Violence Detection: A Multi-Model Approach Towards Automated Video Surveillance and Public Safety

Conference Paper · March 2024

DOI: 10.1109/iCACCESS61735.2024.10499466

6 authors, including:

Sovon Chakraborty
University of Liberal Arts Bangladesh (ULAB)
5 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Md. Ferdous Bin Hafiz
University of Liberal Arts Bangladesh (ULAB)
10 PUBLICATIONS **17** CITATIONS
SEE PROFILE

A O M Shamsuddoha
University of Liberal Arts Bangladesh (ULAB)
4 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Shakib Mahmud Dipto
University of Liberal Arts Bangladesh (ULAB)
20 PUBLICATIONS **23** CITATIONS
SEE PROFILE

2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS), 8-9 March, Dhaka, Bangladesh

Violence Detection: A Multi-Model Approach Towards Automated Video Surveillance and Public Safety

Sovon Chakraborty
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Dhaka, Bangladesh
Email: sovon.chakraborty@ulab.edu.bd

Sabrina Zahir
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
Email: sabrinabintezahir@gmail.com

Nabiha Tasnim Orchi
Computer Vision and Intelligence Systems (CVIS)
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Email: nabiha.tasnim.orchi@g.bracu.ac.bd

Md Ferdous Bin Hafiz
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Dhaka, Bangladesh
Email: ferdous.hafiz@ulab.edu.bd

A O M Shamsuddoha
Department of Electrical and Electronic Engineering
University of Liberal Arts Bangladesh
Dhaka, Bangladesh
Email: shakib.mahmud@ulab.edu.bd

Shakib Mahmud Dipto
Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Dhaka, Bangladesh
Email: shakib.mahmud@ulab.edu.bd

***Abstract*—Detection of violence at an earlier phase is crucial to intercepting potential criminal activities such as murders, rapes, and snatching. It is a critical aspect of public safety and security, involving the identification of aggressive behaviours in numerous settings. In this research, the authors are focused on exploring the efficacy of multiple Convolutional Neural Network (CNN) architectures to detect potential violent activities, especially in developing countries such as Bangladesh. The models are trained and validated with 2834 images, whereas real-life video footages are also utilized for testing purposes. To evaluate the performance of the VGG16, VGG19, and MobileNetV2 architectures, the Intersection over Union (IOU) result is observed. In contrast, the mean Average Precision (mAP) is understood to evaluate the YOLOv8 and YOLO-NAS models. It is found that YOLOv8 exhibits better performance than other architectures in the provided dataset at the training phase. The validation loss is also found to be lower in the case of the YOLOv8 model. The outcomes of this study have significant implications for enhancing security measures, aiding law enforcement, and contributing to the development of more sophisticated surveillance systems. Execution of the models, as mentioned earlier, will lead to faster and more precise identification of violent activities, thereby promoting public safety and facilitating timely interventions.**

***Index Terms*—Violence detection, CCTV images, YOLOv8, YOLO-NAS, Deep learning, transfer Learning, CNN, multi-model.**

I. Introduction

CCTV is frequently employed in monitoring systems to observe a variety of activities. The increasing number of violent incidents has made these cameras necessary for public areas. It makes everything safer by improving general security and keeping an eye on what's going on.

A. Research Motivation

The Ministry of Women and Children Affairs has implemented a project to install CCTV cameras in 108 buses operating on diverse routes in the capital city, as reported by UNB. The inauguration of the initiative, titled 'Safe Journey of Women in Public Transport' [1]. As per industry projections, the global video surveillance market is anticipated to witness significant growth, increasing from $11.5 billion in 2008 to $37.7 billion in 2015. A 2013 poll by The New York Times and CBS revealed that 78% of respondents supported deploying surveillance cameras in public spaces. Authorities often highlight notable successes, such as crucial imagery provided by cameras in identifying the Boston Marathon bombing suspects or those responsible for the 2005 London attacks. Despite these successes, lingering concerns persist regarding the potential infringement on personal privacy and the overall cost-effectiveness of surveillance systems. [2] The New York City Subway System plans to enhance security measures by installing surveillance cameras within train cars by 2025. This decision comes in response to the ongoing challenges posed by the persisting incidents of violence on the subway. City Council approved the installation of 37 ADT commercial cameras, spending over $500,000 to make the community safer. [3] These CCTV cameras require a human touch in a manual way to detect those activities. To tackle this, integrating deep learning approaches, such as

979-8-3503-5028-9/24/$31.00

Convolutional Neural Networks (CNN), enhances the ability of CCTV to detect violent activities more effectively.

B. Contributions of this Research

The contributions of this research are summarized below:

I) Proposing the creation of a deep learning model-based system to detect violent and non-violent activities from real-life CCTV footage to make the surveillance system more powerful.

II) Exploring numerous architectures of CNN in order to identify a suitable model for this system. The models are evaluated on various performance metrics.

III) This research will enhance public security and assist in understanding recent social and behavioural patterns. It will also reflect the current situation in developing countries and what safety measures should be taken.

We have trained and tested three CNN models and two state-of-the-art YOLO models.

The organization of the paper includes related works in section II, which demonstrates the relevant researches in the concerned field with the integration of CNN architectures. Section III provides a detailed insight into the methodologies utilized in this research. Section IV includes the research results and analysis with graphical insights. Finally, Section V concludes the summary and the future direction of this research.

II. Related works

Over time, AI has made significant progress, greatly simplifying our lives. Detecting specific objects is a focus of extensive research where AI plays a major role. The utilization of CCTV for identifying these unusual activities in surveillance has emerged as a major theme in research. In this field of computer vision, a lot of researchers have performed their studies on the detection of violent activity. Some of the research methods are discussed in this section.

Huszar et al. [4] proposed a method for fast and accurate violence detection in surveillance videos using 3D Convolutional Neural Networks (CNNs). They used a lightweight 3D CNN architecture called X3D-M, which was pre-trained on a large-scale action recognition dataset and fine-tuned or transfer-learned on various violence detection datasets. Magdy et al. [5] propose a deep learning architecture for violence detection in surveillance videos using four-dimensional video-level convolutional neural networks (4D CNNs). The architecture incorporates residual blocks with three-dimensional Convolution Neural Networks (3D CNNs) to learn both short-term and long-term spatiotemporal representations from the video. The proposed architecture is evaluated on four benchmark datasets, achieving impressive test accuracies: 94.67% on RWF2000, 97.29% on Crowd Violence, 100% on Movie Fight, and 100% on the Hockey Fight dataset, surpassing previous methods on RWF2000. Rfanullah et al. [6] proposed a real-time violence detection system using surveillance videos, addressing key challenges in the existing literature. The research emphasizes the difficulty of manually defining violent objects and handling uncertainty in the detection process. The results of Python simulations indicate that the MobileNet model outperforms its counterparts, achieving an accuracy of 96.66% and a low loss of 0.1329%.

Vijeikis et al. [7] introduced an innovative approach to intelligent video surveillance systems, focusing on safety monitoring through the detection of violent events, which presents a novel architecture for violence detection in video surveillance cameras. Experimental results, based on a real-world security camera footage dataset derived from RWF-2000, demonstrate compelling outcomes with an average accuracy of 0.82±2% and average precision of 0.81±3%. Honarjoo et al. [8] contribute to the field of violence detection by addressing the need for applicable and automated methods, particularly in the context of visual data acquired from surveillance cameras. The proposed method is evaluated on four public datasets, and the experimental results highlight the efficiency of this low-complexity approach. Mahdi et al. [10] address the critical need for continuous monitoring of public spaces to detect abnormal activities, which may indicate potential threats or risks. The proposed system, applicable to both indoor and outdoor academic settings, demonstrates a high accuracy rate of 95.3%.

Ali's [11] work presents an automated surveillance system for anomaly detection, addressing the challenges associated with human monitoring of surveillance cameras. The system utilizes background subtraction (BS) with a mixture of Gaussians (MoG) to model each pixel, focusing on higher-order learning in the foreground. Validation on various benchmark datasets demonstrates the robustness of the proposed system for complex video anomaly detection, with an average area under the curve (AUC) of 94.94% in frame-level evaluation across all benchmarks. The system outperforms state-of-the-art methods with a notable improvement ratio of 7.7% in AUC. Staniszewski et al.[[12] address the challenges of automatically detecting violent actions in public places through video analysis, emphasizing the limitations of current Artificial Intelligence-based techniques due to generalization problems.

Li et al. [13] present a significant contribution to the automated analysis of violent content in surveillance videos through the proposal of a deep learning model. This model is built upon 3D convolutional neural networks, eliminating the need for hand-crafted features or exclusive use of recurrent neural network (RNN) architectures for encoding temporal information. The improved internal designs incorporate compact yet effective bottleneck units for learning motion

patterns and leverage the DenseNet architecture to enhance feature reusing and channel interaction. Vieira et al. [14] address the growing demand for efficient monitoring systems to combat the increasing number of violence cases. Recognizing the potential susceptibility of such systems to failure, the authors propose the analysis and application of low-cost Convolutional Neural Networks (CNNs) techniques to automatically identify and classify suspicious events. Mobile CNN architectures were adapted and demonstrated a classification accuracy of up to 92.05% with a minimal number of parameters. To validate the practicality of the models, a prototype was implemented on an embedded Raspberry Pi platform, capable of executing the model in real-time at a speed of 4 frames-per-second.

III. Proposed Methodology

We have used three pre-trained CNN model including VGG16, VGG19 & MobileNetv2 and two most recent state-of-the-art YOLO models, including YOLOv8 & YOLO-NAS. In the YOLO model we used data.yml to configure the labels data annotate, and images were used for training on the pre-trained model. Then in the trained model is used for validation. We used the YOLO 'S' model for faster performance in both the YOLOv8 & YOLO-NAS model.

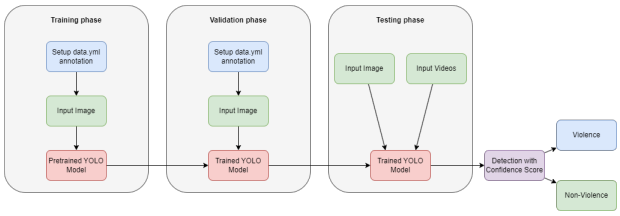

Fig. 1: YOLOv8 & YOLO-NAS Three Phase Diagram

In VGG16, VGG19, and MobileNetv2 we followed the similar approach with three phases. For training and validation, we used Pascal VOC annotation for our label data. Where images are trained using the annotated XML data on the pre-trained model, we used our trained model on the validation dataset to validate our model. Later in the testing phase, we used our trained model on both images and videos to detect violence and non-violence where IoU is observed.

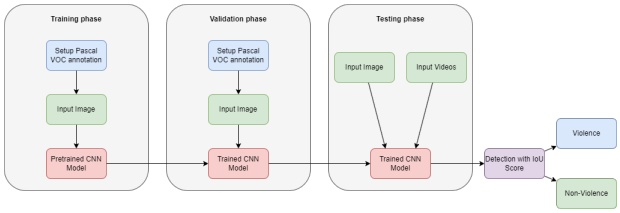

Fig. 2: VGG16, VGG19 & MobileNetv2 Model Three Phase Diagram

A. Dataset Description

We used the Roboflow image dataset [21], which is divided into three parts: train, valid, and test[fig 3]. Each of these has two label datasets, which are violence and non-violence. It includes a total number of 2834 images. Where 1969 was used for training, 575 were used for validation and 290 for testing. Moreover, RWF2000 [22] videos are used for testing only purposes on the model to see how accurately it detects throw videos.

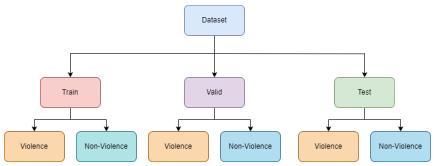

Fig. 3: YOLOv8 & YOLO-NAS Three Phase Diagram

Some of the sample images from our dataset are shown below:

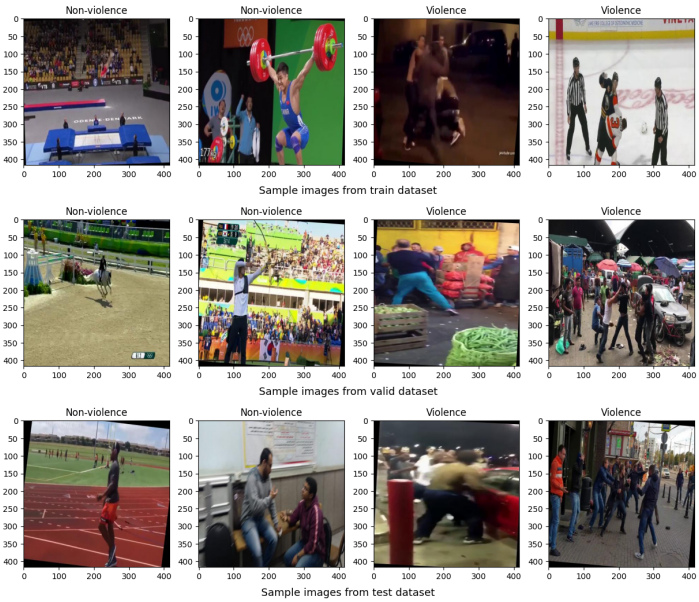

Fig. 4: Sample images from Dataset

B. Preprocessing

Before we use our image for training, we need to annotate it. For annotating, we used two different approaches. One is the normal text format of annotation for the YOLO model, and the other is Pascal VOC XML annotation for three CNN models. That annotated image is then used in our model for training. For the YOLO model, we use data.yml to configure our image directory and annotation.txt file. In the CNN model, we used an XML file for Pascal VOC XML where a bounding box annotated value is present.

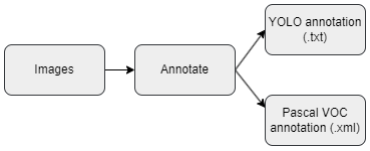

Fig. 5: Dataset annotation

C. Model Implementation

In our proposed deep learning-based CNN model, we used three pre-trained models, including VGG16, VGG19, MobileNetV2 and the state-of-the-art YOLOv8, YOLO-NAS models.

- **VGG16 & VGG19** VGG, short for Visual Geometry Group, is a convolutional neural network (CNN) architecture that includes models like VGG16 and VGG19. VGG16 uses multiple 33 kernel-sized filters sequentially, while VGG19 has a depth of 19 layers and was trained on over a million pictures from the ImageNet database. The primary idea behind the VGG architecture is to keep the convolution size constant and modest while creating an incredibly deep network that can classify images of different classes. The input for VGG is set to a 224 × 244 RGB picture [15].

- **MobileNetV2** It is highly effective for image classification. This lightweight deep learning model is built on the convolutional neural network architecture and utilizes TensorFlow to provide weight values for input images. The base layer of MobileNetV2 is first removed, and a new trainable layer is added to the top of the model. This modified model then operates on the dataset provided, extracting the most relevant features from the images. MobileNetV2 consists of 19 layers, including bottleneck structures that help to minimize computational costs while maintaining high accuracy [16].

- **YOLOv8** The latest model in the YOLO is the YOLOv8 model. It is created by Ultralytics. [17] They also released YOLOv5.There are a total of five versions of YOLOv8 models: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), YOLOv8xl (extra-large). It is capable of different tasks, including object detection, segmentation, pose estimation, tracking and classification.[18] YOLOv8 shares a similar backbone with YOLOv5, with a notable modification in the CSPLayer, now called the C2f module. The C2f module (cross-stage partial bottleneck with two convolutions) enhances detection accuracy by combining high-level features with contextual information. Unlike its predecessors, YOLOv8 adopts an anchor-free model with a decoupled head, enabling independent processing of objectness, classification, and regression tasks. This design enhances overall accuracy by allowing each branch to focus on its specific task. In the output layer, YOLOv8 employs the sigmoid function for the objectness score, indicating the probability of an object being present within the bounding box. For class probabilities, the softmax function is used, representing the likelihood of the object belonging to each possible class.YOLOv8 introduces CIoU and DFL loss functions for bounding-box loss and binary cross-entropy for classification loss. These loss functions significantly improve object detection performance, especially when dealing with smaller objects [18].

- **YOLO-NAS** One the most recent state-of-the-art model, YOLO-NAS, was released by Deci in May 2023 [19]. YOLO-NAS is a specialized model designed for detecting small objects, improving localization accuracy, and optimizing performance for real-time applications on edge devices. Notably, it is open-source, making it accessible for research purposes. Key innovations in YOLO-NAS include:
  - Quantization-aware modules (QSP and QCI): These modules employ re-parameterization for 8-bit quantization, minimizing accuracy loss during post-training quantization.
  - Automatic architecture design (AutoNAC): Leveraging Deci's proprietary NAS technology, YOLO-NAS achieves automatic architecture design.
  - Hybrid quantization method: YOLO-NAS selectively quantizes specific model parts to balance latency and accuracy, deviating from standard quantization and affecting all layers uniformly.
  - Pre-training regimen: This involves automatically labelled data, self-distillation, and large datasets.

The AutoNAC system, integral to YOLO-NAS creation, is versatile, accommodating various tasks, data specifics, inference environments, and performance goals. It aids users in identifying an optimal structure, offering a precise balance between precision and inference speed. Considering factors such as data, hardware, compilers, and quantization, AutoNAC plays a crucial role in the inference process. Moreover, RepVGG blocks are incorporated into the model architecture during the NAS process for compatibility with post-training quantization (PTQ). Three architectures—YOLO-NASS, YOLO-NASM, and YOLO-NASL (representing small, medium, and large configurations, respectively)—are generated by varying the depth and positions of the QSP and QCI blocks.[18] We have used the CNN (VGG16,19 & MobileNetv2) in a similar way to compare it with each model, similar to YOLO models. A total number of 25 epochs are used to train each model. In CNN models, we used a total of 25 epochs, the same as the YOLO model with a batch size of 32. Input shape is 224 and Smooth l1 loss is used. We predicted the bounding box's four values in proportion to IoU. On the other hand, while training the YOLO model, we used a small, faster pre-trained model with a total of 25

epochs and 16 batch size where mAP50 is observed.

D. Performance Evaluation

For performance evaluation, after training each model, we test it through a valid, test dataset. When we assess how well object detection methods perform, we look at two aspects: how accurately they find the object (localization) and how well they determine its category (classification). To measure this, we commonly use evaluation metrics like Mean Average Precision (mAP) [21] and Intersection over Union (IoU). For our model evaluation, we focused on object detection with a keen eye on localization, using IoU and mAP. IoU, also known as the Jaccard Index [20], compares the overlapping area of bounding boxes by dividing the intersection area by the union area (as shown in Eq.1 The IoU metric gives a normalized value between 0 and 1, where 0 means no overlap and 1 means complete overlap. On Eq.2 Mean Average Precision (mAP) is shown.

$$IoU = \frac{(A \cap B)}{(A \cup B)} \tag{1}$$

$$\text{mAP} = \frac{1}{N_{\text{classes}}} \sum_{c=1}^{N_{\text{classes}}} \text{AP}_c \tag{2}$$

Where:

- N_{classes} is the number of classes.
- AP_c is the average precision for class c.

IV. Result Analysis

We have used a total of three CNN models and two state-of-the-art latest YOLO models. In the CNN model, we observed the IoU. The result we found for CNN models is shown in Table 1 below. In YOLO models, we observed mAP, which is shown in Table 2 below.

TABLE I: CNN Model IoU Results

Model	Val Loss	Val Mean IoU	Loss	Mean IoU
VGG16	0.0271	0.8446	0.0040	0.9280
VGG19	0.0263	0.8452	0.0030	0.9365
MobileNetV2	0.0462	0.7811	0.0028	0.9412

From Table I, it is found that, VGG19 has the lowest validation loss despite of the faster execution of MobileNetv2. But when it comes to the training data loss then, MobileNetv2 outperforms the other two models by a slight margin.

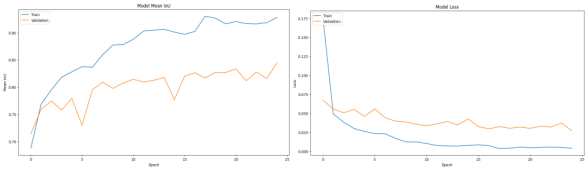

Fig. 6: VGG16 model accuracy & loss

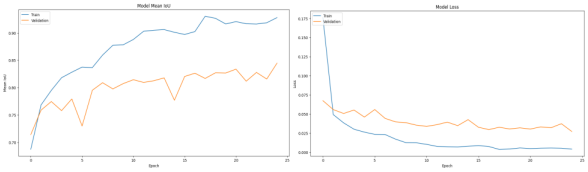

Fig. 7: VGG19 model accuracy & loss

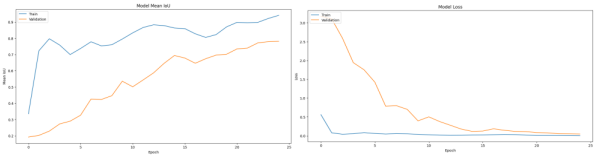

Fig. 8: MobileNetv2 model accuracy & loss

To represent the detailed overview of the explored models, the accuracy and data loss for each epoch is depicted in Fig 6, Fig 7 and Fig 8. These figures show that the data loss for each model has decreased gradually with more training time. On the contrary, a periodic increment in accuracy is shown for each model with the advance of the epoch number.

The overview of the YOLO models are shown below in Table II where it can be observed that the mAP value for YOLOv8 is lesser than YOLO-NAS.

TABLE II: Object Detection Model mAP50

Model	mAP50
YOLOv8	0.884
YOLO-NAS	0.807

As we can see in the YOLO model, YOLOv8 overtake YOLO-NAS where mAP50 of YOLOv8 is 0.884 and YOLO-NAS is 0.807 mAP50 on all classes.

Figure 9 reflects the confusion matrix for YOLOV8, which performs satisfactory numbers in terms of different classes. :

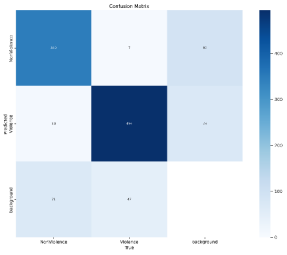

Fig. 9: YOLOv8 Confusion Matrix

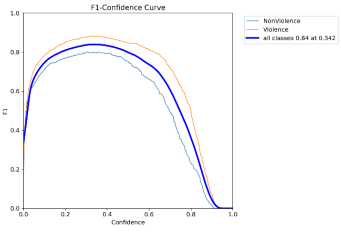

Fig. 10: YOLOv8 F1-Confidence Curve

Fig 11 shows the detection of violence from images while testing the models.

Fig. 11: Sample of images detected by YOLOv8

Some of the detections of violence using YOLOv8 model from Real-time CCTV footages are shown in Fig 12.

Fig. 12: Sample of video dataset image detected by YOLOv8

V. Conclusion

The study aims to contribute to monitoring systems in surveillance and public safety with a variety of experiments by detecting violent and non-violent activity. We used three CNN models and two state-of-the-art YOLO models. We found that YOLOv8 did great on our dataset and performed better than YOLO-NAS and the other three CNN models. From our analysis, we have found that the results we get from CNN models and YOLO models have a big difference. Even though CNN models train well when detecting multiple objects from an image or video, their accuracy is not up to par. The IoU score is lower in that manner. On the other hand, YOLO model performed great on multiple detections with a high confidence score. Between YOLO-NAS and YOLOv8, YOLOv8 performed much better on our dataset. In the future, we will compare the different datasets using the latest state-of-the-art models and integrate more security.

References

[1] Rahman, Matiur (Editor Publisher). "Prothom Alo English Desk," *CCTV cameras installed in 108 buses in Dhaka*, October 16, 2022. https://en.prothomalo.com/bangladesh/city/2ur6c9cg1v

[2] Kille, L.W., Maximino, M. (2014, February 11). *The effect of CCTV on public safety: Research roundup.* Retrieved from https://journalistsresource.org/politics-and-government/surveillance-cameras-and-crime

[3] González-Britt, O. (2023, December 6). *'More eyes in more locations.' AGPD boosts city surveillance with 37 security cameras* [Updated December 7, 2023].

[4] Huszar, V., Adhikarla, V., Negyesi, I., Krasznay, C. (2023). *Toward fast and accurate violence detection for automated video surveillance applications* (pp. 1-1). *IEEE Access*. Retrieved from https://doi.org/10.1109/ACCESS.2023.3245521

[5] Magdy, M., Fakhr, M., Maghraby, F. (2022). *Violence 4D: Violence detection in surveillance using 4D convolutional neural networks*. *IET Computer Vision*, 17. Retrieved from https://doi.org/10.1049/cvi2.12162

[6] Irfanullah, Tariq Hussain, Arshad Iqbal, Bailin Yang, and Altaf Hussain. *Real time violence detection in surveillance videos using Convolutional Neural Networks*.Multimedia Tools and Applications 81, no. 26 (2022): 38151–38173. https://doi.org/10.1007/s11042-022-12599-1

[7] Vijeikis, Romas, Vidas Raudonis, and Gintaras Dervinis. *Efficient Violence Detection in Surveillance*. Sensors 22, no. 6 (2022): 2216. https://doi.org/10.3390/s22062216

[8] Honarjoo, Narges, Ali Abdari, and Azadeh Mansouri. *Violence Detection Using Pre-trained Models*. 2021 IEEE International Conference on Pattern Recognition and Image Analysis (IPRIA), April 2021, pp. 1–4. https://doi.org/10.1109/IPRIA53572.2021.9483558.

[9] Davies, Rick. *The 'Most Significant Change' (MSC) Technique: A Guide to Its Use*. 2005. https://doi.org/10.13140/RG.2.1.4305.3606.

[10] Mahdi, Muthana, Amer Jelwy, Abdulghafor Abdulghafour, B Dewan, and Al-Waqf Al-Sunni. *Detection of Unusual Activity in Surveillance Video Scenes Based on Deep Learning Strategies*. IET Computer Vision 9 pages, vol. 9 (2022): 9. https://doi.org/10.29304/jqcm.2021.13.4.858.

[11] Ali, Manal Mostafa. *Real-time Video Anomaly Detection for Smart Surveillance*. *IET Image Processing* 17, no. 5 (2023): 1375–1388. https://onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12463

[12] Ciampi, Luca, Paweł Foszner, Nicola Messina, Michał Staniszewski, Claudio Gennaro, Fabrizio Falchi, Gianluca Serao, Michał Cogiel, Dominik Golba, Agnieszka Szczesna, and Giuseppe Amato. *Bus Violence: An Open Benchmark for Video Violence Detection on Public Transport*. Sensors 22, no. 21 (2022): 8345. https://doi.org/10.3390/s22218345.

[13] Li, Ji, Xinghao Jiang, Tanfeng Sun, and Ke Xu. *Efficient Violence Detection Using 3D Convolutional Neural Networks*. 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8.

[14] Vieira, Joelton Cezar, Andreza Sartori, Stéfano Frizzo Stefenon, Fábio Luis Perez, Gabriel Schneider de Jesus, and Valderi Reis Quietinho Leithardt. *Low-Cost CNN for Automatic Violence Recognition on Embedded System*. IEEE Access 10 (2022): 25190–25202. https://doi.org/10.1109/ACCESS.2022.3155123.

[15] Akter, S., Shamrat, F. M. J. M., Chakraborty, S., Karim, A., Azam, S. (2021). *COVID-19 Detection Using Deep Learning Algorithm on Chest X-ray Images*. *Biology*, *10*(11), 1174. https://doi.org/10.3390/biology10111174

[16] Chakraborty, Sovon, F.M. Javed Mehedi Shamrat, Md. Masum Billah, Md. Al Jubair, Md. Alauddin, and Rumesh Ranjan. *Implementation of Deep Learning Methods to Identify Rotten Fruits*. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1207–1212. https://doi.org/10.1109/ICOEI51242.2021.9453004.

[17] Jocher, Glenn, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLO*, version 8.0.0, January 10, 2023. GitHub. https://github.com/ultralytics/ultralytics.

[18] Terven, Juan and Diana Cordova-Esparza. *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. arXiv preprint arXiv:2304.00501 (2023). https://doi.org/10.48550/arXiv.2304.00501

[19] Nadeem, Muhammad. *Deep Labeller: Automatic Bounding Box Generation for Synthetic Violence Detection Datasets*. 2021. https://doi.org/10.36227/techrxiv.15169041.

[20] Shah, Deval. *Mean Average Precision (mAP) Explained: Everything You Need to Know*. V7 Labs Blog, March 7, 2022. https://www.v7labs.com/blog/mean-average-precision.

[21] Shah. textitViolence Dataset. 2022. https://universe.roboflow.com/shah-xxxqs/violence-3h8pw. Accessed: 2024-01-27.

[22] Cheng, Ming, Kunjing Cai, and Ming Li. *RWF-2000: An Open Large Scale Video Database for Violence Detection*. 2021. https://doi.org/10.1109/ICPR48806.2021.9412502.