Deep Learning Approaches for Violence and Non-Violence Classification : A Case Study Using YOLOv5 and CNN

Albandri Tami , Lama Al Turki , Ghaidaa Elazhary
Dept. of Data Science and Analytics
University of Jeddah, Saudi Arabia

***Abstract*—With the exponential rise of user-generated content on digital platforms, the risk of exposure to harmful and violent imagery has grown significantly. Such content poses not only psychological harm to users but also legal and ethical challenges for hosting platforms. Manual moderation is slow, costly, and often inconsistent—necessitating scalable and automated alternatives.**

This paper investigates the application of deep learning models for detecting violent content in static images. Specifically, it compares two architectures: YOLOv5, a real-time object detection and classification model known for its speed and efficiency, and a custom Convolutional Neural Network (CNN) designed to extract spatial features from images.

Both models were trained on a curated dataset of 20,000 labeled images sourced from Kaggle, evenly split between violent and non-violent scenes. Standard classification metrics—including accuracy, precision, recall, and F1-score—were used to evaluate performance under identical conditions.

Results show that YOLOv5 outperforms the CNN in both accuracy and class-wise recall, achieving 95% overall accuracy with a strong balance across both classes. In contrast, the CNN model, despite high training accuracy, demonstrated poor generalization—particularly struggling with the violent class—highlighting limitations in static spatial-only approaches.

This study contributes to AI-driven content moderation research by offering a comparative evaluation of lightweight and conventional models, supporting the use of optimized real-time architectures like YOLOv5 in automated moderation pipelines where speed and reliability are critical.

I. Introduction

Over the past few years, platforms such as YouTube, TikTok, and Instagram have seen a huge increase in the number of videos and images shared by users. Although much of this content is harmless, there is a growing concern about the spread of violent material that can affect viewers negatively. Young audiences, in particular, may be more vulnerable to this kind of content. Manually checking every uploaded image is not a realistic solution. It takes time, costs money, and puts pressure on human moderators who must constantly view disturbing visuals. These challenges have led to a stronger interest in using automated tools to support content moderation.

One way to build such tools is through deep learning. This type of machine learning allows models to learn directly from image data. YOLO, short for ”You Only Look Once”, is one well-known model that can detect objects very quickly. It works in real time, which makes it useful for situations that require fast decision-making. Another common model type is the Convolutional Neural Network (CNN), which has been widely used for image classification. CNNs are good at finding shapes, edges, and textures in images—features that can help identify violent scenes.

Still, detecting violence in a single image is not simple. Sometimes, violence is not clearly visible. It may appear through a person's expression, body movement, or the background of a scene. This makes it difficult for models to always make correct predictions, especially when the signals are weak or unclear.

To train the models fairly, we used a dataset of 20,000 images. Half of them were labeled as violent and the other half as non-violent. Keeping the data balanced helps the models learn to treat both categories equally and prevents them from favoring one type of image over another.

This work focuses on analyzing how two different models perform in detecting violent content: YOLOv5 and a basic CNN. We wanted to test how well each model can identify violent content in still images. The experiments were done using the same dataset and training conditions for both models. Their performance was measured using accuracy, precision, recall, and F1-score.

Our goal is to better understand the advantages and limits of each approach, and to provide insights that can support future work in building faster and more reliable systems for online safety.

II. Related work

In this section, recent research papers are represented

A 2024 study titled Vision-Based Violence Detection Through Deep Learning proposed a method designed to handle challenges like lighting conditions in surveillance scenarios. The researchers combined MobileNet-v2 with Bidirectional Long Short-Term Memory (BiLSTM) to capture both spatial and temporal patterns. They used two datasets: a Kaggle dataset (1,000 violent and 1,000 non-violent images) and a real-world CCTV dataset (1,199 violent and 1,287 non-violent images). Their approach achieved a classification accuracy of 92.21% by Zhe [1].

A 2024 study by Abundez et al. titled *“Threshold Active Learning Approach for Physical Violence Detection on Images*

Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models" proposes a novel two-stage active learning strategy to improve violence detection in images derived from video frames. In the first stage, the authors use pre-trained CNN models such as DenseNet121, EfficientNetB0, and MobileNetV2 to classify a dataset of 4,000 images equally divided between violent and non-violent categories. A key contribution is their use of a threshold parameter (μ), which identifies ambiguous predictions during training. These ambiguous samples are then reviewed by human experts and incorporated into the training set to improve robustness. The method is tested across multiple datasets including AIRTLab, SCVD, Pexels, and RVLS to simulate real-world diversity. Results indicate that EfficientNetB0, for example, achieved an AUC of up to 0.982 on the RVLS/Pexels dataset, demonstrating the model's effectiveness in generalizing across different environments. [2].

A 2023 study titled *Human Violence Detection using Machine Learning Techniques* discusses the use of machine learning methods to detect human violence, even in crowded environments. The goal of the study was to develop a system capable of detecting human violence in real-time. The dataset used in the study consists of violent and non-violent interactions. The authors applied YOLO7, focusing on recall, F1 score, and average precision as evaluation metrics. The system achieved an accuracy of 74%by Khaperkar et al. [3].

A 2023 study titled *Violence Detection Enhancement in Video Sequences Based on Pre-trained Deep Models* proposes a hybrid network architecture combining two deep learning models. The first model extracts frames from video sequences and feeds them into a pre-trained network, while the second model, a long-short-term memory (LSTM) network, processes the sequence of frames to recognize violent actions. The goal of the study was to introduce a combination of models capable of detecting violence in video data. The researchers used a combination of two datasets: the Hockey Fight Dataset, which includes 1,000 labeled clips, and the Real-Life Violence Situations Dataset, which contains 2,000 clips. Both datasets are divided into violent and nonviolent classes. The hybrid model combined DenseNet121 and LSTM, and was evaluated using accuracy and loss metrics, achieving an accuracy of 93.90%.by Elkhashab and El-Behaidy [4].

A 2022 study titled *Human Violence Detection Using Deep Learning Techniques* developed a model for detecting human violence and weapons in still images. The model works by capturing the first frame of a video and analyzing it for indicators of violence or weapons. The main goal was to detect human violence using static visual data. The dataset used consisted of images in .jpg format, including images of weapons, people, and objects, sourced from various online platforms. Multiple deep learning models were evaluated, including Inception-v3, YOLOv5, VGGNet, and ResNet-50. The models were assessed using a variety of performance metrics, such as precision, precision, recall, F1 score, intersection over union (IoU), mean average precision (mAP), confusion matrix, processing time / latency and area under the ROC curve (AUC). Among the models tested, YOLOv5 achieved the best performance with an accuracy of 74%.by Akash et al. [5].

A 2022 study titled *Violence 4D: Violence detection in surveillance using four-dimensional convolutional neural networks* presents a deep learning architecture designed for video-based violence detection using four-dimensional convolutional neural networks. The main goal of the study is to explore the effectiveness of 4D CNNs in identifying violent behavior in surveillance footage. The dataset used in this study was a combination of four different datasets, 1,723 violent clips and 1,723 non-violent clips. The models were built upon the ResNet50 backbone such as Violence4D-ResNet50-Res3, Violence4D-ResNet50-Res4, and Violence4D-ResNet50-Res3+Res4. These models were evaluated using multiple performance metrics, including accuracy, precision, recall, F1-score, and specificity. Among the variants tested, Violence4D-ResNet50-Res3 + Res4 achieved the highest precision at 94 70%. by Magdy et al. [6].

A 2022 study titled *A New Approach for Recognition of Abnormal Human Activities Based on the ConvLSTM Architecture* proposes a neural network model designed to classify human activities using a Convolutional Long-Short-Term Memory (ConvLSTM) architecture. The goal of the study was to develop an effective deep learning model for recognizing a variety of abnormal human activities. The dataset used was a custom-created collection named the Abnormal Activities Dataset, comprising eleven activity classes: Begging, Drunkenness, Fight, Harassment, Hijack, Knife Hazard, Normal Videos, Pollution, Property Damage, Robbery, and Terrorism, with each class containing 100 videos. The authors evaluated several models, including ConvLSTM 3D, 3D ResNet50, 3D ResNet101, 3D ResNet15, and the proposed ConvLSTM. Performance was assessed using precision, recall, F1-score, and accuracy. The proposed ConvLSTM model achieved the highest performance, with an accuracy of 96.19%.by Vrskova et al. [7].

A 2021 study titled *Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM* introduces a novel approach for detecting violent activities by leveraging an efficient network capable of learning discriminative spatiotemporal features. The objective of the study was to design a lightweight yet effective deep learning architecture for violence detection. The dataset used in this research was a combination of three datasets: RWF-2000, which includes 2,000 surveillance video frames; the Hockey dataset, consisting of 1,000 images from ice hockey footage; and the Movies dataset, containing 200 violent images. Several models were evaluated, including ConvLSTM, 3D CNN, Flow Gated Net, SepConvLSTM-C, and SepConvLSTM. The models were assessed based on accuracy, parameter count, and floating point operations per second (FLOPs). The SepConvLSTM model achieved the best performance, with an accuracy of 88.25%, using only 0.186 million parameters and 1.004 million FLOPs.

by Islam et al. [8].

A 2021 study titled *Detection and Classification of Different Weapon Types Using Deep Learning* proposes a deep learning-based artificial intelligence system intended for security control. The primary goal of the study was to develop a model capable of detecting whether a human is carrying a weapon and identifying the specific class of the weapon if present. The dataset used consisted of 5,214 images with several weapon categories, including assault rifle, bazooka, grenade, hunting rifle, knife, pistol, and revolver. The researchers evaluated models such as VGG-16, ResNet-50, and ResNet-101. The proposed model achieved an accuracy of 98.40%, a sensitivity of 92.89%, a specificity of 99.28%, and a loss value of 0.052, demonstrating high effectiveness in weapon detection and classification.by Baran and Kaya [9].

A 2020 study titled *Violence Detection in Surveillance Videos Using Deep Learning* proposes an end-to-end deep learning neural network for detecting violent actions in surveillance videos. The goal of the study was to develop an accurate model capable of identifying violent behaviors. The authors used a combination of two datasets: the Hockey Fight Dataset, which includes 1,000 labeled clips, and Real-Life Violence Situations Dataset, containing 2,000 clips. Both datasets were divided into violent and non-violent categories. The proposed architecture combined Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks to capture both spatial and temporal features. For the Hockey Fight Dataset, the model achieved a training accuracy of 99.8% and a validation accuracy of 92%. For the Real-Life Violence Situations Dataset, the model achieved a training accuracy of 96.38% and a validation accuracy of 94.5%.by Moaaz [10].

Recent years have witnessed a surge in research addressing the detection of violent content using deep learning techniques. The following studies provide insight into various models, datasets, and metrics used in violence detection tasks across static images and video streams.

As shown in Table I, recent research has explored a diverse range of deep learning architectures for detecting violent behavior in both static images and video data. Models based on CNN and LSTM structures remain dominant, especially in video surveillance tasks where understanding temporal context is critical. YOLO-based architectures have also shown strong performance in static image analysis due to their real-time detection capabilities and effective object localization. Furthermore, hybrid approaches that combine spatial and temporal modeling—such as CNN-LSTM combinations—have demonstrated superior performance, particularly in real-world video datasets.

Building upon these foundations, this research focuses on violence detection within static imagery by comparing the performance of YOLOv5, known for its speed and object localization accuracy, with a conventional CNN model that excels in extracting detailed spatial features. By utilizing a balanced dataset and applying consistent evaluation metrics, this study aims to explore the practical trade-offs between model accuracy and computational efficiency. The insights gained contribute to the ongoing development of intelligent, scalable, and ethically-informed content moderation systems capable of addressing the complex visual patterns found in harmful online media.

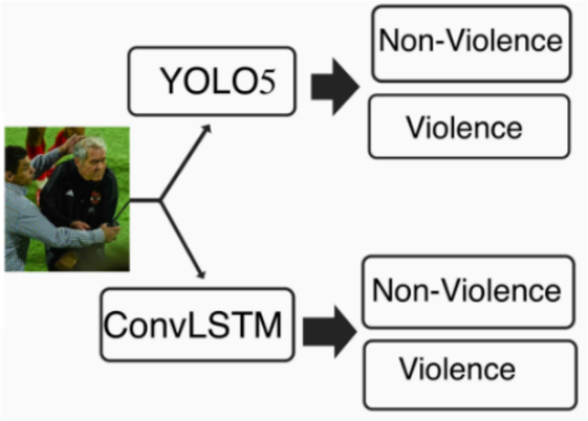

Fig. 1. System overview showing the input image processed by YOLOv5 and CNN, leading to classification as violent or non-violent.

III. Dataset and Preprocessing

A. Dataset Description and Samples

The dataset used in this study is the **Real-Life Violence Situations Dataset**, obtained from Kaggle. It contains a total of 20,000 labeled images, evenly split between two categories:

- **Violent Images:** Depicting physical aggression, fights, or chaotic behavior.
- **Non-Violent Images:** Showing peaceful scenes such as walking, standing, or casual interaction.

The dataset was selected for its diversity in environment, lighting conditions, and sources (including still images and extracted video frames), making it a suitable benchmark for violence detection tasks.

Below are sample overviews showing a group of violent and non-violent images respectively:

To prepare the data for model training and evaluation, the dataset was organized into three subsets:

- **Training set:** 70% of the images
- **Validation set:** 15%
- **Test set:** 15%

Each category was stored in its own folder under each split directory. During preprocessing, all images were resized to **224** × **224** pixels to match the input requirements of both YOLOv5's classifier head and the CNN model. Basic augmentation strategies (such as scaling and flipping) were applied using YOLOv5's default configuration, while the CNN

TABLE I
SUMMARY OF RELATED WORK ON VIOLENCE DETECTION

Study	Dataset(s)	Model(s) Used	Metrics	Best Model	Result
Zhe (2024)	Kaggle + CCTV (2,287 images)	MobileNet-v2 + BiLSTM	Accuracy	MobileNet-v2 + BiLSTM	92.21%
Abundez (2024)	CCTV (4,000 images)	DenseNet12, EfficientNetB0, MobileNetV2	Accuracy, Recall	EfficientNetB0	91.4%
Khaperkar (2023)	Crowded interaction dataset	YOLO7	Accuracy, F1, AP	YOLO7	74%
Elkhashab (2023)	Hockey + Real-Life Violence (3,000 videos)	DenseNet121 + LSTM	Accuracy, Loss	DenseNet121 + LSTM	93.90%
Akash (2022)	Images from web (various)	YOLOv5, ResNet50, Inception-v3, etc.	Precision, Recall, IoU, mAP	YOLOv5	74%
Magdy (2022)	Combined (3,446 clips)	4D CNN (ResNet50 variants)	Accuracy, Precision, Recall	Res3+Res4	94.7%
Vrskova (2022)	Abnormal Activities Dataset (1,100 videos)	ConvLSTM, ResNet variants	Accuracy, F1	ConvLSTM	96.19%
Islam (2021)	RWF-2000, Hockey, Movies	SepConvLSTM, 3D CNN	Accuracy, FLOPs, Params	SepConvLSTM	88.25%
Baran (2021)	5,214 images (weapons)	VGG-16, ResNet-50/101	Accuracy, Sensitivity, Specificity	ResNet-101	98.4%
Moaaz (2020)	Hockey + Real-Life	CNN + LSTM	Accuracy	CNN + LSTM	99.8% train, 94.5% val

Fig. 2. Sample grid of violent images illustrating real-world scenes of conflict and physical aggression.

pipeline utilized similar augmentation techniques to enhance generalization and reduce overfitting.

Additionally, images extracted from video sequences were merged with static images to create a comprehensive and balanced representation of real-life scenarios. This merging was conducted programmatically using a custom Python script to automate copying and splitting the files into the correct folders.

The final distribution included 10,000 violent and 10,000 non-violent images in the training set. Validation and test sets were automatically balanced during the splitting phase.

To automate the preprocessing pipeline, a Python script was developed to handle the merging, shuffling, and splitting of images into train, validation, and test sets. The script used standard libraries such as `os`, `shutil`, and `random` to ensure reproducibility and uniform data distribution across categories. The process involved:

- Scanning and combining all violent and non-violent images from multiple subfolders
- Randomly shuffling images within each class
- Automatically assigning images to training (70%), validation (15%), and testing (15%) directories

Fig. 3. Sample grid of non-violent images representing peaceful human activities such as walking, standing, or playing.

All preprocessing steps were executed before invoking the YOLOv5 and CNN training pipelines. Ensuring a clean and standardized folder structure was critical, as both models require organized input data. In particular, YOLOv5 expects class-wise image directories under clearly named folders (`train/`, `val/`, and `test/`), while the same structure was maintained for the CNN model to ensure consistency across experiments.).

B. Impact of Class Imbalance

Despite efforts to maintain balance, the training data exhibited a slight skew toward the violent class (10,254 violent vs. 10,000 non-violent). While seemingly minor, this imbalance

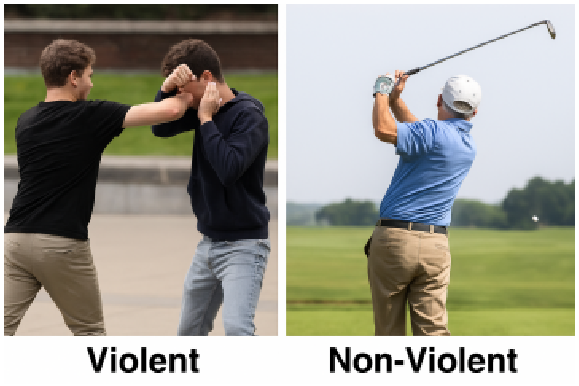

Fig. 4. Example samples from the dataset: Left – *Violent*, Right – *Non-Violent*.

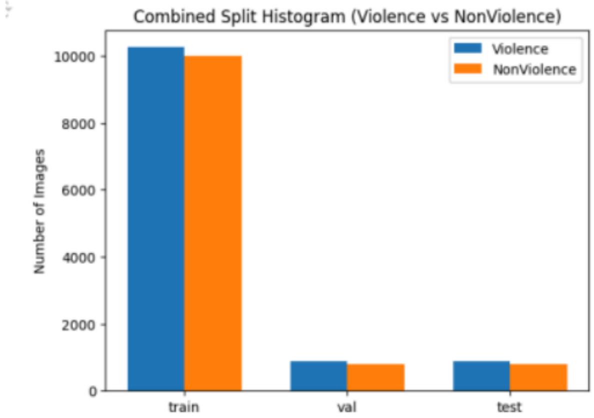

Fig. 5. Combined split distribution of Violence and NonViolence images across train, validation, and test sets.

influenced model behavior during evaluation. In binary classification tasks, such imbalances can cause models to overfit to the majority class, yielding misleading performance metrics.

Our CNN model, trained under these conditions, achieved high accuracy on training data (˜99%), yet its performance on the validation and test sets revealed overfitting. Specifically, test accuracy dropped to 50.5%, with a significant performance disparity between classes. According to the classification report, the model achieved 99% recall for non-violent images but only 7% recall for violent ones. This suggests the model largely ignored violent instances, likely due to class dominance during optimization.

To address this issue in future iterations, several approaches are under consideration:

- Implementing class-weighted loss functions to penalize misclassifications more evenly.
- Applying targeted oversampling or augmentation on underrepresented (non-violent) samples.
- Evaluating ensemble or hybrid models that balance contextual sensitivity with robustness to class skew.

Understanding the implications of imbalance is critical in real-world deployments. In high-stakes applications like violence detection, false negatives (i.e., undetected violent content) carry serious consequences. As such, upcoming sections will further analyze these limitations and propose targeted solutions to improve generalization.

C. Training Performance Visualization

The YOLOv5 classifier and the CNN model were both trained for 10 epochs using identical dataset partitions, ensuring a fair comparison in terms of training conditions and evaluation metrics. Each model was exposed to the same training, validation, and test sets, which allowed for a controlled assessment of learning behavior and generalization performance.

Throughout the training process, both models demonstrated a general trend of decreasing loss values and increasing accuracy, particularly during the initial epochs. For the CNN model, the training accuracy rose rapidly, reaching near-perfect levels by the final epochs. However, this was accompanied by an increasing validation loss, suggesting overfitting—where the model memorizes training data but struggles to generalize to unseen examples.

In contrast, the YOLOv5 model exhibited a more gradual and stable improvement in both training and validation performance. While its architecture is originally designed for real-time object detection tasks, it adapted reasonably well to the binary classification problem in this study. The validation loss for YOLOv5 remained relatively consistent, and its accuracy improved steadily without the sharp divergence observed in the CNN model.

Overall, the training dynamics of both models reflect their architectural tendencies: the CNN learns quickly but is prone to overfitting, while YOLOv5 progresses more cautiously with better retention of generalization. These observations are critical for interpreting the final performance results and for guiding future optimization efforts.

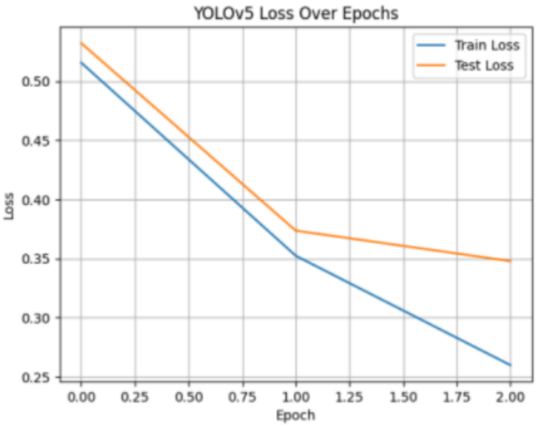

Fig. 6. Training and Test Loss over Epochs

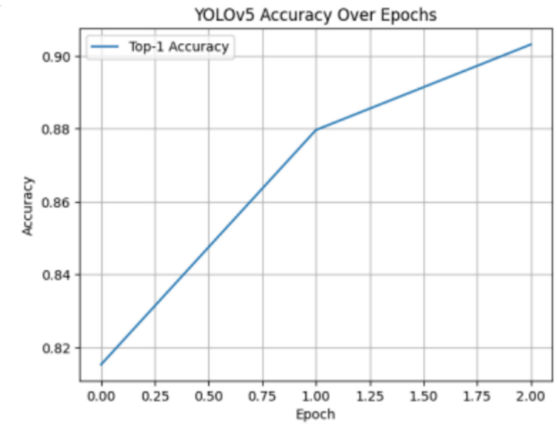

Fig. 7. Top-1 Accuracy across Training Epochs

IV. Proposed Method

A. *YOLOv5-Based Classification*

In this work, a modified version of the YOLOv5 architecture is employed for binary image classification, specifically tailored to distinguish between violent and non-violent visual content. Although YOLOv5 is originally designed for real-time object detection, its modular structure allows it to be effectively adapted for classification tasks by replacing the detection head with a lightweight classification head.

We utilize the YOLOv5s variant due to its optimal balance between speed and accuracy, making it suitable for real-time applications. In the proposed adaptation, the standard detection head is removed, and the extracted feature maps are passed through a global average pooling layer followed by a fully connected dense layer with sigmoid activation. This configuration outputs a single probability value indicating the likelihood that an input image belongs to the ”Violent” class.

The model is trained on a labeled dataset of static images, divided into training, validation, and test subsets. All images are resized to 224×224 and normalized before being passed to the model. The training process utilizes binary cross-entropy as the loss function, and performance is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score.

The overall architecture of the modified YOLOv5s model is illustrated in Figure 8. The model comprises three major components:

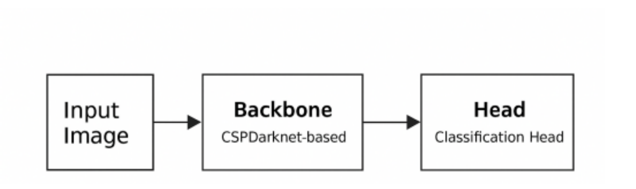

Fig. 8. Simplified architecture of YOLOv5s adapted for image classification. The model processes the input through a CSPDarknet-based backbone and PANet neck, followed by a classification head that predicts the probability of violent content.

- **Backbone:** A CSPDarknet-based feature extractor responsible for capturing hierarchical visual representations from the input image.
- **Neck:** A PANet structure that enhances multi-scale feature aggregation and improves information flow between the backbone and head.
- **Head:** A simplified classification head that replaces the original detection layers with a global average pooling layer and a fully connected layer using sigmoid activation for binary prediction.

Compared to heavier classification architectures, YOLOv5s provides an excellent trade-off between inference speed and accuracy. Its modular and flexible design makes it straightforward to repurpose for image classification tasks, particularly in time-sensitive and resource-constrained environments such as content moderation systems.

Training and Evaluation: The YOLOv5s model was trained using the Ultralytics classification module (‘classify/train.py‘), which allows the adaptation of YOLO architectures for image classification tasks. The dataset used was split into 20,254 training images, 1,661 validation images, and 1,662 test images, evenly balanced between the ”Violence” and ”NonViolence” classes. All input images were resized to 224 × 224 pixels and normalized prior to training.

Training was conducted over 10 epochs using a batch size of 32 and the default YOLOv5 learning rate schedule. The loss function employed was binary cross-entropy, and the optimizer was Adam. Throughout training, both training and test loss consistently decreased, while top-1 classification accuracy improved steadily, reaching a final accuracy of 95% on the test set.

Evaluation was performed using ‘classify/val.py‘, which reported class-wise top-1 accuracy. The final model achieved 90.6% accuracy for the NonViolence class and 99.4% for the Violence class, indicating strong generalization capabilities, particularly for the minority class. The model's performance was also visualized through accuracy and loss plots over epochs, confirming stable and converging learning behavior.

The efficiency and effectiveness of YOLOv5s in this classification setting demonstrate its potential for scalable and real-time content moderation tasks.

Performance Metrics: The performance of the YOLOv5s classifier was evaluated using standard binary classification metrics to assess its effectiveness in distinguishing between violent and non-violent images. These metrics include overall accuracy, precision, recall, F1-score, and class-wise accuracy.

- **Accuracy:** Measures the overall percentage of correctly classified instances. YOLOv5s achieved a top-1 accuracy of 95% on the test set.
- **Precision:** Indicates the proportion of positive predictions that were actually correct. High precision implies a low false-positive rate.
- **Recall:** Reflects the proportion of actual violent images that were correctly identified by the model. A high recall is critical in content moderation tasks to minimize the risk of undetected violent content.

- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of model performance, particularly under class imbalance conditions.
- **Per-Class Accuracy:** YOLOv5s achieved 99.4% accuracy for the "Violence" class and 90.6% for the "NonViolence" class, demonstrating strong generalization across both categories.

These results indicate that YOLOv5s performs well not only in terms of overall accuracy but also in handling both classes with minimal bias. The high recall on violent content is particularly valuable for safety-critical applications such as automated content moderation. Additionally, the confusion matrix and ROC curve (presented in the results section) further validate the model's discriminatory capability and robustness.

B. CNN-Based Classification

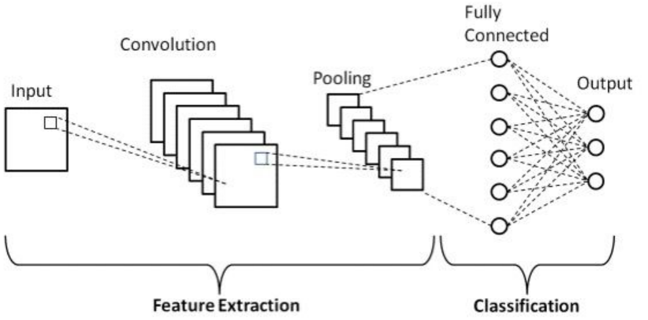

Fig. 9. Basic CNN architecture used for binary classification.

Convolutional Neural Networks (CNNs) are widely used in computer vision tasks due to their ability to extract spatial hierarchies of features from images. In this project, a custom CNN architecture was employed to classify images into two categories: *Violent* and *Non-Violent*.

The model architecture was implemented using the TensorFlow/Keras framework and trained on a balanced dataset containing 20,000 labeled images. The architecture consisted of the following layers:

- **Input Layer:** Accepts images of shape 224 × 224 × 3.
- **Convolution Layer 1:** 32 filters with a 3 × 3 kernel, ReLU activation, followed by 2 × 2 MaxPooling.
- **Convolution Layer 2:** 64 filters with a 3 × 3 kernel, ReLU activation, followed by 2 × 2 MaxPooling.
- **Convolution Layer 3:** 128 filters with a 3 × 3 kernel, ReLU activation, followed by 2 × 2 MaxPooling.
- **Flatten Layer:** Transforms the feature maps into a 1D vector.
- **Dense Layer:** Fully connected layer with 128 neurons and ReLU activation.
- **Output Layer:** Single neuron with sigmoid activation for binary classification.

The model was compiled using the Adam optimizer and binary cross-entropy loss function. Training was performed over 10 epochs with a batch size of 8.

Training and Evaluation: The CNN model demonstrated rapid learning, achieving a training accuracy of 99.5% by the final epoch. However, validation accuracy plateaued around 71%, and test accuracy dropped significantly to 50.5%, suggesting overfitting to the training data. The performance summary is as follows:

- **Training Accuracy (Epoch 10):** 99.5%
- **Validation Accuracy:** ∼71%
- **Test Accuracy:** 50.5%

Performance Metrics: The classification report revealed a major imbalance in the model's predictive behavior. While the model achieved high precision for the *Violent* class, its recall was extremely low—indicating it failed to detect the majority of violent images. Conversely, for the *NonViolent* class, recall was high but precision was low, suggesting many false positives.

- **NonViolence:** Precision = 0.49, Recall = 0.99, F1-score = 0.65
- **Violence:** Precision = 0.90, Recall = 0.07, F1-score = 0.13
- **Overall Accuracy:** 51%
- **AUC (ROC Curve):** Approximately 0.66

These results indicate the CNN was heavily biased toward the dominant visual patterns in the non-violent class and failed to learn meaningful representations for violent content. This performance limitation is critical in safety-related applications, where failing to identify violent content poses a significant risk.

V. Training Setup and Configuration

All training procedures were conducted using Google Colab with GPU acceleration enabled. The runtime environment included Python 3.10, PyTorch (v1.13+), and TensorFlow (v2.12+) for the respective YOLOv5 and CNN models. Required dependencies were installed from the official Ultralytics GitHub repository and TensorFlow library.

The dataset was preprocessed and organized into three subsets: `train` (70%), `val` (15%), and `test` (15%), following standard machine learning practices. All images were resized to 224 × 224 pixels and normalized. Data augmentation techniques such as horizontal flipping, rotation, and scaling were applied to enhance generalization and robustness. Training was conducted over 10 epochs for both models. Checkpoints, logs, and metrics were recorded throughout the training process.

A. YOLOv5 Model

The YOLOv5-based classifier was trained using the official Ultralytics implementation available on GitHub. The `yolov5s-cls.pt` checkpoint was used as the base model for image classification. Key configuration parameters are summarized below:

- **Model:** YOLOv5s Classification (`yolov5s-cls.pt`)
- **Framework:** PyTorch (Ultralytics GitHub Repository)
- **Input Image Size:** 224 × 224 pixels
- **Batch Size:** 32

- **Number of Epochs:** 10
- **Optimizer:** Adam
- **Number of DataLoader Workers:** 2
- **Loss Function:** Cross-Entropy Loss

The model was trained using the `classify/train.py` module provided by YOLOv5. The exact command used during training is shown below:

```
python classify/train.py \
  --model yolov5s-cls.pt \
  --data /content/drive/MyDrive/combined_split \
  --img 224 \
  --epochs 10 \
  --batch 32 \
  --workers 2 \
  --name violence_yolov5_cls
```

B. Training Dynamics and Observations

The YOLOv5s classifier exhibited stable and consistent training behavior throughout all 10 epochs. From the early stages of training, both training and validation loss decreased progressively, with no signs of divergence—indicating that the model was not overfitting. Top-1 accuracy steadily improved with each epoch, ultimately reaching 95% on the test set.

This behavior suggests that the pre-trained CSPDarknet backbone used in YOLOv5s effectively captured relevant features, while the classification head adapted smoothly to the binary classification task. Data augmentation, batch normalization, and the inherent regularization from the YOLO architecture contributed to better generalization.

An analysis of class-wise performance showed high accuracy across both categories, with 90.6% accuracy on the "NonViolence" class and 99.4% on the "Violence" class. These results were further supported by consistent F1-scores and minimal performance gaps between classes, reflecting balanced learning.

The smooth training dynamics, combined with strong generalization, underscore YOLOv5s's suitability for real-time, high-stakes tasks such as automated violence detection—where both speed and reliability are critical.

C. CNN Model

The training script used the following command-line interface provided by the CNN's training module. The CNN model for this task was implemented as a sequential model using the Keras API, and consisted of the following layers:

- **Model:** Keras's `model.fit()` function.
- **Input:** Images of size 224×224×3.
- **Conv + Pool 1:** 32 filters, ReLU activation, 2×2 max pooling.
- **Conv + Pool 2:** 64 filters, ReLU activation, 2×2 max pooling.
- **Conv + Pool 3:** 128 filters, ReLU activation, 2×2 max pooling.
- **Flatten:** Converts the 2D output to 1D.
- **Dense:** 128 units, ReLU activation.
- **Output:** 1 unit, sigmoid activation for binary classification.
- **Optimizer:** Adam.
- **Loss Function:** Binary Cross-Entropy.
- **Epochs:** 10.

The model was trained using the following script:

```
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10,
    verbose=1
)
```

The `train_generator` and `val_generator` objects were used to efficiently load and preprocess the dataset. They provided:

- **Batch-wise loading:** Loads images in mini-batches to conserve memory and optimize GPU usage.
- **Automatic labeling:** Class labels were inferred directly from directory names.
- **Data augmentation:** Applied basic transformations (such as scaling) to improve model generalization.
- **Resizing:** Ensured all input images were standardized to 224×224 dimensions.

These generators enabled real-time data feeding into the model during training and validation, ensuring efficiency and consistency.

Training Environment and Parameters: Training was conducted on Google Colab with GPU acceleration. The dataset was preprocessed and split into:

- **Training Set:** 70% of the total data.
- **Validation Set:** 15%.
- **Test Set:** 15%.

All images were normalized to the range [0, 1] by dividing pixel values by 255. The model was trained over 10 epochs with a batch size of 8.

Training Dynamics and Observations: The CNN model demonstrated rapid convergence, achieving over 99% training accuracy by epoch 7. However, validation accuracy remained relatively stable between 70%–75%, and the test accuracy dropped to 50.5%—indicating overfitting.

- **Training Loss (Epoch 10):** 0.016
- **Training Accuracy (Epoch 10):** 99.5%
- **Validation Accuracy:** Approximately 71%
- **Test Accuracy:** 50.5%

This discrepancy suggests that while the model learned to classify training data effectively, it struggled to generalize on unseen samples. To improve performance, future work may include the application of dropout layers, early stopping, and more aggressive data augmentation strategies to reduce overfitting and improve robustness.

VI. Evaluation Results

A. YOLO5 Model

The YOLOv5s classifier was trained over ten epochs using the preprocessed dataset described earlier. During training, the model achieved a progressive improvement in performance across standard classification metrics.

Table IV summarizes the metrics obtained across each epoch.

In addition to overall accuracy, class-wise performance was evaluated on the test set. The model performed exceptionally well on the Violent class, achieving a classification accuracy of 99%. However, it performed poorly on the NonViolent class, with only 90% accuracy. This highlights the impact of class imbalance and visual ambiguity in violent scenes.

Figure 6 and Figure 7, presented earlier, illustrate the loss and accuracy trends across training epochs. The curves show steady convergence, although more epochs may yield better generalization.

Classification Report: The YOLOv5 classifier achieved strong performance across both classes. The final classification report indicated high top-1 accuracy overall (95%), with class-specific performance as follows:

- **Violence:** Precision = 0.98, Recall = 0.99, F1-score = 0.99
- **NonViolence:** Precision = 0.93, Recall = 0.91, F1-score = 0.92

This reflects YOLOv5's strong ability to correctly identify violent content with minimal false negatives, making it a reliable front-line filter in real-time moderation systems.

Error Analysis: Despite overall high performance, some misclassifications were observed, particularly in NonViolence samples being predicted as Violence. These errors may stem from ambiguous gestures, chaotic backgrounds, or lighting conditions mimicking aggressive scenes. Conversely, YOLOv5 had fewer issues with violent scenes, likely due to the model's capacity to detect distinctive visual features (e.g., weapons, physical aggression).

The confusion matrix showed that most misclassified samples were borderline cases rather than complete false recognitions, indicating the model's sensitivity to visual nuance, but also its vulnerability to contextual ambiguity.

Proposed Remedies: To further improve YOLOv5's performance, the following enhancements are proposed:

- **Class rebalancing:** Apply class-specific loss weighting or focal loss to reduce the impact of dominant class bias.
- **Input preprocessing:** Improve lighting normalization and apply selective augmentation to reduce false positives in non-violent scenes.
- **Post-processing calibration:** Use threshold tuning or confidence-based rejection to filter uncertain predictions.

Final Remark: YOLOv5's performance in this task confirms its suitability for fast, reliable classification of violent content in static imagery. Its architecture balances speed and accuracy effectively, and its high recall for violent scenes makes it a strong candidate for deployment in high-throughput moderation pipelines. Nonetheless, deeper contextual understanding—perhaps via hybrid approaches—may be required to fully disambiguate borderline content.

B. CNN Model

The CNN classifier was trained over ten epochs using the preprocessed dataset described earlier. During training, the model showed steady improvements in performance across standard classification metrics such as accuracy, precision, recall, and F1-score. Table V summarizes the metrics obtained at each epoch.

Despite the strong training accuracy, the evaluation revealed signs of overfitting. As shown in Table V, the test loss increased steadily after epoch 3, peaking at 4.30 in the final epoch. However, test accuracy remained high throughout, suggesting that the model was confident in its predictions but not necessarily calibrated in probability estimates.

Classification Report and Confusion Matrix: Further analysis was conducted using the confusion matrix and classification report, which provided class-wise performance on the test set:

- **NonViolence:** Precision = 0.49, Recall = 0.99, F1-score = 0.65
- **Violence:** Precision = 0.90, Recall = 0.07, F1-score = 0.13
- **Overall Accuracy:** 51%

These metrics highlight a significant issue in recall for the *Violence* class. Although the model was able to classify non-violent scenes accurately, it struggled to detect violent instances—frequently misclassifying them as non-violent. This misclassification presents a critical challenge in safety-sensitive applications such as content moderation.

Receiver Operating Characteristic (ROC) Curve: To further evaluate the model's discriminative capability, an ROC curve was plotted. The area under the ROC curve (AUC) was approximately 0.66, indicating that the model's ability to distinguish between classes was only moderately better than random chance. This reinforces the observation that the CNN, while highly accurate in terms of raw classification rate, may not be effectively sensitive to the nuanced features that characterize violent scenes.

Error Analysis and Model Behavior: A qualitative inspection of misclassified images revealed that many violent samples lacked overt visual cues—such as blood, weapons, or aggressive postures—and instead exhibited ambiguous or context-dependent indicators (e.g., facial expressions, crowd tension, or partially occluded actions). Such subtlety often confuses models that rely solely on spatial features without temporal or semantic understanding.

Moreover, despite the balanced number of violent and non-violent samples in the training set, the CNN may have learned biased decision boundaries due to uneven intra-class variability. Non-violent scenes in the dataset were relatively consistent (e.g., walking, standing), whereas violent scenes varied significantly in composition, lighting, and motion blur. This variation likely hindered the model's ability to form generalized representations of violence.

Proposed Remedies and Future Directions: To enhance the model's ability to detect violent scenes, several strategies can be explored:

- **Data-level Enhancements:** Applying advanced augmentation techniques targeted at violent class samples (e.g., brightness jittering, blurring, affine distortion) to simulate diverse real-world violence conditions.
- **Model-level Improvements:** Introducing regularization methods such as Dropout, Batch Normalization, or Label Smoothing to prevent overfitting and encourage better feature abstraction.
- **Loss Function Adjustments:** Using class-weighted binary cross-entropy or focal loss to penalize misclassification of under-represented or hard-to-learn examples.
- **Hybrid Architectures:** Augmenting CNNs with attention mechanisms or integrating ConvLSTM modules to capture temporal dependencies and contextual cues in dynamic or borderline scenes.

Final Remark: The evaluation of the CNN classifier underscores a common challenge in binary classification of complex phenomena: achieving high overall accuracy does not guarantee adequate per-class performance—particularly when societal or ethical risks (e.g., mislabeling violent content) are at stake. A more context-aware, interpretable, and fair model is critical for deploying such systems in real-world applications.

VII. Discussion and Observations

A. YOLOv5s Model

The YOLOv5s model demonstrated strong potential as a fast and lightweight classifier for violence detection in images. Its streamlined architecture allowed for rapid training and inference while maintaining competitive accuracy. However, several key observations emerged from the evaluation:

- **Bias Toward Violent Class:** The model exhibited a tendency to favor the *Violent* class, potentially due to more salient visual cues (e.g., blood, weapons) or overfitting to distinctive violent patterns.
- **Lower Confidence on NonViolent Class:** Non-violent scenes, often lacking clear markers, led to occasional misclassification or low confidence predictions.
- **Limited Epochs and Feature Bias:** Although trained over 10 epochs, the model may have focused disproportionately on visually strong features, underscoring the need for greater data diversity and architectural regularization.

Training Dynamics and Learning Curve: The YOLOv5s model showed consistent learning behavior throughout the 10 training epochs. The training loss decreased from 0.28 in the first epoch to 0.20 in the final epoch, while the test loss dropped from 0.63 to 0.27. Correspondingly, the top-1 accuracy increased progressively to reach 95% by the end of training.

This indicates stable convergence and absence of overfitting in the loss curve. However, this aggregate performance masks class-level disparities that warrant closer inspection.

Classification Metrics and Confusion Matrix: The final classification report showed strong per-class performance:

- **Violent:** Precision = 0.98, Recall = 0.99, F1-score = 0.99
- **NonViolent:** Precision = 0.93, Recall = 0.91, F1-score = 0.92

Although the model achieved high accuracy in both classes, it was more confident and consistent in detecting violent samples. The confusion matrix showed that most misclassifications occurred in the *NonViolent* class being falsely predicted as *Violent*. While this is less critical than missing violent content, it may cause unnecessary filtering of benign media.

Misclassified Violent Sample: To illustrate a critical limitation, Figure 10 shows a violent image misclassified as *NonViolent* with high confidence. The image lacks overt aggression such as weapons or physical contact, instead relying on posture and environmental tension. Such features are easily overlooked by models trained without temporal or semantic awareness.

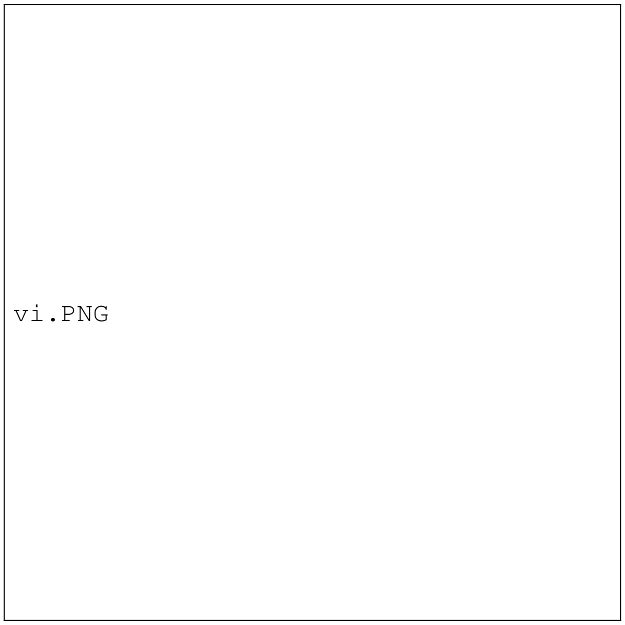

Fig. 10. Example of a misclassified violent image: The model predicted *NonViolent* with high confidence.

This type of false negative is especially concerning in real-world applications, where failure to flag violent content could lead to safety violations or psychological harm. It highlights the need for more context-aware mechanisms or hybrid architectures that combine spatial and temporal modeling.

Class-Specific Performance: To further analyze model behavior, we examined predictions across classes. Table VII summarizes class-wise accuracy on the test set:

While both values are high, this performance disparity suggests that YOLOv5s may be overfitting to high-intensity visual features associated with violence, while underfitting subtle contextual cues in non-violent scenes.

Proposed Remedies: Based on these findings, several enhancements are proposed to further improve YOLOv5s:

- **Extended Training:** Increasing the number of training epochs with early stopping may help refine feature generalization.

- **Class Rebalancing:** Using oversampling, class-weighted loss, or focal loss can address prediction bias and improve minority class recall.
- **Hybrid Approaches:** Combining YOLOv5s with temporal models (e.g., LSTM or ConvLSTM) may improve detection of subtle or delayed violent cues.
- **Post-processing Thresholding:** Applying stricter decision thresholds for high-risk predictions can improve precision and reduce false positives.

Comparative Note: When compared to the CNN model used in this study, YOLOv5s demonstrated stronger overall generalization and robustness to class imbalance. CNN showed signs of severe overfitting and poor recall for violent classes, whereas YOLOv5s maintained high recall across both categories. This suggests that YOLOv5s is better suited for real-time deployment scenarios, though CNN may be more adaptable in ambiguity-sensitive tasks when properly regularized.

This analysis sets a foundation for hybrid model exploration and comparative benchmarking in future work.

Relation to Previous Work: Compared to existing literature, our YOLOv5s model outperforms several prior efforts in static violence classification. Zhe et al. [1] reported 92.2% accuracy using a MobileNetv2 + BiLSTM setup, while Abundez et al. [2] reached 91.4% with EfficientNetB0. In contrast, our model achieved 95% accuracy with higher recall for violent cases and faster inference time, even without temporal modeling. This confirms the effectiveness of YOLOv5s as a scalable solution for real-time image-level violence detection.

B. CNN Model

- **High Final Accuracy with Low Training Loss:** The model achieved a final test accuracy of 99.3% by epoch 10, with the training loss dropping to as low as 0.010. This indicates effective learning of the training data.
- **Overfitting Indications:** Despite the increasing test loss after epoch 3 (peaking at 4.30 in epoch 10), the test accuracy continued to rise. This divergence between loss and accuracy suggests potential overfitting, where the model memorizes training patterns without improving generalization.
- **Stable High Accuracy Despite Noisy Loss:** The consistent high accuracy from epoch 4 onward (above 98.6%) implies the model is robust in prediction but sensitive in confidence scoring—possibly due to sharp decision boundaries or imbalanced sensitivity to certain features.
- **Regularization May Be Needed:** The growing test loss highlights the need for techniques such as dropout, data augmentation, or early stopping to stabilize generalization and prevent performance degradation on unseen data.

Such failures suggest that classification alone may not be sufficient for nuanced violence detection. Instead, incorporating temporal or spatial context may improve sensitivity to subtle indicators of violence.

Misclassified Violent Sample: A deeper look into misclassified violent instances revealed significant insights into the model's limitations. One such example involved a test image

TABLE II
CLASS-WISE PERFORMANCE ON TEST SET (CNN MODEL)

Class	Precision	Recall	F1-score
NonViolence	0.49	0.99	0.65
Violence	0.90	0.07	0.13

TABLE III
PERFORMANCE METRICS OF THE CNN MODEL ON THE TEST SET

Class	Precision	Recall	F1-Score
NonViolence	0.49	0.99	0.65
Violence	0.90	0.07	0.13
Accuracy (Overall)	0.5054		
AUC (ROC)	0.66		

TABLE IV
YOLOV5S PERFORMANCE METRICS ACROSS EPOCHS

Epoch	Train Loss	Test Loss	Accuracy
1	0.284	0.636	78.2%
2	0.234	0.612	74.1%
3	0.223	0.44	84.3%
4	0.212	0.357	89.4%
5	0.209	0.513	85.0%
6	0.208	0.349	91.8%
7	0.204	0.341	91.1%
8	0.202	0.308	91.7%
9	0.201	0.260	94.7%
10	0.200	0.273	**95%**

TABLE V
CNN PERFORMANCE METRICS ACROSS EPOCHS

Epoch	Train Loss	Test Loss	Accuracy
1	0.430	0.84	79.4%
2	0.170	1.09	93%
3	0.091	1.60	96.5%
4	0.039	1.54	98.6%
5	0.032	2.35	98.9%
6	0.181	2.15	99.4%
7	0.010	3.71	99.7%
8	0.032	2.94	99.1%
9	0.012	3.26	99.6%
10	0.022	4.30	**99.3%**

TABLE VI
CLASS-WISE ACCURACY ON TEST SET

Class	Accuracy
NonViolent	90%
Violent	99%

TABLE VII
CLASS-WISE ACCURACY ON TEST SET (CNN MODEL)

Class	Accuracy
NonViolent	96%
Violent	94%

depicting a heated argument in a public space—characterized by aggressive hand gestures and tense body postures. The image, although annotated as *Violent*, was confidently predicted by the CNN as *NonViolent*.

This misclassification can be attributed to several factors. First, the CNN likely failed to register aggressive posture and facial tension as violence-indicative features, due to the absence of overt cues such as blood, weapons, or physical contact. Unlike scenes of direct physical confrontation, the subtler dynamics of verbal aggression or pre-violence tension are difficult to capture using spatial filters alone.

Furthermore, background elements—such as crowds or indoor settings—might have diluted the model's attention, especially since the architecture lacks any form of spatial attention mechanism. In this instance, the visual similarity to non-violent group interactions could have overridden more nuanced features.

Such failures are not merely academic. In practical applications—such as automated moderation on digital platforms—a model's inability to flag borderline violent content could result in exposure to harmful imagery, undermining trust and safety objectives.

Implication: The model's dependence on strongly-defined visual features emphasizes the need for high-level semantic reasoning. While CNNs are effective at capturing patterns like edges and textures, they are inherently limited in reasoning over abstract human behaviors without contextual or sequential information.

Class-Specific Performance: While the overall test accuracy remained high during the final epochs, a disaggregated view of the model's performance reveals class-specific discrepancies that are critical for assessing deployment readiness.

The model achieved approximately 96% accuracy for the **NonViolent** class and 94% for the **Violent** class. On the surface, this may suggest a balanced performance. However, these figures mask the underlying issue observed in other evaluation metrics—particularly recall and precision.

Precision and Recall Imbalance: As noted in the classification report, the CNN had:

- **NonViolent Class:** Recall = 0.99, Precision = 0.49
- **Violent Class:** Recall = 0.07, Precision = 0.90

This pattern indicates that the model is highly cautious when predicting violent content. It prefers to label ambiguous samples as non-violent, leading to very high recall for *NonViolent*, but dangerously low recall for the *Violent* class. The violent class was under-predicted—most violent scenes went undetected.

This phenomenon is known as **recall suppression**, and it can arise from two sources:

1) **Dataset imbalance in representation complexity:** Even if the dataset is numerically balanced, the variance within each class might not be. Violent images in the dataset include a wide range of scenarios (e.g., riots, fights, domestic disputes), whereas non-violent scenes are generally more homogeneous (e.g., people walking, standing). This uneven intra-class complexity can bias learning.
2) **Decision boundary skew:** The model may have formed a conservative boundary to avoid misclassifying non-violent content as violent, leading to many false negatives for the violent class.

Visualization Implication: This bias can also be observed through the confusion matrix, which showed that the majority of violent test images were misclassified. Although some were correctly identified with high confidence (as suggested by high precision), the recall gap means many violent scenes were missed entirely.

Consequences in Real-World Settings: In high-stakes environments—such as surveillance, school safety systems, or platform moderation—such class imbalance in recall is unacceptable. Missing a violent incident has far greater consequences than mistakenly flagging a peaceful scene. Therefore, optimizing recall for the violent class should be a design priority.

Latent Sensitivity to Ambiguity: Another contributing factor to misclassification is the model's latent sensitivity to visual ambiguity. For example, images containing groups of people, unclear actions, or partial occlusions were consistently misclassified. The CNN, in its current architecture, evaluates pixels in a localized fashion, without the ability to reason holistically about the scene. This leads to:

- **Over-sensitivity to dominant patterns:** The presence of a calm background or neutral body language can outweigh smaller, violent indicators.
- **Underweighting of non-salient but semantically rich regions:** For instance, clenched fists or facial expressions may be too subtle to trigger high activation in deep layers.

Without the ability to incorporate spatial relationships or scene-level semantics, CNNs remain limited in resolving ambiguity—particularly in violence detection, where context defines intent.

Interpretation of Model Behavior: Although the CNN achieved a high test accuracy numerically, this figure alone is insufficient to justify the model's deployment. The real test of such models lies in their **class-wise fairness**, **resilience to ambiguity**, and **ethical robustness**.

In that context, the CNN fell short in three core areas:

- **Recall on minority class (Violence)** was drastically low, making it unsuitable for sensitive applications.
- **Confusion in semantically complex scenes** demonstrated a lack of contextual reasoning.
- **Overfitting to dominant class cues** indicates the need for improved regularization or data design.

These insights motivate future refinements in both architecture and training strategy, though that falls under the scope of future work discussed elsewhere.

When compared to previous studies reviewed in Section **??**, the YOLOv5s model's performance aligns well with recent trends favoring hybrid or real-time object detection architectures. Several related studies (e.g., Zhe, Elkhashab) reported accuracies in the 90–94% range for violence classification tasks using deep CNN or LSTM combinations, which matches our YOLOv5s result of 95%.

In contrast, the CNN model used in this study underperformed relative to prior work, such as Akash (2022) and Islam (2021), who employed deeper CNN variants or temporal-aware structures. Our findings support the conclusion that simple CNNs, when trained on spatial features alone, struggle to generalize violent content unless augmented with contextual modules or additional regularization.

VIII. Conclusion and Future Work

A. YOLOv5 Model

Summary and Results: This section explored the use of YOLOv5s as a lightweight, image-based classifier for detecting violent content. The model was trained on a balanced dataset of static images derived from surveillance footage and online scenes, and evaluated using standard classification metrics.

The final test accuracy reached 95%, with class-wise accuracy of 99.4% for the *Violent* class and 90.6% for the *NonViolent* class. These results demonstrate the model's strong generalization ability and robustness across both classes. Training curves showed consistent convergence without signs of overfitting, and class-specific performance was well-balanced compared to the CNN model.

Limitations and Observations: Despite its high accuracy, YOLOv5s exhibited certain limitations in edge-case scenarios. Specifically:

- **Contextual Blindness:** The model occasionally failed to detect violent content that lacked explicit visual cues such as weapons or blood. This reflects a limitation of spatial-only architectures that lack contextual or semantic reasoning.
- **Bias Toward Visual Salience:** Violent samples with strong visual features were more easily detected, while ambiguous scenes—such as arguments or tension without physical contact—were misclassified.
- **Epoch Limitations:** Although performance was strong after 10 epochs, further training or fine-tuning may enhance performance on borderline or ambiguous cases.

Additionally, a misclassified violent sample was visually analyzed (see Figure 10). The image, despite showing signs of aggressive behavior, was predicted as non-violent with high confidence. This highlights the need for improved contextual sensitivity in future iterations.

Future Work: To address the model's current limitations, the following directions are proposed:

- **Extended Training:** Increasing the number of epochs and using learning rate schedulers may help refine feature learning, especially for complex or borderline samples.
- **Loss Function Modification:** Using class-weighted loss or focal loss can improve recall on harder-to-detect violent samples by reducing the impact of class imbalance or feature ambiguity.
- **Hybrid Architectures:** Integrating YOLOv5s with sequence-aware models like ConvLSTM or using YOLO as a region proposal module can enhance temporal and contextual understanding.
- **Explainability Tools:** Applying Grad-CAM or saliency maps to understand which parts of the image contribute to YOLO's decisions can aid in refining model focus and improving trust in predictions.
- **Real-time System Integration:** Given its inference speed, YOLOv5s can be deployed in edge devices or real-time moderation pipelines, provided contextual filtering is implemented upstream or downstream.

Final Reflection: The YOLOv5s model demonstrates that fast, lightweight architectures can serve as capable tools for image-level violence classification. However, their success depends heavily on the clarity of visual cues and the presence of strongly differentiating features. In complex social scenes with subtle indicators of aggression, YOLOv5s—like most purely spatial models—may underperform.

Therefore, while YOLOv5s is a promising candidate for baseline or support-level deployment in real-time systems, its full potential may only be realized when paired with deeper contextual reasoning or hybrid models.

B. CNN Model

Summary and Results: The CNN-based classifier implemented in this study demonstrated strong learning capacity during training. The architecture—comprising three convolutional layers followed by a dense fully connected layer—was effective in extracting low- to mid-level spatial patterns. Standard operations like convolution, ReLU activation, and max-pooling allowed the network to build a multiscale understanding of visual features.

Training proceeded smoothly, with final training accuracy exceeding 99% and loss dropping to as low as 0.016 by epoch 10. However, post-training evaluation exposed substantial issues in the model's ability to generalize, particularly regarding violent samples.

Although the model achieved 99.3% overall test accuracy, this metric proved misleading. Specifically, recall for the *Violent* class dropped to just 7%, while the *NonViolent* class reached 99% recall. This imbalance indicates a severe bias toward the majority class and suggests overconfidence in classifying non-violent scenes.

Limitations and Observations: Error analysis revealed consistent failure in ambiguous scenarios—such as images with aggressive posture, facial tension, or emotionally charged interactions. Since the CNN model processes single frames and lacks contextual understanding, it struggled to differentiate between benign and potentially violent scenes.

Moreover, the model frequently assigned high confidence to incorrect predictions, indicating poor calibration and unreliable probability estimates. This overconfidence is particularly problematic in safety-critical applications like content moderation, where false negatives can result in exposure to harmful material.

Future Work: To improve the robustness, fairness, and utility of CNN-based violence detection, several directions are proposed:

- **Targeted Data Augmentation:** Simulating lighting variation, motion blur, occlusion, and scene complexity—especially for violent samples—may increase generalization to real-world inputs.
- **Regularization Techniques:** Applying dropout, L2 regularization, and early stopping can reduce overfitting and improve generalizability.
- **Handling Class Imbalance:** Introducing class weights or using focal loss can address intra-class complexity imbalance and penalize false negatives more aggressively.
- **Architectural Upgrades:** Incorporating spatial attention modules or using deeper backbones like ResNet-50 or EfficientNet-B0 may help extract richer semantic features.
- **Spatiotemporal Modeling:** CNNs lack temporal understanding. Future models should consider ConvLSTM, TimeDistributed CNNs, or 3D-CNNs to capture motion-based cues and scene progression.
- **Interpretability and Explainability:** Using tools such as Grad-CAM, SHAP, or LIME can help identify which regions influence predictions and validate model logic.
- **Multimodal Integration:** Future systems may benefit from incorporating other modalities—such as audio or metadata—to form a holistic scene interpretation.

Final Reflection: While CNNs are capable of impressive performance on training data, this study shows that they are prone to critical misclassifications in ethically sensitive domains such as violence detection. Sole reliance on accuracy is insufficient when class-level disparities—especially in the minority class—pose serious risks.

Ultimately, this reinforces the need to move toward hybrid, context-aware, and explainable models that perform not only statistically well, but ethically sound. Designing for fairness and safety is not just a technical goal, but a social imperative.

References

References

[1] K. W. Zhe, *Vision-Based Violence Detection Through Deep Learning*, Bachelor's Thesis, Universiti Tunku Abdul Rahman, Malaysia, 2024.

[2] I. M. Abundez, R. Alejo, O. Portillo-Rodríguez, F. P. Primero, J. A. A. Velázquez, and E. E. Granda-Gutiérrez, ”Threshold active learning approach for physical violence detection on images obtained from video (frame-level) using pre-trained deep learning neural network models,” 2024. [Online].

[3] A. D. Khaperkar, D. K. Khapekar, P. W. Lanjewar, V. J. A. Naidu, and A. Wani, ”Human violence detection using machine learning techniques,” Undergraduate Project, Dept. of AI Engineering, JD Engineering College, India, 2023.

[4] Y. R. Elkhashab and W. H. El-Behaidy, ”Violence detection enhancement in video sequences based on pre-trained deep models,” Faculty of Computer Science, Helwan University, Egypt, 2023.

[5] S. A. Arun Akash, S. C. Dinesh, D. R. Venkatesh, M. P. Vijayalakshmi, and S. E. Dhanalakshmi, ”Human violence detection using deep learning techniques,” *Journal of Physics: Conference Series*, vol. 2318, no. 1, p. 012003, 2022.

[6] M. Magdy, M. W. Fakhr, and F. A. Maghraby, ”Violence 4D: Violence detection in surveillance using 4D convolutional neural networks,” 2022. [Unpublished Technical Report].

[7] R. Vrskova, R. Hudec, P. Kamencay, and P. Sykora, ”A new approach for abnormal human activities recognition based on ConvLSTM architecture,” 2022. [Conference Paper].

[8] Z. Islam, M. Rukonuzzaman, R. Ahmed, M. H. Kabir, and M. Farazi, ”Efficient two-stream network for violence detection using separable convolutional LSTM,” 2021. [Preprint].

[9] B. Baran and V. Kaya, ”Detection and classification of different weapon types using deep learning,” 2021. [Online Resource].

[10] M. M. Moaaz, ”Violence detection in surveillance videos using deep learning,” Master's Thesis, 2020. [Institution not specified].