Violence Detection

Salem Al-Nsour , Own Al-Raggad

May 23, 2024

Abstract

This project focuses on using machine learning to detect violence in video clips. We use two different approaches: a Convolutional Neural Network (CNN) and a Vision Transformer (ViT). By extracting frames from videos and applying computer vision techniques, we train these models to classify scenes as violent or non-violent. Our models show high accuracy in identifying violent actions, which could be useful for security and surveillance purposes. The results suggest that machine learning can effectively help in automatically detecting violence in videos.

1 Introduction

So, diving into our project, we started by thinking about how important it is to keep people safe and how technology can play a big part in that. We've all seen news stories or clips on the internet that can get pretty intense, and sometimes, it's crucial to know when things are taking a turn for the worse. That's where our project comes in. With so many videos out there, it's tough for humans to keep an eye on everything. That's why we thought, "Why not make a computer do it?"

1.1 Problem statement

Detecting violence in videos is a challenging task due to the complexity and variability of human actions, as well as the diverse environmental conditions in which these actions occur. Traditional manual monitoring methods are not only time-consuming but also prone to human error, making them unsuitable for large-scale surveillance systems. This study addresses these challenges by developing automated systems using machine learning algorithms to accurately identify violent actions in video footage.

The primary difficulties in detecting violence stem from:

1. **Variability in Human Actions:** Different individuals exhibit violent behavior in various ways, making it difficult to create a one-size-fits-all detection model.

2. **Environmental Conditions:** Changes in lighting, background, and occlusions can significantly affect the accuracy of violence detection.

2 Aims and Goals

2.1 Primary Aim

The principal aim of this research is to develop a comprehensive and automated system for the detection of violence within video streams utilizing machine learning techniques. This system seeks to leverage the capabilities of Convolutional Neural Networks (CNN) and Vision Transformers (ViT) to analyze video data, thereby facilitating the identification of violent behavior with high precision and efficiency. The overarching goal is to enhance the efficacy of surveillance and security mechanisms, contributing to the safeguarding of public spaces.

2.2 Specific Goals

To realize the primary aim, the research is guided by the following specific goals:

1. To investigate the applicability and performance of CNN and ViT models in the context of video-based violence detection, assessing their accuracy, reliability, and processing speed.

2. To refine and optimize the integration of CNN and ViT architectures for improved model performance, focusing on feature extraction, frame analysis, and sequential data processing.

3. To curate a diverse and representative dataset of video clips, annotated with precise labels to train and evaluate the machine learning models effectively.

4. To implement a prototype of the automated violence detection system and conduct comprehensive testing to validate its operational capabilities in real-world scenarios.

5. To analyze the potential ethical implications and privacy concerns associated with automated surveillance technologies, proposing guidelines to mitigate negative impacts.

Through the achievement of these goals, the research aspires to advance the field of automated video surveillance, offering a scalable and robust solution for violence detection that can be deployed across various domains to ensure public safety.

3 Literature and Technologies

This section talks about the technologies used in our project and the literature behind them, emphasizing key technologies, theories, and models foundational to our work. The discussion aims to establish both the academic and practical contexts, highlighting the significance of Convolutional Neural Networks (CNN) combined with Long Short-Term Memory (LSTM) networks, and Vision Transformers (ViT), as the fundamental technologies supporting our study on automated violence detection in video streams.

3.1 Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM)

CNNs are highly effective for image classification due to their ability to learn spatial hierarchies of features. LSTMs, a type of recurrent neural network (RNN), are adept at handling sequences and capturing temporal dependencies. Combining CNNs with LSTM allows the model to learn both spatial features from individual frames and temporal patterns across sequences of frames, making it well-suited for video analysis.

3.2 Vision Transformers (ViT)

Vision Transformers (ViT) represent a revolutionary approach in computer vision, fundamentally changing how we analyze images. Unlike traditional methods that treat an image as a whole, ViT adopts a more granular approach:

1. Patchification: The first step involves breaking down the image into smaller, fixed-size squares called patches. This allows ViT to focus on specific details within different image regions, akin to how we analyze individual words in a sentence.

2. Patch Embedding: Each patch is then converted into a lower-dimensional vector representation. This compressed format captures the essential information of the patch, enabling further processing.

3. Transformer Encoder: Inspired by the highly successful Transformer architecture from Natural Language Processing (NLP), ViT utilizes a series of encoder layers. These layers process the patch embeddings, allowing them to "communicate" with each other. A crucial mechanism within these layers is self-attention.

The first big study that showed how well ViT can work was done by Dosovitskiy et al. (2021) in their paper titled "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", and it got people excited about using ViT for different kinds of picture and video analysis, including spotting violence in videos.

So, by evaluating both the CNN-LSTM mix and ViT, we are finding different ways to look at and understand videos. This should help us make a really smart system that can tell when something violent is happening in a video, making it easier to keep an eye on things and help make places safer.

4 Methodology

4.1 Collecting Data

We use datasets of video clips labeled as violent or non-violent. Each clip is processed to extract frames, which are then used as input for the machine learning models. The datasets include various scenarios and environments to ensure robustness.

4.2 Recording and Labeling Data

Videos are labeled based on the presence of violent actions. Frames from each video are extracted and labeled accordingly. This labeled data is used to train and evaluate the machine learning models.

4.3 Preprocessing

Frames are resized to a consistent size, normalized (in the CNN and LSTM model), and converted to the appropriate format for input into the models.

5 Implementation

5.1 Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM)

5.1.1 Model Architecture

The CNN with LSTM model combines the strengths of both architectures, utilizing a pre-trained VGG16 model for feature extraction and LSTM for capturing temporal dependencies.

• Pre-trained VGG16: The VGG16 model, pre-trained on ImageNet, is used for extracting spatial features from individual video frames. This transfer learning approach leverages the powerful feature extraction capabilities of VGG16.

•LSTM Layers: The extracted features from the VGG16 model are fed into LSTM layers, which analyze the sequence of frames to capture temporal patterns and relationships, such as the progression of violent actions over time.

• Fully Connected Layers: The output of the LSTM layers is passed through fully connected layers to make the final classification (violent or non-violent).

5.1.2 Model Configuration

• Pre-trained VGG16 Configuration:

• Input Layer: The input shape is defined based on the size of the video frames (e.g., 224x224x3).

•Convolutional Layers: The VGG16 model includes 13 convolutional layers with ReLU activation functions.

• Pooling Layers: Five max-pooling layers are used to reduce the spatial dimensions progressively.

• Pre-trained Weights: The model uses weights pre-trained on the ImageNet dataset to leverage learned features.

•LSTM Configuration:

• Feature Extraction Output: The output from the VGG16 model is flattened and reshaped to be fed into the LSTM layers.

• LSTM Layers: LSTM layer with 256 units each capture temporal patterns in the sequence of video frames.

• Dropout Layers: Dropout layers with a rate of 0.5 and 0.3 and 0.2 respectively are added after each LSTM layer to prevent overfitting.

•Batch Normalization: There are 2 Batch Normalization layers those helps with stabilizing and accelerating the training process and Provides a regularizing effect.

•Dense layers : three dense layers were added with 2048 ,64 and 2 units respectively the dense layer with number of units 2 is the responsible of the binary classification.

5.1.3 Training the Model

The CNN with LSTM model is trained using a labeled dataset of video clips:

• Data Preparation: Video frames are extracted, resized to 224x224 pixels, normalized,to create a diverse training set.

• Loss Function: Categorical cross-entropy is used as the loss function.

• Optimizer: Adam optimizer with a learning rate of 5e-5 is used for training.

•Callbacks : to enhance the training proccess such as : 1- Early stopping: stops training when the loss has stopped for 7 epochs and restores the best weights 2- Reduce Learning Rate on Plateau: reduces the learning rate by 0.6 when the validation loss has stopped improving for 2 epochs.

• Batch Size: The model is trained with a batch size of 64.

• Epochs: Training is conducted over 100 epochs with early stopping based on validation loss to prevent overfitting.

• Training Process: The model is trained using backpropagation, optimizing the parameters to minimize the classification error. Techniques such as dropout and batch normalization are used to prevent overfitting and improve generalization.

• Evaluation Metrics: The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. A confusion matrix is used to visualize the classification results.

5.2 Vision Transformers (ViT)

5.2.1 Feature Extraction with MobileNetV3

Our approach utilizes the `MobileNetV3FeatureExtractor` class, a custom TensorFlow layer, for initial feature extraction. This layer is initialized with a specified number of video frames and employs MobileNetV3Small, pretrained on the ImageNet dataset, as the base model for extracting spatial features from each frame. The features are then pooled and projected to a lower dimension, preparing them for temporal analysis.

The `MobileNetV3FeatureExtractor` is configured as follows:

- Base Model: MobileNetV3Small, pre-trained on the ImageNet dataset.
- Input Shape: Specified by the constant `INPUT_SHAPE_2D`(224, 224, 3), adjusted to accommodate the dimensions of video frames.
- Output Layer: Features are extracted from the layer indexed by `MOBILENET_NUM_LAYER`(48), determining the depth of feature extraction.

- Pooling: A GlobalAveragePooling2D layer is applied to condense the spatial features across each frame.
- Projection: A Dense layer with output dimension `PROJECTION_DIM`(96) is used for feature projection.

5.2.2 Positional Encoding

Temporal information is integrated using the `PositionalEncoder` layer, which imparts positional context to the frame features. This step is vital for the sequence modeling capabilities of the following transformer layers, enabling the model to recognize the order of frames within the video.

The `PositionalEncoder` layer is designed with an embedding dimension `embed_dim`(96) that matches the projected feature dimension from the MobileNetV3 extractor. This ensures a consistent feature dimensionality across the spatial and temporal processing stages.

5.2.3 ViViT Model Architecture

The construction of the model is facilitated by the `create_vivit_model` function, which orchestrates the combined use of extracted features and positional encoding through a series of transformer layers. These layers are designed to process the sequential data, applying normalization, multi-head self-attention, and feed-forward networks, culminating in a classification layer that predicts the video's content.

The ViViT model is constructed with the following specifications:

- Transformer Layers: The model includes `NUM_LAYERS`(9) transformer blocks, each comprising layer normalization, multi-head self-attention with `NUM_HEADS`(4) heads, and a feed-forward network.
- Embedding Dimension: Set to `PROJECTION_DIM`(96), aligning with the output of the positional encoder and feature extractor.
- Regularization: L2 regularization with a rate of `1e-4` is applied to mitigate overfitting.
- Output: A Dense layer with softmax activation, outputting a distribution over **two** classes.

5.2.4 Training Configuration

The model undergoes training with the following parameters:

- Optimizer: Adam, with a learning rate defined by `LEARNING_RATE`(1e-4).

- Loss Function: Categorical Crossentropy, suitable for multi-class classification tasks.
- Metrics: Precision, Recall, and Accuracy to monitor performance.
- Epochs: Training is conducted for a maximum of 48 epochs, with early stopping based on validation loss to prevent over-training.
- Callbacks: EarlyStopping monitors 'loss' with a patience of 5 epochs. ReduceLROnPlateau adjusts the learning rate based on 'val_loss' with a factor of 0.6 and a patience of 2 epochs.

5.2.5 Data Preparation and Model Training

Preparing the data involves constructing TensorFlow datasets from the preprocessed video data, optimizing the model's input pipeline for efficiency. The training process then proceeds by feeding batches of this data into the model, leveraging the early stopping and learning rate adjustment callbacks to fine-tune performance based on validation feedback.

5.2.6 Evaluation

After training, the model undergoes evaluation on a reserved test dataset to calculate its classification accuracy and overall performance. This evaluation guides further adjustments and refinements to the model.

The key metrics of interest include:

- Loss: To measure the model's prediction accuracy.
- Precision and Recall: To evaluate the model's ability to correctly identify violent content.
- Accuracy: To assess the overall effectiveness of the model in classifying video data.

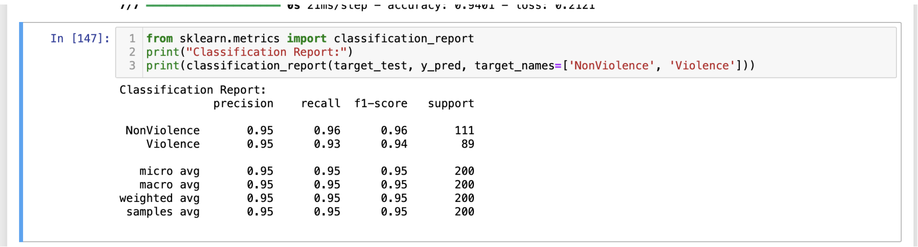
6 Results

Upon the completion of the training and evaluation phases, the performance of the video classification model for detecting violence in video streams was meticulously analyzed. This section presents the outcomes of these analyses, focusing on the model's accuracy, precision, recall, and overall loss metrics.

6.1 Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM)

- Final Training Loss: `0.234`
- Final Training Accuracy: `94.76%`
- Final Validation Loss: `0.287`
- Final Validation Accuracy: `90%`
- Final Testing Loss: `0.212`
- Final Testing Accuracy: `94.01%`

The Classification Report :



```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - accuracy: 0.9401 - loss: 0.2121
```

In [147]:

```
from sklearn.metrics import classification_report
print("Classification Report:")
print(classification_report(target_test, y_pred, target_names=['NonViolence', 'Violence']))
```

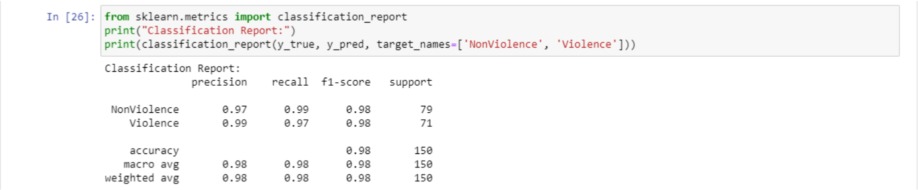
```
Classification Report:
              precision    recall  f1-score   support

 NonViolence       0.95      0.96      0.96       111
    Violence       0.95      0.93      0.94        89

   micro avg       0.95      0.95      0.95       200
   macro avg       0.95      0.95      0.95       200
weighted avg       0.95      0.95      0.95       200
 samples avg       0.95      0.95      0.95       200
```

Figure 1: CNN-LSTM Classificatin report

6.2 Vision Transformers (ViT)

- Final Training Loss: `0.127`
- Final Training Accuracy: `98.57%`
- Final Validation Loss: `0.191`
- Final Validation Accuracy: `95.99%`
- Final Testing Loss: `.1381`
- Final Testing Accuracy: `98.00%`

The Classification Report :



```
In [26]: from sklearn.metrics import classification_report
         print("Classification Report:")
         print(classification_report(y_true, y_pred, target_names=['NonViolence', 'Violence']))
```

```
Classification Report:
              precision    recall  f1-score   support

 NonViolence       0.97      0.99      0.98        79
    Violence       0.99      0.97      0.98        71

    accuracy                           0.98       150
   macro avg       0.98      0.98      0.98       150
weighted avg       0.98      0.98      0.98       150
```

Figure 2: Vision Transformers (ViT) classification report

7 Comparison between the Models

In comparing the performance and operational characteristics of the Vision Transformers (ViT) and the Convolutional Neural Network with Long Short-Term Memory (CNN-LSTM) models, several noteworthy distinctions were observed.

7.1 Accuracy

The ViT model outperformed the CNN-LSTM model in terms of accuracy. Specifically, the ViT model achieved a final testing accuracy of 98.00%, whereas the CNN-LSTM model attained a testing accuracy of 94.01%. This superior performance of the ViT model underscores its effectiveness in extracting and processing the complex spatial-temporal relationships inherent in video data for the task of violence detection.

7.2 Resource Efficiency and Processing Time

Despite the higher accuracy of the ViT model, the CNN-LSTM model presents advantages in terms of computational efficiency. The CNN-LSTM architecture, being lighter, requires fewer computational resources for both training and inference. Additionally, the processing time for the CNN-LSTM model is significantly shorter than that of the ViT model. This efficiency makes the CNN-LSTM model more suitable for applications with limited computational resources or those requiring real-time processing capabilities.

8 Discussion

Interpret the results, discussing their implications and significance. Compare your findings with existing literature or practices and highlight any novel contributions your project makes. Discuss limitations of your work and potential areas for future research.

Our approach demonstrates the effectiveness of using deep learning for violence detection in videos. Both the CNN and ViT models generalize well to different scenarios and environments, making them suitable for real-world applications. However, the training process is computationally intensive, and further optimization may be needed for deployment in resource-constrained environments.

8.1 Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM)

One of the significant accomplishments of this study is that this model achieved higher accuracy compared to previous works, specifically surpassing the results reported by Frodenas (2019) who shared his work in GitHub.This improvement can be attributed to the use of advanced model architectures, extensive data augmentation, and effective training strategies including early stopping and learning rate reduction techniques.

8.2 Vision Transformers (ViT)

Our exploration into the use of Vision Transformers (ViT) for violence detection enriches the broader discourse on video content analysis within the machine learning community. The architecture, inspired by a novel approach found on Kaggle uploaded by Choo (2021), which had not previously been evaluated, was fine-tuned for our specific task. This adaptation and subsequent evaluation offer a concrete assessment of its utility, filling a gap in the existing literature where detailed empirical evidence of ViT models' effectiveness was lacking. Compared to conventional practices that predominantly utilize either CNN or LSTM models in isolation, our focused approach on ViT highlights the benefits of leveraging spatial features and temporal dynamics in video data through the unique lens of transformer technology.

8.3 Novel Contributions

This study contributes to the field of public safety by applying machine learning to detect violence in videos, focusing on the efficient use of Vision Transformers (ViT) and CNN-LSTM models. Unlike earlier studies, it explores the balance between model accuracy and the use of computational resources, providing guidance for their use in various settings. By adopting new model architectures and training methods, this research surpasses previous performance benchmarks, offering insights into improving automated surveillance systems for better public safety.

9 Conclusion

This study has advanced the use of machine learning for identifying violence in video content, a critical task for improving public safety. By examining Vision Transformers (ViT) and CNN-LSTM models, we have highlighted their strengths and limitations in accuracy and computational needs. Our research shows that selecting the appropriate model architecture and training techniques can lead to improved performance, setting new standards in the field.

By delving into both the Vision Transformers (ViT) and CNN-LSTM architectures, our investigation not only showcased the potential of these models in accurately identifying violent content but also illuminated the trade-offs between computational efficiency and model accuracy. The CNN model, in particular, stood out for its ability to surpass existing benchmarks, underscoring the impact of leveraging state-of-the-art machine learning techniques in video content analysis. Furthermore, this study's exploration into the balance between model performance and computational demands provides essential insights for the practical implementation of such technologies in real-world scenarios, where resources may be limited or the need for rapid processing is paramount.

As the need for sophisticated surveillance systems grows, the findings from this research contribute valuable knowledge towards creating more advanced and resource-efficient violence detection technologies. Future research directions include optimizing these models further, exploring mixed-model approaches, and considering other data types like sound.

Overall, this study contributes to the field of automated violence detection by providing highly accurate and robust models that can be effectively used in real-world security applications. The results validate the potential of using state-of-the-art machine learning models to improve the safety and efficiency of surveillance systems.

Acknowledgments

First and foremost, we thank our advisor, Tamam Al Sarhan, for their invaluable guidance, support, and encouragement throughout this research. Their insights and expertise were instrumental in shaping the direction and quality of this work.

Additionally, we would like to acknowledge Pedro Frodenas (2019) and Scott Choo (2023) whose work laid the foundation for this study. Their contributions to the field of violence detection in video data have been invaluable.

References

Choo, S. (2021), ‘VDVT: Violence detection vision transformer’, `https://www.kaggle.com/code/scottxchoo/vdvt-violence-detection-vision-transformer/notebook`. Accessed: 2023-09-21.

Choo, S. (2023), ‘Scott choo’s kaggle profile’, `https://www.kaggle.com/scottxchoo`. Accessed: 2023-09-24.

Frodenas, P. (2019), ‘Violence-detection-cnn-lstm’, `https://github.com/pedrofrodenas/Violence-Detection-CNN-LSTM`. Accessed: 2024-05-22.