Violence Detection Using YOLOv8 Towards Automated Video Surveillance and Public Safety *

*Note: Sub-titles are not captured in Xplore and should not be used

1st Md. Alahi Almin Tansen
Dept. of Computer Science and Engineering
European University of Bangladesh
Dhaka, Bangladesh
tansen550@gmail.com

2nd Umayer Mohammad Affan
Dept. of Computer Science and Engineering
European University of Bangladesh
Dhaka, Bangladesh
umaffan18@gmail.com

3rd Afruja Sultana muniya
Dept. of Computer Science and Engineering
European University of Bangladesh
Dhaka, Bangladesh
afrujasultanamuniya@gmail.com

4th Md. Delower Hosen
Dept. of Computer Science and Engineering
European University of Bangladesh
Dhaka, Bangladesh
delowerhosen713@gmail.com

5thSovon Chakraborty
Dept. of Computer Science and Engineering
University of Liberal Arts
Dhaka, Bangladesh
sovon.chakraborty@ulab.edu.bd

***Abstract*—Violence detection is a critical aspect of public safety and security, involving the identification of aggressive behaviors in various settings. In this study, we explore the efficacy of convolutional neural network (CNN) models, specifically VGG16, VGG19, and MobileNetV2, alongside two YOLO (You Only Look Once) models, YOLOv8 and YOLO-NAS, for detecting violence activities in video footage. Utilizing dataset from Roboflow violence dataset consisting of 2834 images. Our findings reveal notable variations in the models' performance. Where VGG16, VGG19 and MobileNetv2 performance is measured by IoU result. While measuring YOLOv8 and YOLO-NAS we did mAP result comparison. We found that YOLOv8 did great on our dataset compared to YOLO-NAS and the other three CNN models. This study's outcomes have significant implications for enhancing security measures, aiding law enforcement, and contributing to the development of more sophisticated surveillance systems. The adoption of these models could lead to quicker and more precise identification of violent activities, thereby promoting public safety and facilitating timely interventions.**

***Index Terms*—Violence detection, CCTV images, VGG16, VGG19, MobileNetV2, YOLOv8, YOLO-NAS, Deep learning, Transfer learning, CNN, Multi Model.**

I. Introduction

In a monitoring system, CCTV is widely used to observe various activities. These cameras have become essential for public places due to the rising incidents of violence. It helps keep an eye on what's happening, enhancing overall security, and making it safer for everyone. 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 conducted by The New York Times and CBS revealed that 78 percent of respondents expressed support for the deployment of 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] 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.[3] City Council approved the installation of 37 ADT Commercial cameras, spending over 500,000 to make the community safer.[4] These CCTV cameras required human touch manual way to detect those activity. To tackle this the integration of deep learning approaches, such as Convolutional Neural Networks (CNN), enhances the ability of CCTV to detect violent activities more effectively.

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. The goal is to enhance the monitoring process with reduced deployment costs. To support this, the researchers curated a dataset containing instances of both violence and non-violence actions in crowded and non-crowded environments. Mobile CNN architectures were adapted and demonstrated a classification accuracy of up to 92.05 percent 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. Additionally, a warning system was developed to recognize pre-fight behavior and anticipate violent acts, enabling the timely alerting of security to potential threatening situations.

II. Related works

Over time, AI has made significant progress, greatly simplifying our lives. Detecting specific objects, a focus of extensive research where AI plays a major role. Detecting particular objects, such as fights or violent activities, has become a key area of interest for researchers. 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 done their detection of violent activity. Some of the research methods are discussed in this section.

Huszar et al.[5] 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. They showed that their method outperformed several state-of-the-art methods on different metrics, such as accuracy and area under curve. They also demonstrated the robustness of their method under video compression artifacts, which are common in remote server processing applications.

Magdy et al.[6] 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. Additionally, they utilize ResNet50 as the backbone for the 3D convolutional networks and dense optical flow for region-of-interest localization. The proposed architecture is evaluated on four benchmark datasets, achieving impressive test accuracies: 94.67 percent on RWF2000, 97.29 percent on Crowd violence, 100 percent on Movie fight, and 100 percent on the Hockey Fight dataset, surpassing previous methods on RWF2000.

Rfanullah et al.[7] 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. An additional challenge involves the scarcity of labeled datasets due to the labor-intensive nature of manual video annotation. The study introduces a novel approach by evaluating Convolutional Neural Network (CNN) models, specifically comparing the proposed MobileNet model with established architectures such as AlexNet, VGG-16, and GoogleNet. Through Python simulations, the results indicate that the MobileNet model outperforms its counterparts, achieving an accuracy of 96.66model exhibits superior performance in terms of accuracy, loss, and computation time, particularly demonstrated on the hockey fight dataset. This research fills a gap in the current literature by providing an effective violence detection solution with low computation requirements and high accuracy in surveillance environments. Vijeikis et al.[8] introduced an innovative approach to intelligent video surveillance systems, focusing on safety monitoring through the detection of violent events. The paper presents a novel architecture for violence detection in video surveillance cameras. The proposed model employs a spatial feature-extracting U-Net-like network, utilizing MobileNet V2 as an encoder, and incorporates Long Short-Term Memory (LSTM) for temporal feature extraction and classification. Notably, the model is computationally light while maintaining effective performance. 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 percent and average precision of 0.81±3 percent.

Honarjoo et al.[9] 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. In their study, the authors employ pre-trained deep neural networks to develop a low-complexity approach for violence detection. The features extracted from pre-trained models, specifically ResNet-50 and VGG16, are pooled and input into a fully connected network to ascertain the occurrence of a violent action. The proposed method is evaluated on four public datasets, and the experimental results highlight the efficiency of this low-complexity approach. Notably, the study compares favorably with other methods that employ more time-consuming networks, such as recurrent ones.

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 paper emphasizes the importance of intelligent video surveillance in this context, particularly with the integration of artificial intelligence, machine learning, and deep learning technologies. The authors propose a deep learning technique for

distinguishing normal and abnormal behaviors in real-time video footage, with the system triggering an alarm message to authorities when suspicious activity is detected. The method involves extracting successive frames from a video, calculating features in the first phase, and employing a classifier in the second phase to predict whether the behavior is suspicious or normal. The proposed system, applicable to both indoor and outdoor academic settings, demonstrates a high accuracy rate of 95.3 percent.

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. The foreground objects are then processed through convolutional autoencoders to distinguish abnormal events from normal ones, identifying signs of threat and violence in real-time. Additionally, object detection is applied to the entire scene, highlighting regions of interest with bounding boxes to minimize human intervention. The system generates alarms during recognition time, notifying the presence of anomalies and potentially suspicious actions. 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 percent in frame-level evaluation across all benchmarks. The system outperforms state-of-the-art methods with a notable improvement ratio of 7.7 percent 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. These algorithms heavily rely on large amounts of annotated data and often experience a significant drop in performance when faced with scenarios not encountered during the supervised learning phase. To overcome this, the authors introduce the Bus Violence benchmark, a pioneering large-scale collection of video clips specifically designed for violence detection in the context of public transport. The benchmark includes simulated violent actions inside a moving bus, considering changing conditions such as varying backgrounds and lighting. The paper also conducts a performance analysis of several state-of-the-art video violence detectors pre-trained on general violence detection databases, revealing moderate performances. This highlights the difficulties in generalizing these popular methods to the new scenario, underscoring the necessity for specialized labeled data, as provided by the newly established Bus Violence benchmark.

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. These design choices prove more effective in capturing spatiotemporal features while requiring relatively fewer parameters. The performance of the proposed model is rigorously validated on three standard datasets, demonstrating superior recognition accuracy compared to other advanced approaches. Supplementary experiments further attest to the effectiveness and efficiency of the model, with final results showcasing its advantages over state-of-the-art methods in both recognition accuracy and computational efficiency.

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. The goal is to enhance the monitoring process with reduced deployment costs. To support this, the researchers curated a dataset containing instances of both violence and non-violence actions in crowded and non-crowded environments. Mobile CNN architectures were adapted and demonstrated a classification accuracy of up to 92.05 percent 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. Additionally, a warning system was developed to recognize pre-fight behavior and anticipate violent acts, enabling the timely alerting of security to potential threatening situations.

III. Proposed Methodology

We have used three pretrained CNN model including VGG16, VGG19 and MobileNetv2 and two most recent state of the art YOLO models including YOLOv8 and YOLO-NAS. In our proposed method there are three phase; Training Phase, Validation Phase and Testing phase. In YOLO model we used data.yml to configure the labels data annotate and images are used for training on pretrained model then in validation trained model is used for validation. In the testing phase both images and videos are used on our trained model for testing and detecting violence and non-violence where confidence scored is observed. We used YOLO 'S' model for faster performance in both YOLOv8 and YOLO-NAS model. 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 pretrained model

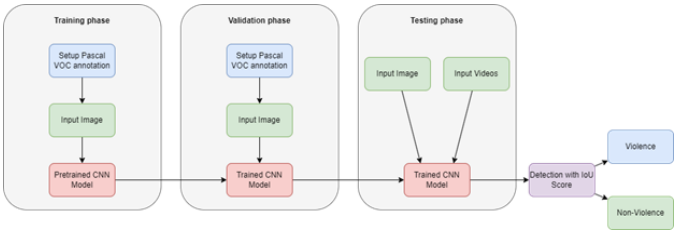

Fig. 1. *YOLOv8 & YOLO-NAS Three Phase Diagram*

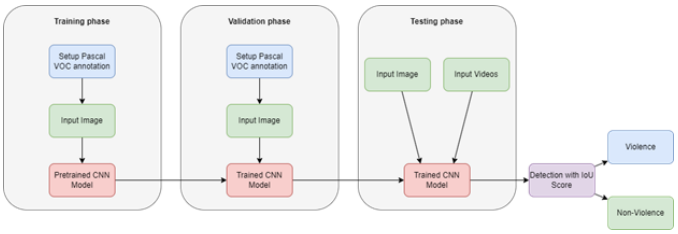

Fig. 2. VGG16, VGG19 and MobileNetv2 Model Three Phase Diagram

then validation we used our trained model on 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.

A. Dataset Description

We used Roboflow image dataset[22] which is divided into three part train,valid and test[fig 3]. Each of these has two label dataset which is violence and non-violence.It includes total number of 2834 images.h Were 1969 used for training, 575 used for validation and 290 for testing. Moreover RWF200[23] videos are used for testing only purpose on the model to see how accurately it detect throw videos.

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

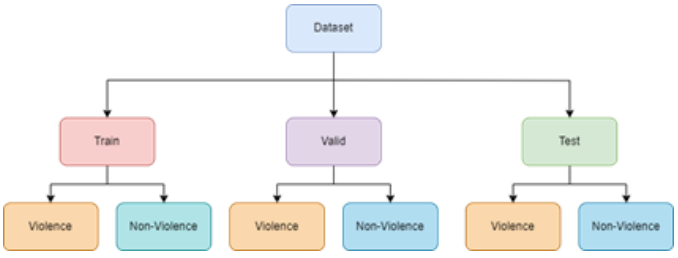

Fig. 3. Dataset diagram

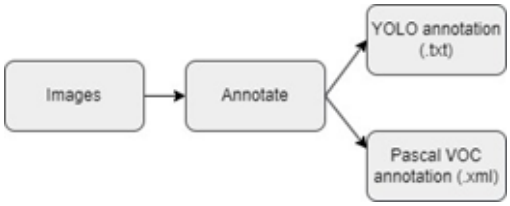

Fig. 4. Sample images from Dataset

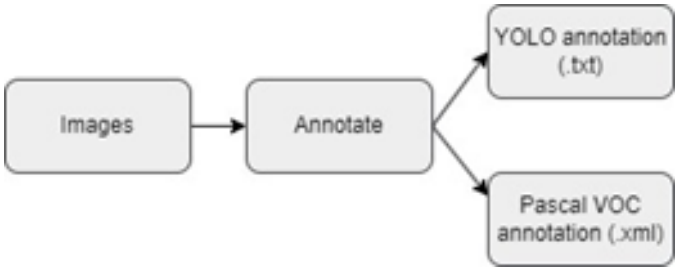

Fig. 5. Dataset annotation

B. Preprocessing

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

C. Model Implementation

In our proposed deep learning based CNN model we used three pre-trained model including VGG16, VGG19 , MobileNetV2 and State of the art YOLOv8, YOLO-NAS model. **VGG16 and 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 kernelsized 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 YOLOv8 model. It is created by Ultralytics.[17] They also released YOLOv5.There are total of five version of YOLOv8 models: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), YOLOv8xl (extra-large). It is capable of different task 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 are model YOLO-NAS 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 affecting all layers uniformly.
- - Pre-training regimen: This involves automatically labeled 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, YOLONASM, 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 and MobileNetv2) in similar way to compare it with each models, Similar with YOLO models. A total number of 25 epoch is used to train

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$
$$AP_k = \text{the AP of class } k$$
$$n = \text{the number of classes}$$

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

each model. In CNN models we used total of 25 epoch same as YOLO model. Batch size of 32 is used. Input shape is 224. Smooth l1 loss is used. Where we predicted the bounding box four value in proportion to IoU. On the other hand while training YOLO model we used small faster pretrained model where total of 25 epoch and 16 batch size is used. Where mAP50 is ovserved.

IV. Result Analysis

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

More overview on CNN models: Below shows the overview of YOLO models.

Table 2: YOLO model mAP result

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

More detail result of best performed YOLOv8 model: fig:Some of the result of YOLOv8 model detection sample:

V. Discussion

From our analysis we have found that the result we get from CNN models and YOLO models is has a big difference. Even though CNN models train well but when it comes to detect multiple object from a image or video its not that much accurate. IoU score is less on that manner. On the other hand YOLO model did great on multiple detection with confidence score. Between YOLO-NAS and YOLOv8 , YOLOv8 did much better on our dataset

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

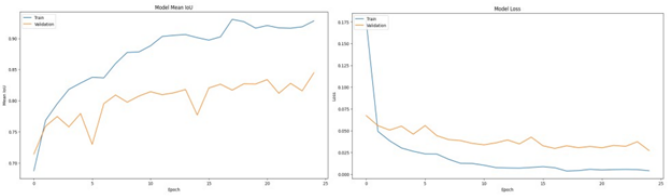

Fig. 6. VGG16 model accuracy & loss

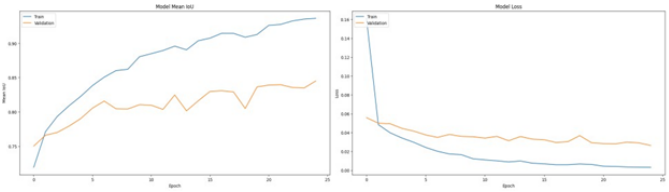

Fig. 7. *: VGG19 model accuracy & loss*

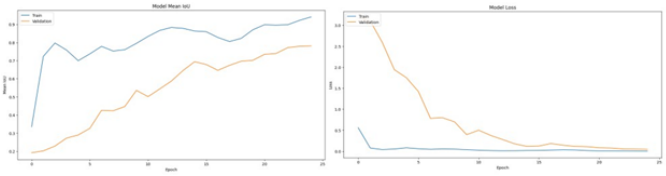

Fig. 8. *MobileNetv2 model accuracy & loss*

Model	mAP50
YOLOv8	0.884
YOLO-NAS	0.807

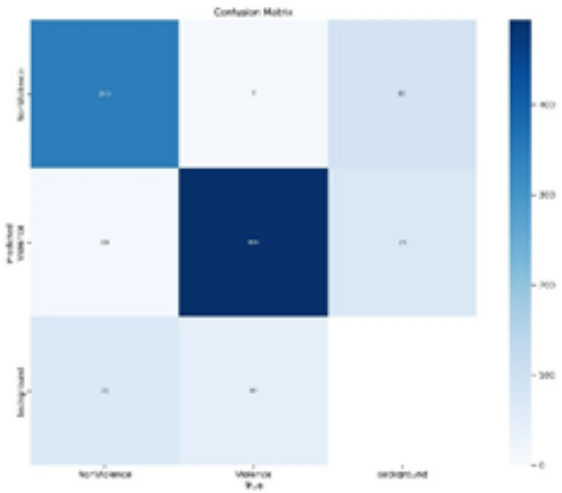

Fig. 9. *YOLOv8 Confusion Matrix*

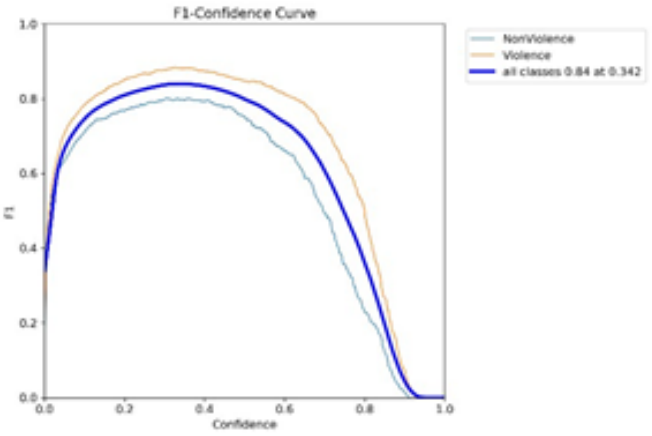

Fig. 10. *YOLOv8 F1-Confidence Curve*

VI. Conclusion

The purpose of the study is to contribute monitoring system in surveillance and public safety with verity of experiment by detecting violence and non-violence activity. Where we used three CNN models and two state of the art YOLO models. We found that YOLOv8 did great on our dataset and did better than YOLO- NAS and other three CNN models. We future work we will do more comparison of different dataset on latest state of the art models.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.