so that it can do a second task that is similar [8, 9]. In the process of fine tuning a previously trained model with accepted level of performance is taken and the one or more layers including classification layer is added. The weights and biases of previously trained models are copied to the new network. Figure 4 shows the transfer learning and fine-tuning process.

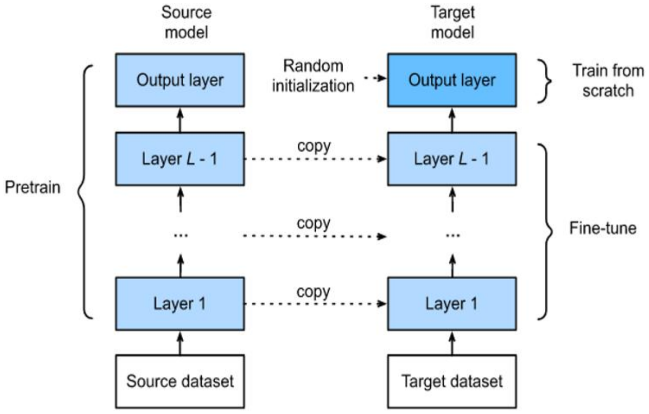

Figure 4. Transfer learning and fine tuning a CNN classifier

4. PROPOSED APPROACH

The core component of an artificial neural network is the neuron, which is represented by a summation function and a filter which is followed by a cutoff function commonly known as activation function. These neurons are arranged into layers through which the information flows, neurons in one layer are connected with neurons in another layer. The decision of firing a neuron is taken by the activation function associated with the neuron. And depending upon how we want to train the neural network model, these activation functions enable neural networks to learn complex decision boundaries. CNNs have had an undeniable effect on computer vision because they can learn high-capacity models from big, annotated training sets. One of the most interesting things about them is that they can move information from a big source dataset to a smaller target dataset. Most of the time, this is done by fine-tuning a fixed-size network based on new goal data. Fine-tuning is a process that occurs after an initial round of training on a base dataset or a pre-trained model [8]. It involves adjusting the model's parameters to improve its performance on a new, related task or dataset. The goal is to leverage the knowledge gained from the base model and transfer it to the new task, thus saving time and resources compared to training a new model from scratch.

Generally, during the fine tuning the architecture of the neural network remains constant during the entire training process. Only the weights and biases are updated during each epoch. The training process is being stopped when we find that there is no more improvement or chance of improvement in validation error or accuracy. In order to achieve further better performance of the model we require to vary different parameters like learning rate, optimizer and inclusion of more layers etc. Till now, we have not found any well-known work in which the activation functions of the layers of the neural network have been changed dynamically during the model training in between epochs.

As we know that activation functions have a great role in the training of the deep neural network. We have a number of activation functions such as sigmoid, tanh, relu, leaky relu, elu, celu, relu6, linear, swish etc. These activation functions can be grouped into different categories based on their mathematical formulation as for example we can group relu, elu, celu, leaky relu and relu6 in one group because all of them have same curve for $x \geq 0$ and minor change for x< 0. Due to the inherent similarity of these activation functions, the substitution of one for another during the runtime of model training will yield minimal alteration to neural network weights and biases. Although minor fluctuations may occur in these weights and biases, they can contribute to stabilizing the training process and mitigating overfitting tendencies. In this research, we meticulously investigate the impact of dynamically adjusting activation functions throughout the training epochs. Specifically, we conduct this analysis on a 3D CNN-driven neural network known as Video ResNet, utilizing the RWF-2000 dataset for detecting instances of fight versus non-fight scenarios.Algorithm given below is the proposed approach.

4.1 Algorithm

```
a. Initialization
   i. Load Dataset
   ii. Load pre-trained model
   iii. Initialize number of epochs and convergence criteria
   iv. Select a set of hyper-parameters
   v. Initialize number of epochs
   vi. Select convergence criteria
b. Add classification head to the pre-trained model
c. Make a pool of activation functions
d. Locate activation layers in the pre-trained model
e. Training process
   i. Prepare sliding window-based training data
   ii. Select sequence length (4/8/12)
   iii. For epoch 1 to N do
   I. Do model training
   II. After some epoch if validation loss increases change the activation function
   iv. Stop if convergence criteria are satisfied
```

4.2 Experimental setup

In this section, we are going to explain the experimental setup.

4.2.1 Dataset

We have used the RWF-2000 benchmark dataset. The details of the RWF-2000 dataset have been described in the related work section.

4.2.2 Network architecture

We have used a standard pre-trained 2Plus1D Video ResNet model from torchvision library and included a binary classification layer at the head of the pre-trained model.

4.2.3 Activation functions

We have used a pool of activation functions which have been assigned to the different activation layers dynamically and randomly during the training of the network. Following activation functions are used in this experimental setup.

a.) ReLU,
b.) Leaky ReLU,
c.) CELU, and
d.) ReLU6

4.2.4 Hyper-parameters

Temporal Sequence Lengths	Activation Functions
4, 6, 8, 10, 12	ReLU, Leaky ReLU, ReLU6, CELU

4.2.5 Details of infrastructure resources used

i. CPU: Intel(R) Xeon(R) CPU @ 2.00GHz
ii. OS: Ubuntu 20.04.5 LTS
iii. RAM: 16 GB
iv. GPU: Tesla T4 16 GB
v. Python Version: 3.8.10
vi. Libraries: Pytorch, Scikit-learn, Numpy, Pandas,
vii. Cuda Version: V11.2.152, This resource is accessible through the free tier of Google Colab, granting users the opportunity to utilize a GPU for their computations.

5. RESULTS AND EVALUATION IMPACT ON FINE-TUNING PROCESS

In this section, we are going to explain the experimental observation. Since, the generalization and performance of the model are evaluated on the validation dataset. Hence, we will give our primary attention towards validation error and accuracy. Table 1 presents the evolution of validation loss and accuracy over a transitional period across various sequence lengths. Table 2 displays the best achieved validation error and validation accuracy attained during the training process. Table 3 showcases a performance comparison between the proposed method and the current state-of-the-art approach on validation data.

Table 1. Transition table showing validation loss and validation accuracy change during transition

Sequence Length	Activation Function Transition	Validation Loss Transition	Validation Accuracy Transition
4	CELU→Leaky Relu	0.773→0.475	71%→78.5%
	Leaky ReLU→Relu	0.78→0.601	79.25%→82%
	ReLU→Relu6	0.911→0.605	81%→84.5%
8	CELU→Leaky Relu	0.794→0.68	73%→79.25%
	Leaky ReLU→Relu	1.027→0.936	79.25%→82.5%
	ReLU→Relu6	0.931→0.627	80.25%→85.5%
12	CELU→Leaky Relu	0.726→0.612	78.25%→80.75%
	Leaky ReLU→Relu	0.843→0.497	81.25%→84.25%
	ReLU→Relu6	0.514→0.418	83.5%→85.5%

Table 2. Best validation error and validation accuracy achieved during training

Sequence Length	Activation Function	Validation Loss	Validation Accuracy
4	CELU	0.5476	76.75%
	Leaky ReLU	0.7093	81%
	ReLU	0.5552	84.25%
	Relu6	0.9475	82.25%
8	CELU	0.8745	80.25%
	Leaky ReLU	0.6522	83.75%
	ReLU	0.5044	85.5%
	Relu6	0.8255	84.75%
12	CELU	0.5258	82.5%
	Leaky ReLU	0.4873	85.5%
	ReLU	0.3945	87.75%
	Relu6	0.6235	86.25%

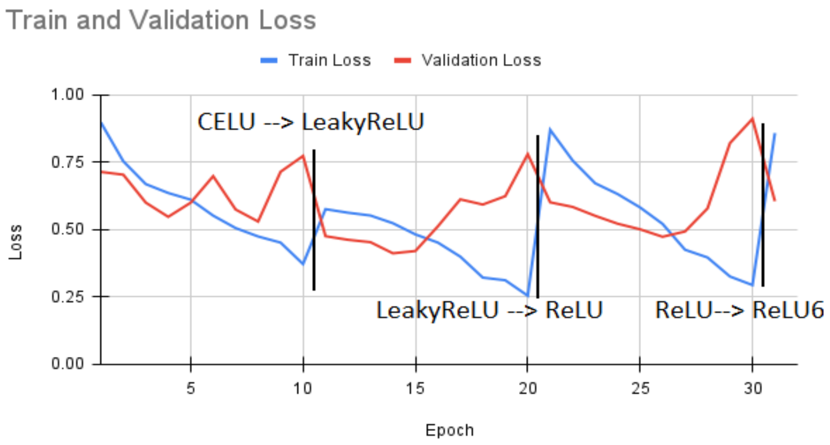

Figure 5. For sequence length 4, decrease in validation loss during transition from one activation to another activation

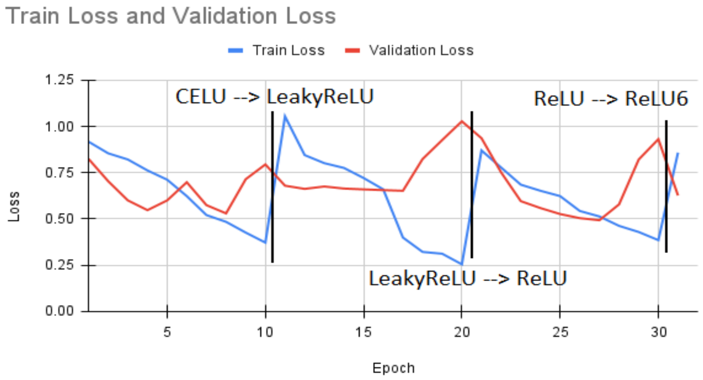

Figure 6. For sequence length 8, decrease in validation loss during transition from one activation to another activation

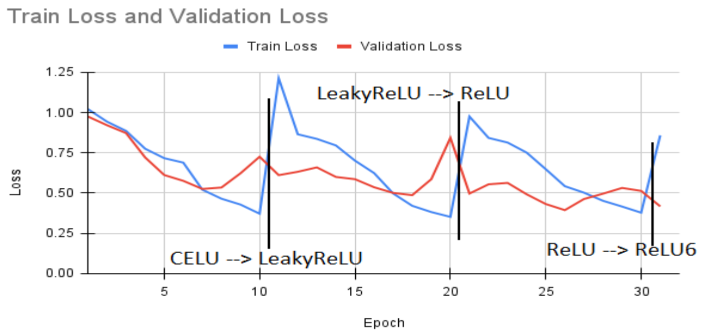

Figure 7. For sequence length 12, decrease in validation loss during transition from one activation to another activation

Table 3. Performance comparison between state-of-the-art and proposed method on validation data

Method	Validation Accuracy
State-of-the-art	87.25%
Ours	87.75%

When analyzing the experimental results, we have noticed following observations:

i. Upon careful analysis of Figure 5, Figure 6, and Figure 7, a notable observation emerges: there is a substantial rise in training loss when the activation function is altered between the two epochs, as opposed to maintaining a consistent activation function throughout.

ii. Another interesting observation can be made from Figure 5, Figure 6, and Figure 7. When the activation function is changed to a similar alternative (with a slight mathematical variation, like moving from ReLU to Leaky ReLU) between two epochs, there is a significant decrease in the validation loss. This reduction is especially noticeable when the model training shows signs of overfitting. This finding suggests that instead of retraining an overfitted model from the beginning, a straightforward adjustment of the activation function and training for a few epochs can save time and effectively improve the model's regularization.

iii. An interesting observation can be made from Figure 5, Figure 6, and Figure 7. If the consecutive activation functions have a significantly different mathematical intuition, changing the activation function between epochs will lead to an increase in both training and validation error.

iv. Temporal sequence length has a significant impact in the performance of the Video Classification. More the sequence length the higher the accuracy and much higher the memory requirements. However, we have not experimented with long sequences (more than 12) due to memory constraint.

v. Through a comprehensive examination of Table 4, it becomes evident that the model consistently exhibits a harmonious and balanced performance across sequence lengths of 4, 8, and 12. This observation is reinforced by the consistent and equitable distribution of true positives, false positives, true negatives, and false negatives.

vi. Furthermore, an observation derived from Table 4 is that the performance of the model improves as the sequence length increases.

Table 4. Confusion Matrix of the models with 4, 8 and 12 sequence length

	Sequence Length					
	4		8		12	
	Fight	**No Fight**	**Fight**	**No Fight**	**Fight**	**No Fight**
Fight	838	162	844	156	887	113
No Fight	153	847	136	864	132	868

6. CONCLUSIONS

We have proposed a very simple, effective, and clean method to fine-tune the 3d (2Plus1d) CNN-based video ResNet to solve spatiotemporal video classification problems. we demonstrated that our end-to-end model fine tuning approach makes transfer learning easier. the concept of replacing activation function dynamically during training epoch of neural network models can be used to regularize an overfitted model without retraining. in other words, the proposed technique can make a rotten model into a fresh model with a little bit of effort. the proposed method attempted to increase learning capability of a neural network model. we believe that our proposed method will open doors in video understanding.

We can make a conclusion stating that training a Video ResNet model on the fight/non-fight dataset RWF-2000 dataset is a promising approach for video classification tasks. The 3D CNN architecture factored into 2Plus1D architecture is a well-established deep learning neural network that has been shown to be effective in video or spatiotemporal classification tasks, and the fight/non-fight RWF-2000 dataset provides a challenging and diverse set of videos to validate the performance of the model. At last, the dynamic adaptation of the activation function improves the fine-tuning process.

REFERENCES

[1] Maturana, D., Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 922-928. https://doi.org/10.1109/IROS.2015.7353481

[2] Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Esesn, B.C.V., Awwal, A.A.S., Asari, V.K. (2018). The history began from AlexNet: A comprehensive survey on deep learning approaches. ArXiv Preprint ArXiv: 1803.01164. https://doi.org/10.48550/arXiv.1803.01164

[3] Griffith, D.A. (2003). Spatial filtering. Springer Berlin Heidelberg.

[4] Cui, Z., Chen, W., Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv: 1603.06995. https://doi.org/10.48550/arXiv.1603.06995

[5] Wu, D., Wang, Y., Xia, S.T., Bailey, J., Ma, X. (2020). Skip connections matter: On the transferability of adversarial examples generated with resnets. ArXiv Preprint ArXiv: 2002.05990. https://doi.org/10.48550/arXiv.2002.05990

[6] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[7] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1): 1929-1958.

[8] Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Transactions

on Medical Imaging, 35(5): 1299-1312. https://doi.org/10.1109/TMI.2016.2535302

[9] Liu, X., Song, H., Ma, H., Fan, Y., Yang, Y. (2016). Transferring deep representation for NIR-VIS heterogeneous face recognition. In 2016 International Conference on Biometrics (ICB), Sweden, pp. 1-7. IEEE. https://doi.org/10.1109/ICB.2016.7550064

[10] Margeta, J., Crainic, K., Gigi, A., Katic, D., Cepanec, M., Loncaric, S. (2017). Fine-tuned convolutional neural nets for cardiac MRI acquisition plane recognition. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 5(5): 339-349. https://doi.org/10.1080/21681163.2015.1061448

[11] Jana, E., Subban, R., Saraswathi, S.S. (2017). Research on skin cancer cell detection using image processing. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), India. https://doi.org/10.1109/ICCIC.2017.8524554

[12] Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V. (2020). U-Net and its variants for medical image segmentation: Theory and applications. IEEE Access, 9: 82031-82057. https://doi.org/10.1109/ACCESS.2021.3086020

[13] Carreira, J., Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.48550/arXiv.1705.07750

[14] Dubey, S., Boragule, A., Jeon, M. (2019). 3D resnet with ranking loss function for abnormal activity detection in videos. 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), China. https://doi.org/10.1109/ICCAIS46528.2019.9074586

[15] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A. (2011). Sequential deep learning for human action recognition. Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings 2. https://doi.org/10.1007/978-3-642-25446-8_4

[16] Schindler, K., Van Gool, L. (2008). Action snippets: How many frames does human action recognition require? 2008 IEEE Conference on Computer Vision and Pattern Recognition, USA. https://doi.org/10.1109/CVPR.2008.4587730

[17] Cheng, M., Cai, K., Li, M. (2021). RWF-2000: An open large scale video database for violence detection. 2020 25th International Conference on Pattern Recognition (ICPR). https://doi.org/10.48550/arXiv.1911.05913

[18] Islam, Z., Rukonuzzaman, S., Ahmed, R., Kabir, M.H., Farazi, M. (2021). Efficient two-stream network for violence detection using separable convolutional LSTM. 2021 International Joint Conference on Neural Networks (IJCNN), China, pp. 1-6. https://doi.org/10.1109/IJCNN52387.2021.9534280

[19] Ullah, F.U.M., Muhammad, K., Haq, I.U., Khan, N., Heidari, A.A., Baik, S.W., de Albuquerque, V.H.C. (2021). AI-assisted edge vision for violence detection in IoT-based industrial surveillance networks. IEEE Transactions on Industrial Informatics, 18(8): 5359–5370. https://doi.org/10.1109/TII.2021.3116377

[20] Mumtaz, N., Ejaz, N., Habib, S., Mohsin, S.M., Tiwari, P., Band, S.S., Kumar, N. (2022). An overview of violence detection techniques: Current challenges and future directions. Artificial Intelligence Review, 56: 1-26. https://doi.org/10.1007/s10462-022-10285-3

[21] Mahmoodi, J., Salajeghe, A. (2019). A classification method based on optical flow for violence detection. Expert Systems with Applications, 127: 121-127. https://doi.org/10.1016/j.eswa.2019.02.032

[22] Halder, R., Chatterjee, R. (2020). CNN-BiLSTM model for violence detection in smart surveillance. SN Computer Science, 1(4), 201. https://doi.org/10.1007/s42979-020-00207-x

[23] Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. ArXiv Preprint ArXiv: 1811.03378. https://doi.org/10.48550/arXiv.1811.03378

[24] Mishra, P. (2022). Introduction to neural networks using PyTorch. In PyTorch Recipes: A Problem-Solution Approach to Build, Train and Deploy Neural Network Models, pp. 117-133. Berkeley, CA: Apress.

[25] Maas, A.L., Hannun, A.Y., Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[26] Barron, J.T. (2017). Continuously differentiable exponential linear units. ArXiv Preprint ArXiv: 1704.07483. https://doi.org/10.48550/arXiv.1704.07483

[27] Tensorflow. (n.d.). Video classification. https://www.tensorflow.org/tutorials/video/video_classification

[28] Xie, S., Zhang, X., Cai, J. (2019). Video crowd detection and abnormal behavior model detection based on machine learning method. Neural Computing and Applications, 31(1): 175-184. https://doi.org/10.1007/s00521-018-3692-x

[29] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, USA. https://doi.org/10.1109/CVPR.2018.00675