Deep Learning for Computer Vision Final Project: Real-Time Violence Detection

Student: Ngan Bao, Columbia University

Professor: Peter Belhumeur

December 9, 2022

1 Introduction

Nowadays videos are widely posted and used on many platforms, social media, and surveillance. Social media platforms such as Instagram, Tiktok, and Youtube have continuously made attempt to censor videos and photos with violent elements to prevent and protect the community against these materials that are highly triggering, distressing and potentially could lead to attacks and/or be causal factors for other crimes and problems. Previously, these videos/materials/surveillance are monitored by real people and it is considered a traumatizing task (Ohlheiser (2017)). Fortunately, the current advancement of AI models allow us to train a network to detect Violent components in an image and/or video. With these models we can look at more videos in a much shorter time and more efficiently, serving as the screening stage to filter out videos with a high probability of containing violence. In this project, I want to explore various network implementations to train a model to detect Violence in a video in real-time.

Github Repo link: PASTE LINK TO REPO HERE.

2 Related Work

There have been various groundbreaking research tackling Violence detection in videos. In a research published in 2020, Febin et al, ”Violence detection in videos for an intelligent surveillance system using MoBSIFT and movement filtering algorithm”. proposed a cascaded method of violence detection based on motion boundary SIFT (MoBSIFT) and movement filtering, in which the surveillance videos are checked through a movement filtering algorithm based on temporal derivative and avoid most of the nonviolent actions from going through feature extraction. Their experimental results show that the proposed MoBSIFT outperforms the existing methods in accuracy by its high tolerance to camera movements. (Febin, Jayasree, and Joy (2020)). Another notable work tackles this problem is published in IEEE by Abdali et al. in 2019, ”Robust Real-Time Violence Detection in Video Using CNN And LSTM”. Their model consists of CNN as a spatial feature extractor and LSTM as temporal relation learning method with a focus on the three-factor (overall generality - accuracy - fast response time). and achieved 98% accuracy with a speed of 131 frames/sec. (Abdali and Al-Tuma (2019))

3 Dataset Description

While most of other published papers on Violence/NonViolence Detection uses the UCF101 - Action Recognition Data Set (https://www.crcv.ucf.edu/data/UCF101.php) for training, I use Kaggle’s Real Life Violence Situations Dataset (https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset?resource=download), the idea is to get the model to learn specific patterns of violence in real life situation.

After downloading, for the implementation of Transformers, although the structure is a bit different since I tried with this method first, although it also generates individual frames, it does not save them to separate folders but instead, to NumPy arrays, which I then save as x_test.npy, y_test.npy, x_train.npy, y_train.npy, x_val.npy, y_val.npy. I separate the original dataset of 1000 videos of each category to 800 each for the train set, 100 each for the test set, and 100 each for the validation set. The structure is as below:

Real_Life_Violence_Dataset

- test_split
- train_split
- val_split

I convert each video into individual frames, I also take only 25 frames of each video, and pad the shorter videos to this length as well. Then I resize all frames to 128x128 before feeding it to the network due to the ability of my device. This idea caused me a lot of problems since the data processing took a lot of time every time I run the training (it took almost 2 days just to process the frames), which led to the creation of the below structure instead. The below structure I use for implementations of ResNet18, and Vgg16 implementations, also has 800 of each category for the train set, 100 each for the test set, and 100 each for the validation set, videos are converted into individual frames, I took all frames in this case, but also I resize all frames to 128x128.

Real_Life_Violence_Dataset

- test_: NV_, V_
- train_: NV_,V_
- val_: NV_, V_

4 Methods

The method is very straightforward, the idea for VGG16 and ResNet18 is that we feed the pre-trained models with frames extracted from videos, labeled as either Violence or NonViolence. Although other paper suggests methods to train models as a sequence of frames, I wanted to see if treating as individual frames, can the model detect images with an actual element of violence, even when it might have a few frames in the video that is nonviolent.

In ResNet implementation, we have over 11 million trainable parameters (when images are 224x224), the input image feeds into each layer and each layer feeds into the next layer and directly into the layers which is 2 strides away, called identity connections. These identity connections, placed between every two Conv layers, take the input directly to the end of each residual block.

In VGG implementation, it takes exponentional more time for the training to complete since this network has total of over 138 million parameters (when images are 224x224). With each set of a Conv layer, the number of filters doubles, and with each pooling layer, the width and height of the image are reduced by half. I used Keras ImageDataGenerator to make adjustments to the training set, and artificially add transformed images to the data frame. Throughout the architecture, convolution and max pool layers are consistently arranged, convolution layers of 3x3 filter with stride 1 and always used the same padding and max pool layer of 2x2 filter of stride 2, more details on the architecture can be found under Section 5.2 VGG16.

In the Transformer implementation (Vaswani et al. (2017)), the general idea is that with this method, the classification describes the entire video with correct classification since the video might be just a street, which is non-violent, but other frames might have a shooting, which should be labeled ”violence”. This proposed network is based solely on attention mechanisms, dispensing with recurrence and convolutions. , the first sequence based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention (Vaswani et al. (2017)).

5 Experiments

In this project, I used my own implementation of torchvision’s ResNet-18, VGG16 pre-trained models, fine-tuned for this specific Violence-Non Violence training dataset and Keras Transformer-based model proposed by (Vaswani et al. (2017)) to classify videos.

5.1. ResNet18

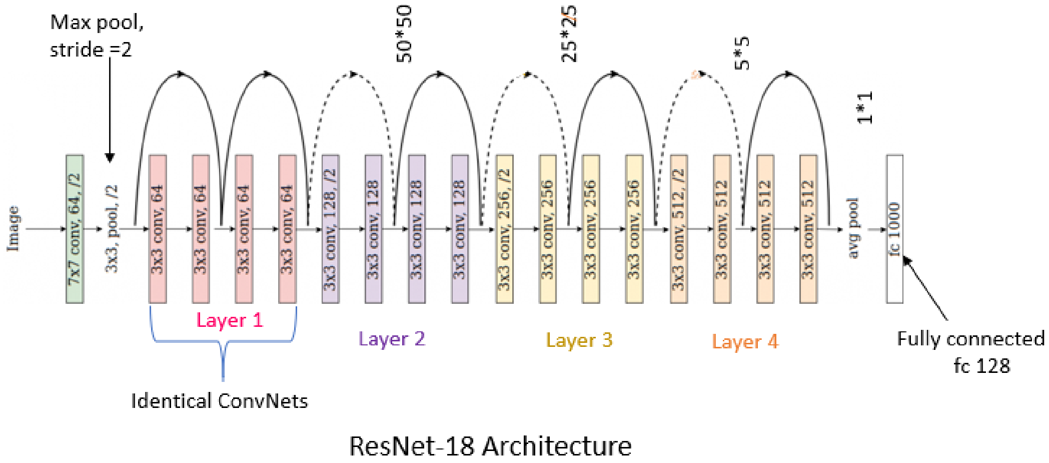


Figure 1: ResNet-18 dataflow.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 2: ResNet-18 Architecture.

Figure 1 is the data flow of the image through the network model, and figure 2 describes the details of the network with input and output sizes. I replaced the last layer with a fully connected layer followed by a sigmoid activation function instead of softmax as shown in Figure 2 (Napoletano, Piccoli, and Schettini (2018)), the output classifying if the frame contains violence feature.

The achieved accuracy is at 95%, where loss is 0.43.

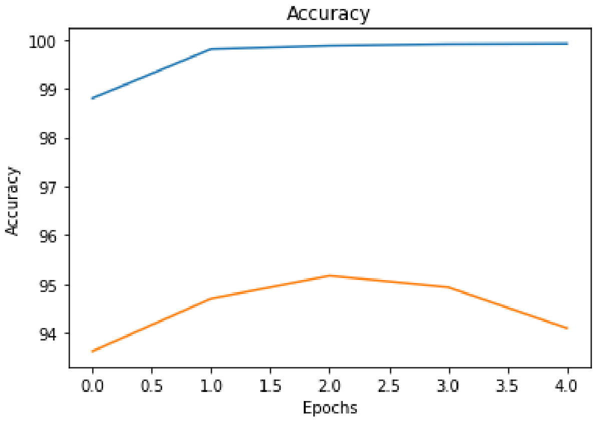
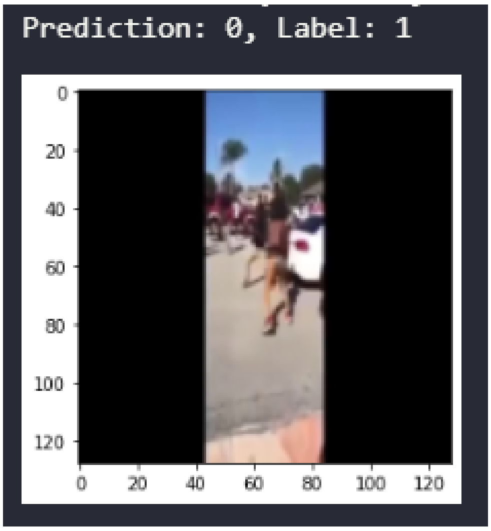


Figure 3: ResNet-18 Accuracy over 5 epochs.

In this implementation, I use Binary Cross Entropy as my function to compute loss, with Adam optimizer, 5 epochs, and a learning rate of 1e-5. Below is a few results when I fed the model a few random frames to test the model:



```
Prediction: 0, Label: 1
```

Figure 4: ResNet-18 Test Prediction, Classified as NonViolent (0), when it's Violent (1)

5.2 VGG16

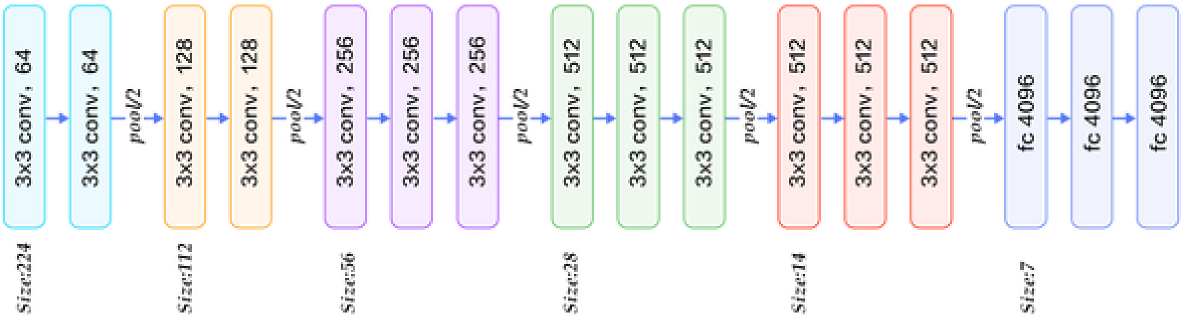
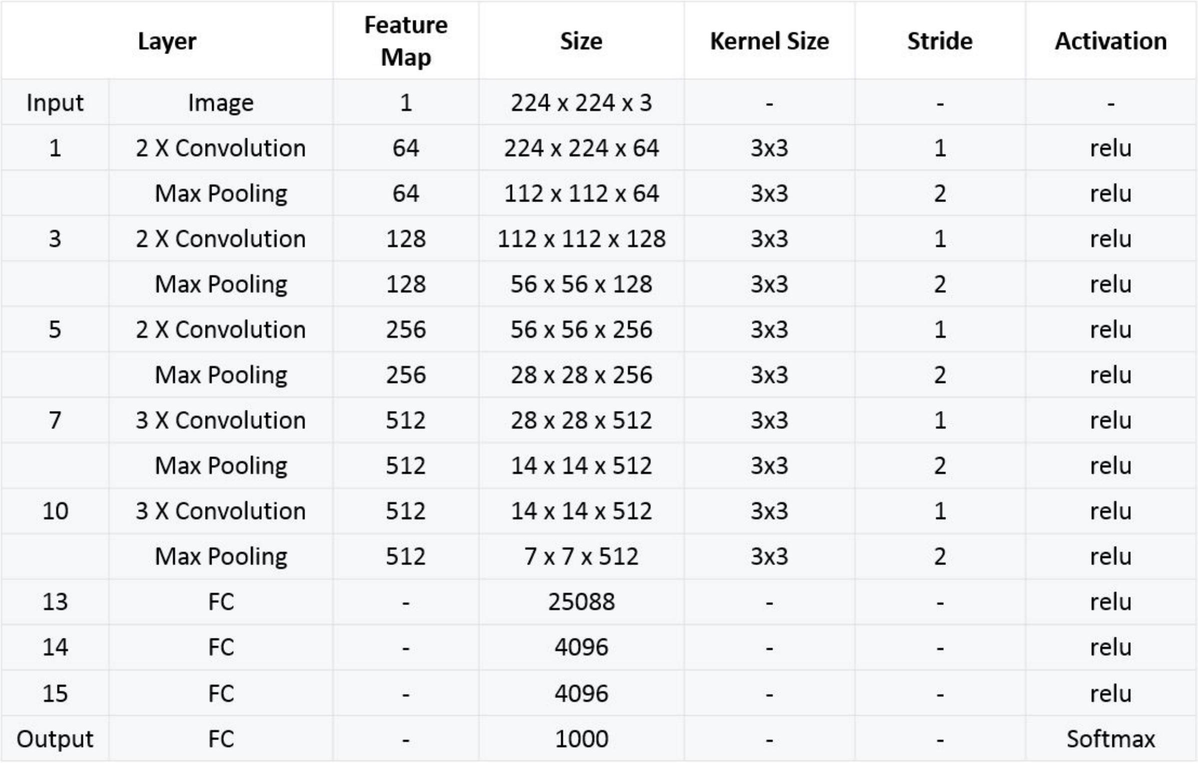


Figure 5: VGG16 dataflow (Google).



Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

Figure 6: VGG16 Architecture (Google).

While the traditional VGG16 input accepts an image size of 224x224, I resize my frames to 128x128 dues to the limitations of the device I have. I changed the last layer output features to be 2 since the original pre-trained network has 1000 outputs, and for our case, I only want the Violent/Nonviolence classification. In this implementation, I use Tensorflow Keras pre-trained model and apply new layers for the classification. Same as for resNet18, I also use ReLU as activation function, loss function to be categorical_crossentropy since we have 2 classes but only want to have one single true class for each image, with Adam optimizer, and a softmax layer, using 3 epochs, and the learning rate is also 1e-5. The accuracy achieved is 93%

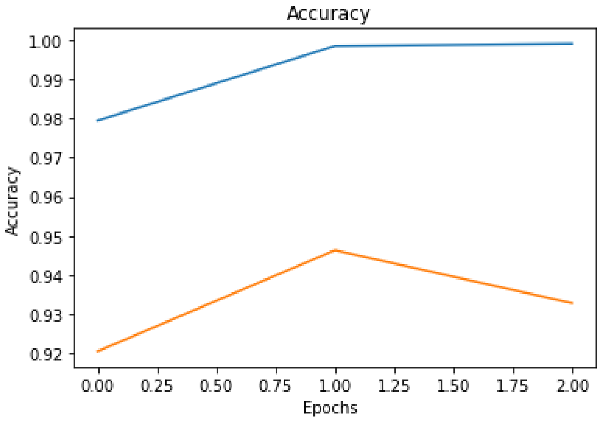


Figure 7: VGG16 Accuracy over 3 epochs.

Below is a sample output result.

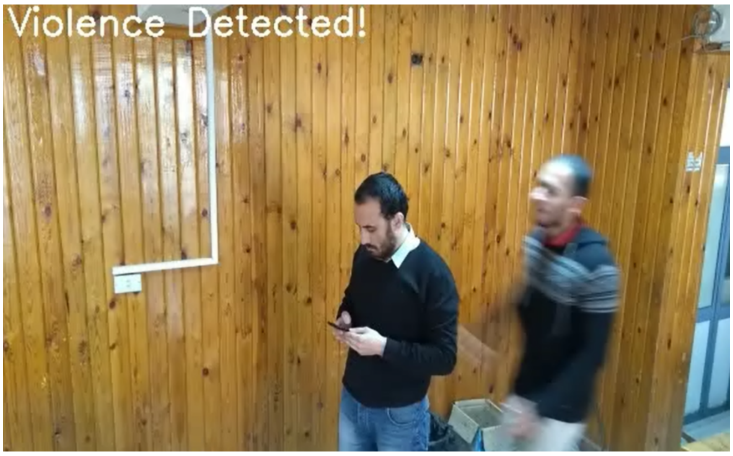


Figure 8: VGG16 Test Prediction. Correctly classified

5.3 Transformers

I also explored following Keras's tutorial for Video Classification with Transformers with 10 epochs, we achieved a test accuracy to be 89.37

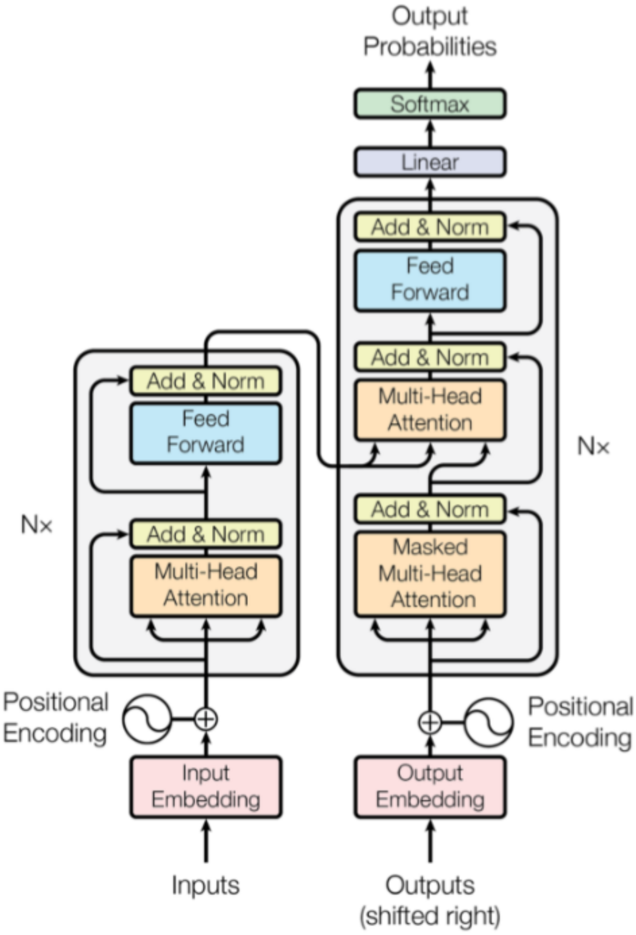


Figure 9: Transformer Architecture (Vaswani et al. (2017)).

Below is the result when I fed the network a test video, the video is NonViolent, even when the model gives relatively high probability that the video is violent (38%), we still achive higher percentage for NonViolence and correctly classify it as NonViolent.

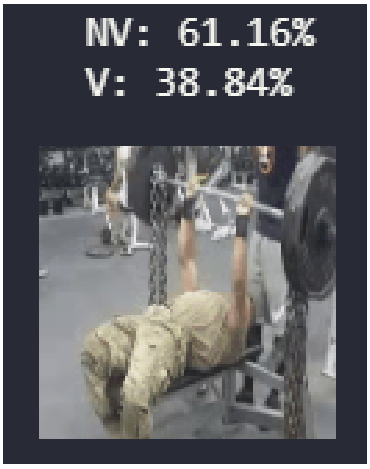


Figure 10: Transformer Test Prediction. Correctly classified

6 Conclusion

From exploring different implementation, I found that the training is done the fastest with Transformer model since I cut down the frames read to 20, it is much less input to train, whe model finished training and testing winthin less than two hour. However, the achieved accuracy test is only 89%, much less than VGG16 and ResNet18 implementations, which took 9 hours and 13 hours to finish training, respectively. However, with ResNet18, we did achieve 95% accuracy and 93% accuracy for VGG16, which is considerably high for pre-trained networks and implementation without many modifications.

References

Abdali, A.-M. R., & Al-Tuma, R. F. (2019). Robust real-time violence detection in video using cnn and lstm. In *2019 2nd scientific conference of computer sciences (sccs)* (p. 104-108). DOI: 10.1109/SCCS.2019.8852616

Febin, I. P., Jayasree, K., & Joy, P. T. (2020). Violence detection in videos for an intelligent surveillance system using mobsift and movement filtering algorithm. SpringerLink. DOI: https://doi.org/10.1007/s10044-019-00821-3

Napoletano, P., Piccoli, F., & Schettini, R. (2018, 01). Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors (Basel, Switzerland)*, *18*. DOI: 10.3390/s18010209

Ohlheiser, A. (2017). The work of monitoring violence online can cause real trauma. and facebook is hiring. Retrieved from https://www.washingtonpost.com/news/the-intersect/wp/2017/05/04/the-work-of-monitoring-violence-online-can-cause-real-trauma-and-facebook-is-hiring/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. Retrieved from http://arxiv.org/abs/1706.03762