Violence Detection in Movies: A Multimodal Approach

Konstantinos Chaldaiopoulos | mtn2220
University of Piraeus & NCSR Demokritos
kostaschaldaio@gmail.com

Natalia Koliou | mtn2205
University of Piraeus & NCSR Demokritos
natalykoliou@yahoo.gr

Abstract

With the rapid growth of video content, the development of efficient ways to detect violence in videos has become increasingly crucial to ensure effective regulation. We aim to study various techniques for violence detection in videos, with special interest in multimodal violence detection frameworks that combine visual, audio, and text features. We explore both early and late fusion techniques to integrate insights derived from each modality. In this paper, we present a report on these findings for both single and multimodal approaches. The different modalities were tested in a new movie dataset composed of 260 violent and non-violent 8-second videos which we handcrafted from public domain movies.

We examine the problems associated with different detection techniques, and the approaches we adopted to navigate them. Furthermore, we offer a comparative analysis between single and multimodal detection methods, providing insights into the implications of integrating multiple data types in violence detection.

I. Introduction

Detecting violence in videos is of significant importance. Accurate violent detection can protect an audience from harmful content. However, the task is far from easy. There are many complexities for this task and different techniques present different challenges.

Our study aims to address these challenges and explore different techniques focusing on the challenges presented by frameworks that combine visual, audio and text features.

We begin with an exploration of single modalities. For the visual modality, we employ a deep learning framework, fusing two different convolutional networks (CNN), AlexNet and SqueezeNet, combining them with Convolution Long Short Term Memory (ConvLSTM) [4] for higher accuracy. The architecture we employ was first presented in the paper titled: Detecting Violence in Video Based on Deep Features Fusion Technique [1]. For the audio modality, we use features extracted from the PyAudioAnalysis [2] library which provides a set of audio analysis tools focusing on feature extraction, classification, segmentateon and visualization.

Finally, for the text modality, we use DistilBERT [3]. DistilBERT is a method of pre-training language representations, just like BERT, but it is a distilled version, meaning it has a smaller architecture with fewer parameters. Despite its smaller size, DistilBERT retains much of BERT's expressive power, making it computationally more efficient while still providing useful contextualized word embeddings for natural language tasks.

Through the exploration of the above modalities , single and fused, we aim to understand the complexities and challenges associated with violence detection in videos.

II. Models

✤ Image violence detection:

For image violence detection we use a deep learning model that combines the pre-trained models AlexNet [5] and SqueezeNet [6] with a convolutional LSTM. The process in detail is presented below:

Extracting the frames: The first step is to extract key frames from the video. This is done using a method that selects frames based on the amount of motion they contain. The idea is to capture the most significant moments in the video, which are likely to contain the most information about whether the video is violent or not. This reduces the amount of data that needs to be processed, making the model more efficient. In our model, the video is split into 20 parts and 20 key frames are extracted with the highest difference. After this the frames are inserted into the model.

Image Classification Model: The first part of the model involves feature extraction, utilizing both AlexNet and SqueezeNet. Both networks, which have been pre-trained on ImageNet, are employed to extract deep features from each frame of the video. AlexNet is a CNN with five convolutional layers and three fully connected layers. On the other hand, SqueezeNet is a simpler network that requires fewer parameters but still manages to deliver comparable performance.

After the feature extraction phase, the features from both networks are passed into a ConvLSTM. The ConvLSTM is a type of LSTM that has been adapted to handle spatial data. It includes convolutions within the LSTM cell, enabling it to capture both spatial and temporal dependencies in the video data. In our case, the ConvLSTM takes a sequence of frames as input. Each frame is a 2D grid of pixels, and the ConvLSTM can capture the spatial relationships between these pixels over the sequence of frames, thus capturing 3D information. This makes it a fitting choice for our task, where both the temporal sequence of frames and the spatial layout of pixels within each frame are important.

Once the ConvLSTM has processed the features from both networks, the features are fused together. This fusion happens by concatenating the features from the final output of the ConvLSTM that processed the features from both networks. The fu-

sion of features from different networks helps to capture a wider range of features and boosts the robustness of the model.

Finally, the output of the ConvLSTM is passed through a softmax layer for classification. The softmax layer outputs the probabilities of the video being violent or non-violent.

Training Details:

Parameter	Value
Learning rate	0.0001
Epochs	20
Batch size	8
Loss function	Cross entropy loss
Optimizer	RMSProp
Weight decay	0.05
Train data length	400 videos

The frames are extracted from 200 videos but due to horizontal flipping for training data augmentation we end up with 400 training instances.

✤ Audio violence detection:

For audio violence detection, we use a simple SVM (Support Vector Machine) classifier, which was fine-tuned for our specific task. The process in detail is presented below:

Extracting the audios: To obtain the audio content from the videos of our movie-dataset, we use the MoviePy library. Once the audio is extracted, we proceed to compute relevant audio features using the pyAudioAnalysis library [2].

We opt to extract midterm features, as they offer a more comprehensive representation of audio characteristics compared to short features. To achieve this, we first define the sampling rate (fs) and set a midterm window size of $1 \cdot fs$. The midterm step is then calculated using the formula: $1 \cdot fs \cdot (1 - overlap=0.5)$, which gives an effective step of $0.5 \cdot fs$ for the midterm feature extraction. This ensures that successive audio segments overlap by 50%, capturing relevant information for our analysis. Additionally, we employ a short window of $0.05 \cdot fs$ and obtained the corresponding short step by using the same formula, resulting in a short step of: $0.05 \cdot fs \cdot (1 - 0.5)$, which provides an overlap of 50% for the short feature extraction.

The phenomenon of having 50% overlap for both short-term and midterm feature extraction means that the real overlap for the midterm features will be higher since each midterm segment comprises overlapping short segments, leading to a more substantial overlap across the duration of the midterm features.

As for the metric to represent the extracted midterm features array, we choose to use the standard deviation (std) because it plays a crucial role in our classification task of distinguishing between violence and non-violence videos. By considering the variability and dispersion of the audio characteristics, the std metric effectively captures how the sound varies within an audio clip. This variability is particularly relevant for our analysis, as it holds significant information regarding the intensity and fluctuations of audio events, which are essential factors in determining whether an audio is violent or non-violent. In comparison to other metrics like the mean, the std metric provides more sensitivity to variations in the audio, making it more appropriate for our classification task.

Audio Classification Model: Our audio classifier is a simple machine-learning classification model. Specifically, we decide to use the support vector machine (SVM) classifier by setting the parameter "svm" in the initialization of our model.

To boost its classification accuracy and overall efficiency, we select the k=7 best features from the audio data using the SelectKBest method. This feature selection process identifies and retains the most relevant audio features, which have a strong relationship with the target variable. By focusing on these key features, our classifier concentrates on the most informative aspects, while reducing noise and irrelevant information.

To optimize the performance of our classifier, we proceed with fine-tuning the model's hyperparameters. For this purpose, we use the GridSearchCV method, which creates a grid search object to explore various combinations of hyperparameters and perform 2-fold cross-validation to evaluate their effectiveness. The best hyperparameters are then determined based on the results of the cross-validation, and the model's configuration is updated accordingly. This fine-tuning process ensures that our classifier is equipped with the most suitable hyperparameters, leading to improved accuracy and generalization.

Notably, we choose not to scale the selected features, as we find that scaling is not necessary for the SVM classifier's performance. This decision is reflected in our classifier's initialization, where we set the scaling parameter to None.

Training Details:

Parameter	Value
Classifier	SVC
Kernel	[linear, rbf, poly] $^{opt(*)}$
C	[0.1, 0.5, 1, 2, 5, 10, 20] $^{opt(*)}$
Gamma	[scale, auto] $^{opt(*)}$
Scaler	None

✤ Text violence detection:

For text violence detection, we use the same SVM (Support Vector Machine) classifier, fine-tuned in a similar way for our classification task. The process in detail is presented below:

Extracting the texts: To obtain the textual content from the videos of our movie dataset, we use the Whisper Automatic Speech Recognition (ASR) system developed by OpenAI [7]. Whisper is a state-of-the-art ASR model designed to convert spoken words in audio recordings into accurate and coherent text for-mat. Since all the videos in our movie dataset are in English, we take advantage of Whisper's advanced capabilities in transcribing the English language. With Whisper's specialized English language training, which ranked as the 3rd best performing language, we accurately transcribe the spoken dialogues and monologues present in our videos.

We then process these extracted texts using DistilBERT and its tokenizer for feature extraction. DistilBERT is a light-weight variant of the BERT model, renowned for its contextual understanding of text. The tokenizer breaks down each text into tokens, converting them into numerical representations suitable for the DistilBERT model. Each batch of tokenized text is then fed into DistilBERT, where it undergoes multiple layers of self-attention mechanisms, generating BERT embeddings that capture the contextual meaning of the text.

Text Classification Model: Our text classifier is the same machine-learning classification model that we use for the audio classification task. Here as well, we choose the support vector machine (SVM) classifier by setting the parameter "svm" in the initialization of our model.

To boost its classification accuracy and overall efficiency, we select the k=10 best features from the audio data using the SelectKBest method. To optimize the performance of our classifier, we proceed with fine-tuning the model's hyperparameters. Finally, we choose not to scale the selected features, as we find that scaling is neither here suitable.

Training Details:

Parameter	Value
Classifier	SVC
Kernel	[linear, rbf, poly] $^{opt(*)}$
C	[0.1, 0.5, 1, 2, 5, 10, 20] $^{opt(*)}$
Gamma	[scale, auto] $^{opt(*)}$
Scaler	None

✤ Early fusion of three modalities:

In the Early Fusion approach, we aim to achieve multi-modal classification by combining the information from different modalities (audio, image, and text) at an early stage of the model architecture.

Audio and Text Data Processing: The audio and text data are initially processed using their respective autoencoders. An autoencoder is a type of neural network that is used to learn efficient data representations in an unsupervised manner. It is particularly effective for dimensionality reduction, where the aim is to preserve as much information as possible while representing the data in a different dimensional space.

Early Fusion of Data: Once we change the dimensions of text and audio data, we combine them with the processed image data. This is done via concatenation along the feature dimension. The combination of different types of data at this early stage in the model is referred to as "early fusion". By performing early fusion, the model can learn shared representations across different types of data especially if we use an autoencoder in the process as described later.

Further Processing and Output Generation: The fused data is further processed through another autoencoder (the feature autoencoder), a dropout layer (for regularization), and a final fully connected layer. The fully connected layer is responsible for generating the final output predictions of the model. Finally, a softmax function is applied to the outputs to convert them to probabilities, which provides the final output of the model.

Training Details:

Parameter	Value
Learning rate	0.0001
Epochs	25
Batch size	8
Loss function	Cross entropy loss
Optimizer	RMSProp
Weight decay	None
Train data length	400 image-set files, 400 audio files , 400 text files

✤ Late fusion of three modalities:

In the late fusion approach, we extend our classification to an independent multi-modal setting by combining the outputs of separate classifiers, each trained on a specific modality (audio, image, or text), to make a final decision.

Evaluating Classifiers on Validation Dataset: To implement the concept of late fusion, we first need to evaluate the performance of each modality's classifier using a validation dataset. For each modality, we obtain the validation accuracy, which reflects how well the classifier performs on unseen data.

Accuracy Calculation: To compute the late fusion accuracy, we create a custom function. This function takes two tuples as input. The first tuple contains the validation accuracies for each modality (audio, text, and image) and the second one the corresponding test accuracies for each modality, which measure the performance of the classifiers on the final test dataset.

The late fusion accuracy is calculated by weighting each test accuracy with its corresponding validation accuracy and then summing up these weighted accuracies. This process ensures that the performance of each modality's classifier is appropriately considered when making the final decision. By combining the predictions from multiple modalities, we can leverage the unique information provided by each modality and potentially improve the overall accuracy and reliability of our multi-modal classification system.

III. Results

✤ Audio Classification Results:

The audio classifier performed quite well. It achieved a validation accuracy of 0.7 and a test accuracy of 0.7, indicating that it was able to distinguish between violent and non-violent audio samples with a 70% accuracy rate on both the validation and test datasets. This suggests that the Late Fusion approach with the SVM classifier and the selected seven audio features showed promising results in classifying audio samples based on their violence content.

✤ Text Classification Results:

The text classifier outperformed the audio classifier with slightly higher validation and test accuracies (0.7667) compared to the audio's (0.7). This indicates that the Late Fusion approach with the SVM classifier and the ten selected text embeddings showed more promising results in classifying text samples based on their violence content compared to the audio-based approach.

The use of text embeddings allowed the SVM classifier to capture more informative patterns and features specific to each class, leading to a better discrimination between "violent" and "non-violent" samples. Text-based features seem to carry more discriminative power in this particular dataset, which might be attributed to the nature of the data and the underlying characteristics of the text descriptions related to violence.

✤ Image Classification Results:

The image classifier has performed really well for our task. Below are a few observations:

- **i)** The model has shown significant improvement over the course of the training period. Starting with a validation accuracy of 70% in the first epoch, it has managed to reach a peak accuracy of 96.67% .
- **ii)** The validation accuracy has shown some fluctuations during the training process, with drops in epochs . However, this is expected in deep learning model training and is not a cause for concern, particularly since the overall trend is upward.
- **iii)** The training loss decreased throughout the epochs, which shows that the model was learning and improving its prediction capability with each iteration.
- **iv)** The test accuracy achieved was 76.67% suggesting that the model's generalization ability is a little bit lower than what was shown in the validation accuracy but still reasonably high.

Given these results, the model appears to be trained well without any signs of overfitting, as the validation metrics also improved along with the training loss. The training parameters for the training seem to work well for the given dataset.

Figure 1: *Late Fusion – Image Classification Model*

✤ Late Fusion Classification Results

First, we compute the weights for each modality based on their respective validation accuracies. These weights are determined by dividing each validation accuracy by the sum of all validation accuracies. The rationale behind this is to assign higher weights to modalities that have demonstrated better performance on the validation set, suggesting their greater reliability in contributing to the final prediction.

$$w_a = \frac{val_a}{val_a + val_t + val_i} = \frac{0.7}{0.7 + 0.7667 + 0.9333} \approx 0.29$$

$$w_t = \frac{val_t}{val_a + val_t + val_i} = \frac{0.7667}{0.7 + 0.7667 + 0.9333} \approx 0.32$$

$$w_i = \frac{val_i}{val_a + val_t + val_i} = \frac{0.9333}{0.7 + 0.7667 + 0.9333} \approx 0.39$$

Next, we perform a weighted average of the test accuracies using the computed weights. This means we multiply each test accuracy by its corresponding weight and then sum up these weighted accuracies. The resulting value is the late fusion accuracy, which provides an aggregated measure of performance that takes into account the individual strengths of each modality.

$$acc_{LF} = (w_a \cdot test_{acc_a}) + (w_t \cdot test_{acc_t}) + (w_i \cdot test_{acc_i}) =$$
$$= (0.29 \cdot 0.7) + (0.32 \cdot 0.7667) + (0.39 \cdot 0.7667) \approx$$
$$\approx 0.203 + 0.245 + 0.299 \approx 0.75$$

✤ Early Fusion Classification Results:

The model has demonstrated exceptional performance for our task, surpassing the previous results obtained through late fusion. Below are a few observations:

- **i)** The model demonstrated robust generalization capabilities, achieving peak validation accuracy of 83.33%.
- **ii)** Despite the complexity of the task and the multimodal nature of the data, the model maintained a robust test accuracy of 80%.
- **iii)** A significant decision in the model design was to utilize only audio and video data for early fusion. This decision was based on observations from preliminary experiments. The inclusion of text data in the early fusion process led to overfitting, potentially due to its high-dimensionality and complexity.
- **iv)** The comparison of the model's performance when using only image data versus the hybrid model using both audio and video data is insightful. The image-only model reached a validation accuracy of 96%, demonstrating a strong fit to the validation data. However, when it came to the test data, the accuracy dropped to 76.67%, indicating some loss of generalizability.

On the other hand, the hybrid model that utilized both audio and video data achieved a slightly lower validation accuracy of 83% but outperformed the image-only model on the test data, with an accuracy of 80%. This suggests that while the image-only model was highly successful at fitting the validation data, the hybrid model was more effective at generalizing to unseen test data.

$$acc_{LF} = 0.75 < acc_{EF} = 0.80$$

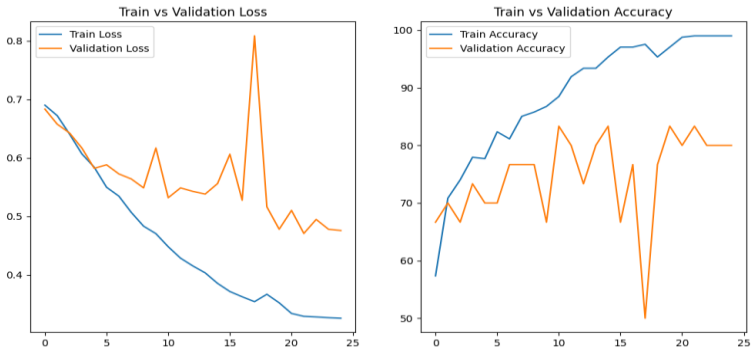

Figure 2: *Early Fusion – Hybrid Classification Model*

IV. Conclusions

In this paper, we have investigated various techniques for violence detection in videos using a multimodal approach that combines visual, audio, and text features. We addressed the challenges associated with different detection techniques and explored both single and multimodal frameworks.

For image violence detection, we utilized a deep learning model that fuses the features from pre-trained convolutional neural networks (AlexNet and SqueezeNet) using a Convolutional Long Short-Term Memory (ConvLSTM) architecture. The results showed significant improvements in accuracy, with the model achieving high classification performance. For audio violence detection, a simple Support Vector Machi-ne (SVM) classifier was used, fine-tuned for the task. By selecting relevant audio features and optimizing the model's hyperparameters, we achieved accurate violence detection in the audio domain. Text violence detection was performed using a similar SVM classifier, fine-tuned with DistilBERT embeddings. Transcribed texts obtained from videos were converted into numerical representations using DistilBERT's tokenizer, and selective feature extraction led to accurate text-based violence detection.

We also explored the early fusion and late fusion approaches to multimodal classification. In early fusion, we combined the information from different modalities at an early stage, whereas in late fusion, we combined the outputs of separate classifiers for a final decision. Both approaches showed promise in benefitting from the strengths of each modality and improving the overall accuracy of violence detection.

The results of our study demonstrate the effectiveness of multimodal violence detection in movies. Combining visual, audio, and text features allows us to capture complementary information and enhance the accuracy of the detection process. Our findings provide valuable insights into the complexities of violence detection in videos and offer guidance for future research in this area of study.

References

[1] Bin Jahlan, H. M., & Elrefaei, L. A. (2022). Detecting Violence in Video Based on Deep Features Fusion Technique. ArXiv, abs/2204.07443.

[2] Giannakopoulos, T. (2015). PyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. PLoS ONE, 10(12), e0144610.

[3] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter. arXiv:1910.01108.

[4] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-C. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. arXiv preprint arXiv:1506.04214

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems 25 (NIPS 2012)

[6] Iandola, F. N., & Dally, W. J. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360

[7] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2021). Robust Speech Recognition via Large-Scale Weak Supervision. OpenAI. arXiv preprint arXiv:2104.04494.