Mini Project Report

on

Public Safety Monitoring through Violence Detection

Submitted by

B. Raghu Vamsi	**D. Sathvik**	**S. Raj Kumar**	**Suhaas**
22BDS014	22BDS020	22BDS055	22BDS056

Under the guidance of

Dr. Shirsendu Layek

Designation

DEPARTMENT OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

12/04/2025

Certificate

This is to certify that the project entitled **Public Safety Monitoring through Violence Detection** is a bonafide record of the Mini Project coursework presented by the students whose names are given below during the academic year **2025**, in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering.

Roll No	**Name of Student**
22BDS014	B. Raghu Vamsi
22BDS020	D. Sathvik
22BDS055	S. Raj Kumar
22BDS056	Suhaas

Dr. Shirsendu Layek
(Project Supervisor)

Contents

List of Figures

List of Tables

1 Introduction

Public safety has become an increasingly critical concern in modern society, particularly in densely populated or high-risk environments such as shopping malls, public transport stations, and urban streets. Traditional surveillance systems rely heavily on manual monitoring, which is time-consuming, prone to human error, and inefficient in promptly identifying violent incidents.

To address this issue, we propose an automated violence detection system using deep learning techniques to analyze video footage and detect aggressive behavior. The goal is to enhance surveillance systems by providing real-time alerts for potentially violent events, allowing authorities to respond swiftly and effectively.

For this purpose, we used the Real-Life Violence Dataset, which consists of 2,000 videos evenly divided into two categories: *Violence* and *NonViolence*. These clips include real-world scenarios such as fighting, sports, daily activities, and crowd interactions.

To evaluate and compare performance, we implemented three different deep learning architectures:

- **MobileNet + BiLSTM:** A lightweight and efficient combination of a pre-trained convolutional neural network and a bidirectional LSTM for temporal pattern learning.
- **VGG16 + LSTM:** A classical CNN architecture used to extract spatial features, followed by LSTM for sequence modeling.
- **Vision Transformer (ViT):** A transformer-based model that directly captures both spatial and temporal relationships using attention mechanisms.

These models were trained and evaluated to assess their accuracy, speed, and suitability for real-time deployment in public safety monitoring systems.

2 Related Work

The field of automated violence detection in video surveillance has evolved significantly with the advancement of deep learning techniques. Traditional surveillance systems rely heavily on manual monitoring, which is prone to human error and delayed response times. To address these challenges, numerous studies have explored the integration of artificial intelligence, particularly deep learning, to automate threat detection.

Early approaches employed hand-crafted features and shallow classifiers, which lacked the robustness and generalization capability required for real-world deployment. The advent of Convolutional Neural Networks (CNNs) revolutionized visual recognition tasks, including violence detection. CNNs are adept at capturing spatial features from individual frames, making them suitable for recognizing violent cues such as aggressive postures or movements.

However, since violent actions unfold over time, temporal modeling became essential. This led to the adoption of Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, to model the sequential nature of video data. Several studies have proposed CNN-LSTM hybrids, where CNNs extract spatial features and LSTMs capture temporal dynamics. For example, models like VGG16 + LSTM and MobileNet + BiLSTM have been effectively used for activity recognition and violence detection due to their balance of accuracy and computational efficiency.

Transformer-based models, such as the Vision Transformer (ViT), have recently emerged as powerful alternatives to CNNs. ViTs offer a global receptive field and excel at modeling long-range dependencies, making them suitable for complex video understanding tasks. Some studies simulate ViT behavior by combining pre-trained CNNs (like ResNet50) with LSTMs, enabling both efficient spatial extraction and temporal analysis.

Various publicly available datasets such as Hockey Fight, Movies Fight, RWF-2000, and the Real-Life Violence Dataset have been widely used to train and benchmark these models. These datasets help in developing systems that can distinguish between violent and non-violent scenarios under diverse environmental conditions.

Despite these advancements, many existing models face challenges such as high computational cost, limited scalability to real-time systems, or poor generalization to real-world surveillance footage. Our work addresses these issues by implementing and comparing three deep learning architectures—VGG16 + LSTM, MobileNet + BiLSTM, and ViT + LSTM—on the Real-Life Violence Dataset to evaluate their performance and suitability for real-time public safety applications.

3 Data and Methods

3.1 Dataset Description

For this project, we used the **Real-Life Violence Dataset**, which consists of 2,000 short video clips categorized into two classes: **Violence** and **NonViolence**.

- **Total Videos:** 2,000
- **Violence Class:** 1,000 videos containing real-life violent activities such as fights, assaults, and aggressive behavior.
- **NonViolence Class:** 1,000 videos depicting peaceful activities like walking, eating, dancing, or playing sports.
- **Format:** AVI
- **Video Length:** Ranges from a few seconds to short clips.
- **Frame Extraction:** 16 equally spaced frames per video were extracted for training and evaluation.

This dataset provides a realistic and balanced distribution of both violent and non-violent scenarios, making it suitable for training deep learning models to identify patterns associated with violence in public settings.

3.2 Data Preprocessing

To prepare the dataset for training and evaluation, the following preprocessing steps were applied:

- **Frame Extraction:** From each video, 16 equally spaced frames were extracted to capture temporal dynamics across the clip.
- **Resizing:** All frames were resized to 64×64 pixels to ensure consistency and reduce computational load.
- **Normalization:** Pixel values were scaled to the range [0, 1] by dividing by 255 to improve model convergence.
- **Label Encoding:** Video labels were converted into one-hot encoded vectors for classification ($Violence = [1, 0]$, $NonViolence = [0, 1]$).
- **Train-Test Split:** The dataset was divided into training, validation, and test sets to evaluate generalization performance.
- **Shuffling:** The data was shuffled to ensure a balanced class distribution and to reduce learning bias.

These steps ensured that the input data was properly formatted for deep learning, while preserving the temporal and spatial information necessary for violence detection.

3.3 Model Architectures

To effectively classify violent and non-violent video clips, we implemented three deep learning models that combine Convolutional Neural Networks (CNNs) for spatial feature extraction and Recurrent Neural Networks (RNNs) for temporal modeling. Each input video was represented as a sequence of 16 frames, with each frame sized $64 \times 64 \times 3$.

VGG16 + LSTM

This model employs the VGG16 architecture pre-trained on ImageNet for spatial feature extraction. The CNN is encapsulated within a `TimeDistributed` layer, allowing it to process each frame individually. The extracted features are then passed to an LSTM layer to capture temporal dependencies.

- **Input:** Sequence of 16 frames of size $64 \times 64 \times 3$
- **Feature Extractor:** Pre-trained VGG16 (frozen layers), wrapped in `TimeDistributed`
- **Temporal Modeling:** LSTM with 64 units
- **Classification Head:**
  - Dense (256 units, ReLU activation) + Dropout (rate = 0.5)
  - Output: Dense (2 units, Softmax activation)
- **Loss Function:** Categorical Crossentropy
- **Optimizer:** Stochastic Gradient Descent (SGD)

Model Summary:

- Total Parameters: 15,272,770
- Trainable Parameters: 558,082
- Non-trainable Parameters: 14,714,688

Layer (type)	Output Shape	Param #
time_distributed (TimeDistributed)	(None, 16, 2, 2, 512)	14,714,688
time_distributed_1 (TimeDistributed)	(None, 16, 2048)	0
lstm (LSTM)	(None, 64)	540,928
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 256)	16,640
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 2)	514

Figure 1. Architecture of the VGG16 + LSTM model.

MobileNet + BiLSTM

This architecture uses MobileNet for lightweight spatial feature extraction, followed by a Bidirectional LSTM layer to capture temporal patterns in both forward and backward directions. It is computationally efficient while maintaining strong performance.

- **Input:** Sequence of 16 frames of size $64 \times 64 \times 3$
- **Feature Extractor:** Pre-trained MobileNet inside `TimeDistributed`
- **Temporal Modeling:** Bidirectional LSTM with 64 units
- **Classification Head:**
  - Dense (256 units, ReLU) + Dropout (0.25)
  - Dense (128 units, ReLU) + Dropout (0.25)
  - Dense (64 units, ReLU) + Dropout (0.25)
  - Dense (32 units, ReLU) + Dropout (0.25)
  - Output: Dense (2 units, Softmax)
- **Loss Function:** Categorical Crossentropy
- **Optimizer:** Stochastic Gradient Descent (SGD)

Model Summary:

- Total Parameters: 3,637,090
- Trainable Parameters: 3,060,642
- Non-trainable Parameters: 576,448

Layer (type)	Output Shape	Param #
time_distributed (TimeDistributed)	(None, 16, 2, 2, 1280)	2,257,984
dropout (Dropout)	(None, 16, 2, 2, 1280)	0
time_distributed_1 (TimeDistributed)	(None, 16, 5120)	0
bidirectional (Bidirectional)	(None, 64)	1,319,168
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 256)	16,640
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_4 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
dropout_5 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 2)	66

Figure 2. Architecture of the MobileNet + BiLSTM model.

Vision Transformer (ViT) + LSTM

To leverage transformer-based concepts, this model uses ResNet50 (pre-trained on ImageNet) to simulate a ViT-like pipeline. Frame-level features extracted via ResNet50 are processed sequentially using an LSTM layer to model temporal dynamics.

- **Input:** Sequence of 16 frames of size $64 \times 64 \times 3$
- **Feature Extractor:** Pre-trained ResNet50 with Global Average Pooling inside `TimeDistributed`
- **Temporal Modeling:** LSTM with 128 units
- **Classification Head:**
  - Dense (64 units, ReLU)
  - Output: Dense (2 units, Softmax)
- **Loss Function:** Sparse Categorical Crossentropy
- **Optimizer:** Adam (learning rate $= 1 \times 10^{-4}$)

Model Summary:

- Total Parameters: 24,710,722
- Trainable Parameters: 24,657,602
- Non-trainable Parameters: 53,120

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 16, 64, 64, 3)	0
time_distributed (TimeDistributed)	(None, 16, 2048)	23,587,712
lstm (LSTM)	(None, 128)	1,114,624
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dense_1 (Dense)	(None, 2)	130

Figure 3. Architecture of the Vision Transformer (ViT) + LSTM model.

3.4 Training and Evaluation

To ensure a fair and objective comparison among the three models—VGG16 + LSTM, MobileNet + BiLSTM, and Vision Transformer (ViT with a ResNet50 backbone)—a uniform training strategy was employed across all experiments. Each model was trained using the same dataset splits, preprocessing techniques, and evaluation metrics.

Training Setup

All models were trained for 10 epochs with a batch size of 8. A validation split of 20% was used to monitor model performance during training. The MobileNet and VGG16 models were trained using categorical cross-entropy loss, while the ViT model employed sparse categorical cross-entropy. Stochastic Gradient Descent (SGD) served as the optimizer for MobileNet and VGG16, whereas the ViT model used the Adam optimizer with a learning rate of 1×10^{-4}.

Callbacks Used

To improve generalization and mitigate overfitting, the following callbacks were utilized:

- **EarlyStopping:** Monitored validation accuracy and restored the best weights if no improvement was observed for 10 consecutive epochs.
- **ReduceLROnPlateau:** Decreased the learning rate when the validation loss plateaued, with a reduction factor of 0.6 and a patience of 10 epochs.

Hardware Configuration

Training was conducted on GPU-enabled hardware to accelerate computation. The average training time per epoch for each model was:

- **MobileNet + BiLSTM:** Approximately 150 seconds
- **VGG16 + LSTM:** Approximately 450 seconds
- **ViT (ResNet50 backbone):** Approximately 1300 seconds

Final Evaluation Results

Post-training, all models were evaluated on the test set. The classification accuracy and loss values are summarized below:

- **MobileNet + BiLSTM:** Accuracy = **89.45%**, Loss = 0.2138
- **VGG16 + LSTM:** Accuracy = **70.00%**, Loss = 0.5864
- **Vision Transformer (ViT):** Accuracy = **92.75%**, Loss = 0.1506

Among the three models, the ViT-based architecture achieved the highest accuracy, slightly outperforming MobileNet + BiLSTM. The underperformance of the VGG16-based model can be attributed to its large number of frozen layers and its reduced effectiveness in extracting features from low-resolution frames.

4 Results and Discussions

This section presents the performance outcomes of the three deep learning models—MobileNet + BiLSTM, VGG16 + LSTM, and Vision Transformer (ViT with ResNet50 backbone)—on the Real-Life Violence Dataset.

4.1 Model Performance

The models were evaluated on the test set after training. Table 1 summarizes their classification accuracy and loss:

Table 1
Test accuracy and loss for each model

Model	Test Accuracy	Test Loss
MobileNet + BiLSTM	**89.45%**	0.2138
VGG16 + LSTM	70.00%	0.5864
ViT (ResNet50 Backbone) + LSTM	**92.75%**	0.1506

The Vision Transformer model achieved the highest accuracy, followed closely by the MobileNet + BiLSTM model. The VGG16-based model performed comparatively lower, possibly due to its heavier architecture and reduced generalization on low-resolution frames.

4.2 Training Curves

To gain insights into the training dynamics and convergence behavior, we plotted the training and validation accuracy/loss curves for each model. These visualizations help identify signs of underfitting, overfitting, and training stability.

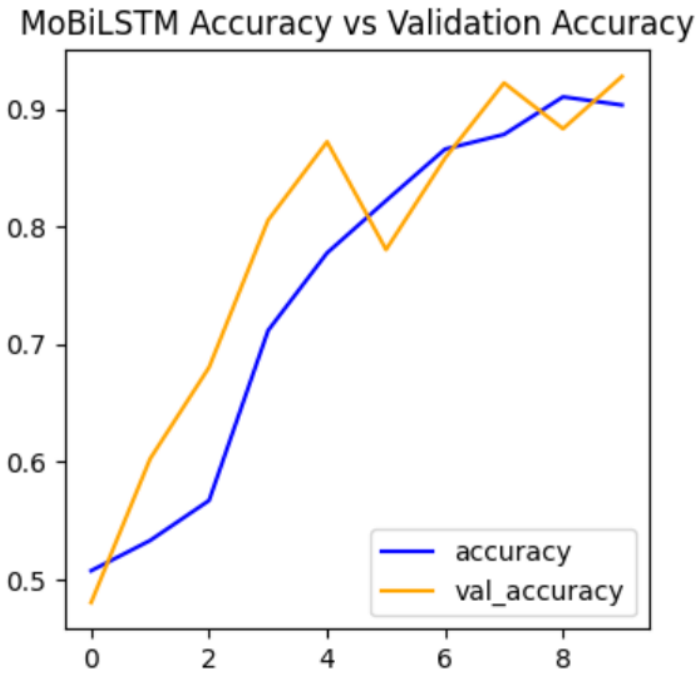

Figure 4. Training and validation accuracy/loss curves for MobileNet + BiLSTM.

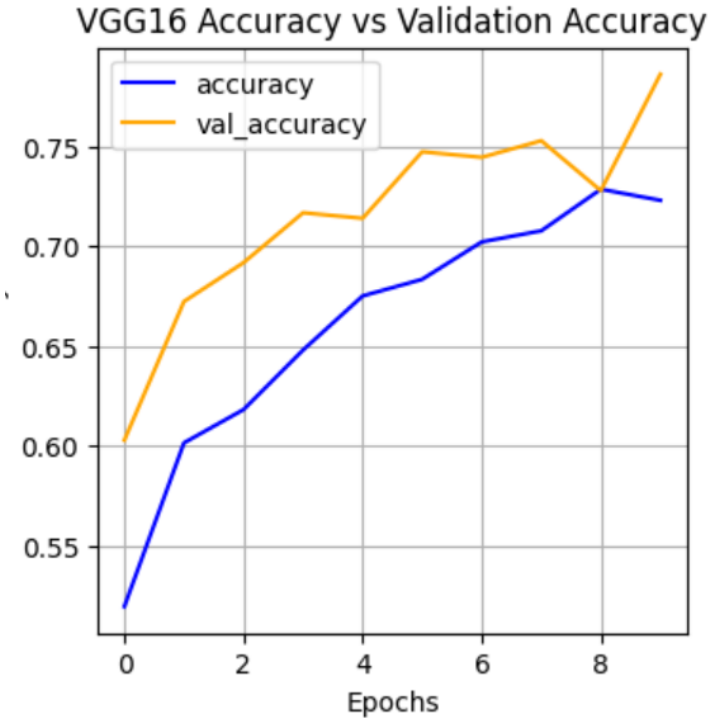

Figure 5. Training and validation accuracy/loss curves for VGG16 + LSTM.

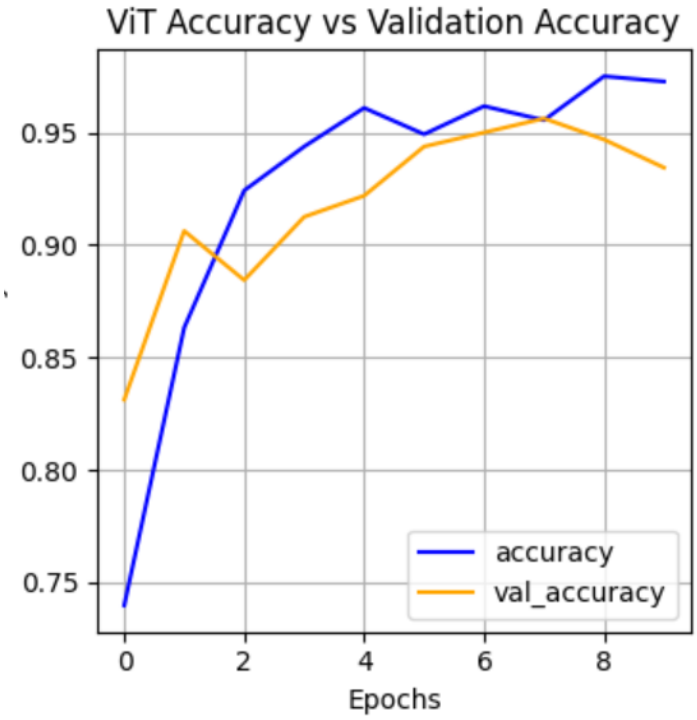

Figure 6. Training and validation accuracy/loss curves for Vision Transformer (ViT + LSTM).

4.3 Confusion Matrices

To evaluate class-wise performance and identify misclassifications, confusion matrices were generated for each model. These matrices illustrate the number of correctly and incorrectly classified instances for both *Violence* and *Non-Violence* classes.

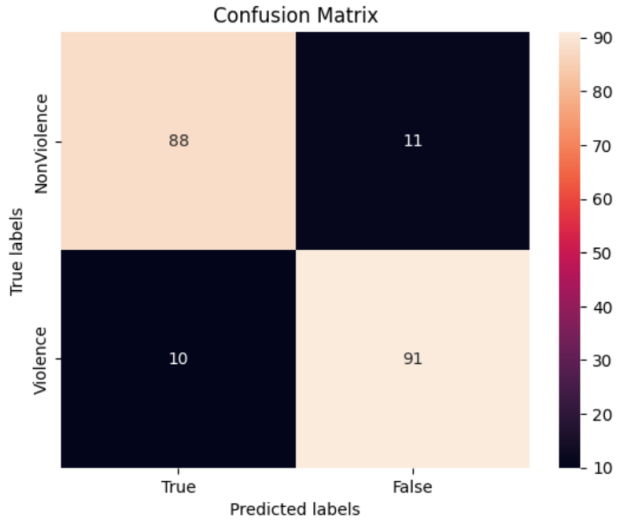

Figure 7. Confusion matrix for the MobileNet + BiLSTM model.

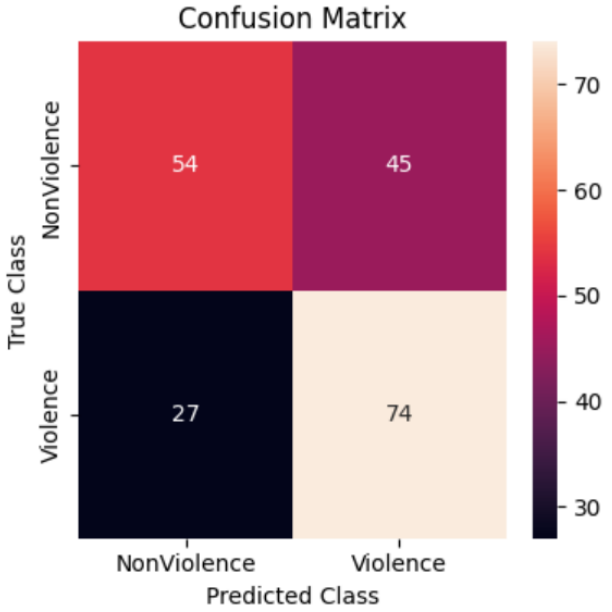

Figure 8. Confusion matrix for the VGG16 + LSTM model.

5 Predictions and Visualizations

To gain insights into the model behavior in real-world scenarios, we performed qualitative evaluations using randomly selected video clips from the test set. Each clip comprises a sequence of 16 frames, processed and classified by the trained models into either `Violence` or `NonViolence` categories.

5.1 Prediction Pipeline

The prediction pipeline involves the following sequential steps:

1. Load a video sample and extract a sequence of 16 consecutive frames.
2. Preprocess the frames by resizing them to 64×64 and normalizing pixel values.

3. Feed the processed frame sequence into the trained model.
4. Obtain the predicted class label from the model output — either `Violence` or `NonViolence`.

5.2 Sample Predictions

Figures 9 and 10 illustrate examples of predictions made by the MobileNet + LSTM model, demonstrating its ability to differentiate between aggressive and benign human activities.

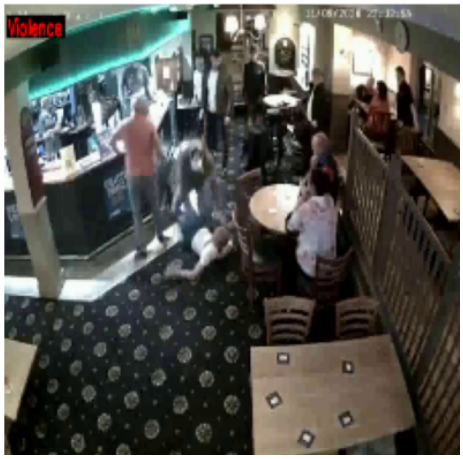

Figure 9. Prediction: `Violence` — The model correctly identifies an instance of aggressive or hostile behavior.

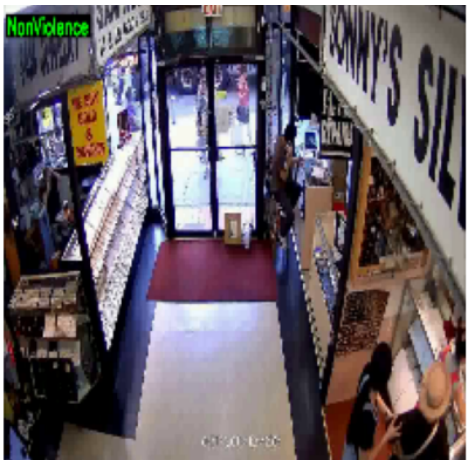

Figure 10. Prediction: `NonViolence` — The model accurately classifies calm actions such as walking, talking, or playing.

6 Conclusion

In this project, we addressed the challenge of real-time violence detection in videos to aid public safety monitoring. Leveraging the Real-Life Violence Dataset, we trained and evaluated three deep learning models with spatial-temporal learning capabilities:

- **MobileNet + BiLSTM:** A compact and efficient architecture, balancing performance and speed.
- **VGG16 + LSTM:** A classical CNN-LSTM stack, which underperformed due to architectural inefficiencies on low-resolution frames.
- **Vision Transformer (ViT + LSTM):** A transformer-based architecture that achieved the highest accuracy, highlighting its effectiveness in capturing spatial features and temporal dependencies.

Our experimental results showed that the ViT-based model achieved the best performance with a test accuracy of 92.75%, followed by the MobileNet + BiLSTM model at 89.45%. The VGG16-based model lagged behind with 70.00% accuracy. These outcomes suggest that transformer-based vision models, even with limited training epochs and moderate resolution, are highly capable of handling the intricacies of violence detection tasks.

References

[1] Real-Life Violence Dataset. Kaggle. https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset/data

[2] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861.

[3] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).

[4] Dosovitskiy, A., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929.

[5] Chollet, F., et al. (2015). Keras. https://keras.io

[6] Abadi, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org