Weapon Detection in Surveillance Videos Using YOLOV8 and PELSF-DCNN

Dr Raman Dugyala[1*], *M Vishnu Vardhan* Reddy[2], *Ch Tharun* Reddy[3] and *G* Vijendar[4]

[1]Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad-500075
[2]Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad-500075
[3]Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad-500075
[4]Department of Information and Technology, Gokaraju Rangaraju Institute of Technology, Hyderabad-500090

Abstract. Weapon detection (WD) provides early detection of potentially violent situations. Despite deep learning (DL) algorithms and sophisticated closed-circuit television (CCTVs), detecting weapons is still a difficult task. So, this paper proposes a WD model using PELSF-DCNN. Initially, the input video is converted into frames and pre-processed. The objects in the pre-processed frames are detected using the YOLOv8. In meantime, motion estimation is done using the DS algorithm in the pre-processed images to cover all the information. Then, the detected weapons undergo a sliding window process by considering the motion estimated frames. The silhouette score is calculated for detected humans and other objects. Now, the features are extracted and the important features are selected using the CSBO algorithm. The selected features and the output of YOLOv8 are given to the PELSF-DCNN classifier. Finally, the confidence score is calculated for the frame to define the number of weapons. In an experimental evaluation, the proposed method is found to be more efficient than the existing methods.

1. Introduction

Due to the rise in crime during crowded events, security is always a top concern in all fields. The installation of video surveillance systems can recognise and analyse the scene due to the increasing demand for safety protection [9]. A low-cost way to monitor vast regions without obstructing the movement of people is through video surveillance [10]. They are developed as tools in public administration-related activities to maintain public order, to regulate unpleasant and anti-social behaviour [11]. Dangerous weapons are being used in criminal activities and terrorism. Therefore, the implementation of WD in a surveillance camera can detect weapons from the video feed [12].

* Corresponding author: raman.vsd@gmail.com

According to Jain et al. [9], WD is the detection of unexpected, unpredictable, and odd items that are not thought of as regularly occurring events. Although there are numerous WD models available, Kaya et al. [13] found that it was challenging to identify weapons because to their varied sizes, shapes, and backdrop colors. Self-occlusion and similarity between objects and backdrop structures are the biggest problems in WD [8]. When a portion of the gun is obstructed on one side, self-occlusion happens. When many objects resemble weapons, there is a similarity between them [14].

The aforementioned circumstances will worsen, which could increase occurrences of threat detection that is delayed. Various parties created an artificial intelligence (AI)-based solutions with a focus on the detection of hazardous objects in response to these difficulties [15]. However, because these delicate systems are connected to alarms, the rate of false-positive and false-negative results was a problem in AI systems [7]. Convolutional neural networks (CNNs) and DL approaches have produced impressive object detection results in recent years.

1.1 Problem Statement

Existing WD techniques have some drawbacks, which are described below,

- Customized weapons that vary from the generic look of pistols lead to severe accidents and CCTV recordings had problems in quality.
- The dangerous object is hidden by the perpetrator most of the time. Existing approaches faced false positive and negative performance.

To overcome these issues, a novel PELSF-DCNN method for WD is proposed. The main objectives are as follows,

- To find varying weapons and increase the quality, the sliding window process and PDFMSR is introduced.
- To detect hidden weapons and to classify the weapons, silhouette function and PELSF-DCNN is proposed.

The organization of this paper's structure is as follows: The earlier research pertaining to the suggested strategy is analyzed in Section 2. Section 3 goes over the suggested approach. The effectiveness of the suggested strategy is examined in Section 4. Section 5 concludes the paper with a thesis statement.

2. Literature Survey

(Narejo et al., 2021) built a computer-based, fully automated method to recognize common weapons. YOLOv3 object detection model was implemented to detect weapons. The training results confirmed that YOLOv3 outperformed YOLOv2 and CNN. However, YOLOv3 suffered from large localization errors.

(Ahmed et al., 2022) presented a real-time WD system usingScaled-YOLOv4. The model was enhanced using the TensorRT network on a well-known edge computing device. The analysis showed that the model resulted in improved Mean Average Precision (mAP) scores as well as lowcost. However, there were a lot of false positives.

(Fathy & Saleh, 2022) integrated DLand Fog Computing with Software-Defined Networking for WD in Video Surveillance Systems. To extract the video frames with firearms, the DL model was used inside the artificial intelligence model. Although it was capable of detecting concealed weapons, it was constrained by the presence of pricey technology.

(Ashraf et al., 2022) aimed to reduce the number of false positives and false negatives in WD. The framework was based on YOLO and Area of Interest. The results made it clear that YOLO-v5 achieved a high recall rate and speed of detection. But, the high recall rate failed to spot weapons in some scenarios.

(Bhatti et al., 2021) focused to detect harmful weapons using CCTV footage. The relevant confusion objects inclusion notion was proposed, and the binary classification was done using the pistol class as the reference class. Guns made of other materials cannot be identified because the device was made for metal detection.

(Galab et al., 2021) suggested a technique for brightness enhancement of knife detection in surveillance systems. The research proposed a threshold to evaluate the brightness of the frame. The effectiveness of the technique in identifying knives was confirmed by the results. However, the research did not offer a comprehensive strategy for finding additional harmful objects like guns, pistols, etc.

(Kalla & Suma, 2022) introduced a new automated model for effective WD in CCTV. A support Vector Machine was used for classification. The outcomes showed that the model had a high level of accuracy. The constructed model, however, needed a lot of computing power to train on the data.

3. Proposed Weapon Detection Framework

In this paper, the PELSF-DCNN model is proposed for WD. The proposed work includes the following phases. Figure 1 displays the block diagram of the suggested methodology.

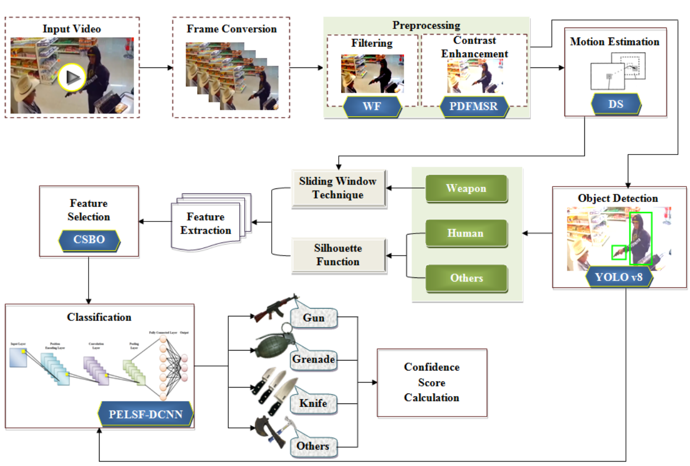


Fig. 1 Proposed Model's Block Diagram

3.1 Pre-processing

Initially, the input video is converted into frames. Then, these converted frames $\Re$ undergo preprocessing in two steps namely; filtering and contrast enhancement.

Filtering: This filtering process by WF helps in reducing the sensor-based noises present in the frames. The filtering process is expressed as,

$$WF_{\Re(u,v)} = \frac{H^*(u,v)G_{xx}(u,v)}{|H(u,v)|^2 G_{xx}(u,v) + G_{nn}(u,v)} \tag{1}$$

Where, u, v denotes the pixels, the image's power spectra are specified as $G_{xx}(u,v)$, the additive noise is signified as $G_{nn}(u,v)$, the blurring filter is symbolized as $H(u,v)$, and $WF_{\Re(u,v)}$ denotes the noise removed frames.

Contrast enhancement: After filtering, the contrast of the frames is enhanced using the PDFMSR algorithm. The MSR algorithm estimates the right illumination image. But, it did not focus on different pixels because it considers the single scale value. So, this research methodology assigns the weight value byprobability density function of pixel intensities. The enhancement $(C(u,v))$ is defined as,

$$C(u,v) = \frac{1}{n}\sum_{i=1}^{n}\left\{\log\left[WF_{\Re(u,v)}\right] - I \times \mu \times \log\left[g(u,v) * WF_{\Re(u,v)}\right]\right\} \tag{2}$$

$$\mu(u,v) = \mu_{\max}\left\{\frac{\mu(u,v) - \mu_{\min}}{\mu_{\max} - \mu_{\min}}\right\} \tag{3}$$

Where, I denotes the illumination factor, $g(u,v)$ is the Gaussian function, and μ specifies the weight value calculated using PDF. Finally, the preprocessed frames are obtained and are denoted as $\Re_{pre}$.

3.2 Object Detection

In this phase, object detection is done in the preprocessed frames $\Re_{pre}$ using the YOLOv8. A detailed explanation of the YOLOv8 is given below.

- Initially, the frames $\Re_{pre}$ are divided into a number of grids. For each grid, the anchor boxes are defined. A bounding box prediction is also defined in addition.

- The head predicts the bounding box of the object and outputs the center coordinates, width, and height, such that $\{x_{ctr}, y_{ctr}, wd, ht\}$. Then, the predicted bounding boxes (β) are expressed as,

$$\beta_x = \theta(Z_x) + c_x \tag{4}$$

$$\beta_y = \theta(Z_y) + c_y \tag{5}$$

$$\beta_{wd} = P_{wd}.e^{Z_{wd}} \tag{6}$$

$$\beta_{ht} = P_{ht}.e^{Z_{ht}} \tag{7}$$

Where, P_{wd} and P_{ht} represent the width and height of the anchor box, (c_x, c_y) is the coordinate of the top left corner, θ denotes the sigmoid function, and $(Z_x, Z_y, Z_{wd}, Z_{ht})$ are the predicted offsets from the anchor box.

- Now, the probability score is calculated for all bounding boxes. The boxes with minimum probabilityand the bounding boxes that sufficiently overlap are eliminated. The overlap is determined by the expression,

$$IoU = \frac{\rho_\beta \cap T_\beta}{\rho_\beta \cup T_\beta} \tag{8}$$

Where, IoU refers to the Intersection over Union, ρ_β denotes the predicted bounding box, and T_β denotes the ground-truth bounding box.Thus, the bounding box with the maximum score is considered, and the object detected frames $(\aleph)$ are denoted as,

$$\aleph = \{\aleph_{wea}, \aleph_{hum}, \aleph_{oth}\} \tag{9}$$

Where, $\aleph_{wea}, \aleph_{hum}, \aleph_{oth}$ denotes the detected weapons, humans, and other objects.

3.3 Motion Estimation

In the meantime, the motion estimation is carried out in the preprocessed frames $\Re_{pre}$ using the DS algorithm. This process helps in covering all the information of the frames in different orientation angles. DS uses a large diamond search pattern (LDSP) and a small diamond search pattern (SDSP).

Initially, the LDSP is used and the nine search points of DS are tested between the frames to find the least amount of block distortion (MBD). If the least point is found at the edges, then LDSP is continued. If the least MBD is found at the centre, then change LDSP to SDSP.The MBD point identified in the last step is the motion vector's final solution, pointing to the best matching block. The estimated motion for the object is denoted as $\Re_{est}$.

3.4 Processing of Object-Detected Frames

Here, the object-detected frames $(\aleph)$ undergo the following process,

Sliding Window: In this phase, the weapon-detected frames $\aleph_{wea}$ and the estimated motion $\Re_{est}$ undergoes the sliding window process. In this process, the frames are analyzed multiple times by enlarging the sliding window's size. This helps to estimate the motion of different frames. The frames after undergoing the sliding window process are denoted as $\Im_{sw}$.

Silhouette Function: The silhouette is obtained for the frames $\aleph_{hum}$ and $\aleph_{oth}$ to gather depth information about the normal thing with a suspicious appearance. The silhouette function is expressed as,

$$sil(\aleph_{hum}, \aleph_{oth}) = \frac{Y - X}{\max(X, Y)} \tag{10}$$

Where, X and Y denotes the average distance between the frames and the objects, respectively.

3.5 Feature Extraction

In this phase, the features such as Speed Up Robust Features $(SURF)$, Histogram of Oriented Gradient (HOG), geometric feature (GF), texture (Tex), orientation (Or), gradient magnitude channel feature (GMC), and Gray Level Co-Occurrence Matrix $(GLCM)$ features are extracted from the $\Im_{sw}$ and $sil(\aleph_{hum}, \aleph_{oth})$. The extracted feature set (δ) is expressed as,

$$\delta = \{SURF, HOG, GF, Tex, Or, GMC, GLCM\} \tag{11}$$

Now, this extracted set is taken for feature selection.

3.6 Feature Selection

Here, the important features from δ are selected using the CSBO algorithm. BO finds a good solution in a decent amount of time. In BO, if we select the probability without considering any factors, then it directly moves to the local optimum problem.So, the chi-square distribution function is calculated here.

The algorithm starts by initializing the population δ .The member with the highest rank is called as alpha-Bonobo (alp). Fission-fusion strategy entails the formation of several groups within a community. The maximum size (M_s) of the group is formulated as,

$$\mathrm{M}_s = \max(2, \mathrm{M}_{ts} \times N) \tag{12}$$

Where, M_{ts} denotes the size of temporary groups, and N is the population size. The k^{th} member is the best solution if it has the best fitness value than the i^{th} member. Here, the fitness (fit) is based on the maximum classification accuracy (acc) and it is formulated as,

$$fit(\delta) = \max(acc) \tag{13}$$

Now, the new Bonobo (δ^{new}) is formed based on the strategies, such as promiscuous, restrictive mating, consortship, and extragroup mating. This creation is formulated as,

$$\delta_j^{new} = \delta_j^i + rand \times w_1 \times (alp_j - \delta_j^i) + (1 - rand) \times w_2 \times \wp \times (\delta_j^k - \delta_j^i), \quad j = 1,2,3.,\ldots,N \tag{14}$$

Where, w_1 and w_2 are constants, $rand$ is the random number, δ_j^k and δ_j^i are best and random members, and $\wp$ refers to the directional probability, which is computed using the chi-square distribution as,

$$\wp = \frac{1}{\left(\frac{R}{2}\right) 2^{\frac{R}{2}}} (\delta_j)^{\frac{R}{2-1}} e^{\delta_j/2} \tag{15}$$

Where, R is the random number.If the created member is better than the old alp, then the new member is considered as the new alpha. Repeat the process until it satisfies the termination criteria. Finally, the best solution $(\delta_{best})$with the best fitness is selected by CSBO. Algorithm 1 explains the procedure of CSBO.

Algorithm 1: CSBO Technique

Input: feature set (δ)
Output: Important features (δ_{best})

Begin
Initialize the population and algorithm parameters
Select alp,
Evaluate maximum size,

$$\mathrm{M}_s = \max(2, \mathrm{M}_{ts} \times N)$$

For $i = 1\, to\, n$ **d**
Calculate the fitness using equation,

$$fit(\delta) = \max(acc)$$

Create new bonobo,

$$\delta_j^{new} = \delta_j^i + rand \times w_1 \times (alp_j - \delta_j^i) + (1 - rand) \times w_2 \times \wp \times (\delta_j^k - \delta_j^i),$$

Update alp
End For
Save the best candidate solution
Return δ_{best}
End

3.7 Classification

In this phase, the important features (δ_{best}) and the object-detected frames $(\aleph)$ are given to the PELSF-DCNN. The DCNN automatically learns the inputs of the image but it failed to encode the position of the input. So, the positional encoding layer is used here. Further, the Swishsoft activation is used to avoid zero classification results for probabilities. The architecture of PELSF-DCNN is shown in Figure 2.

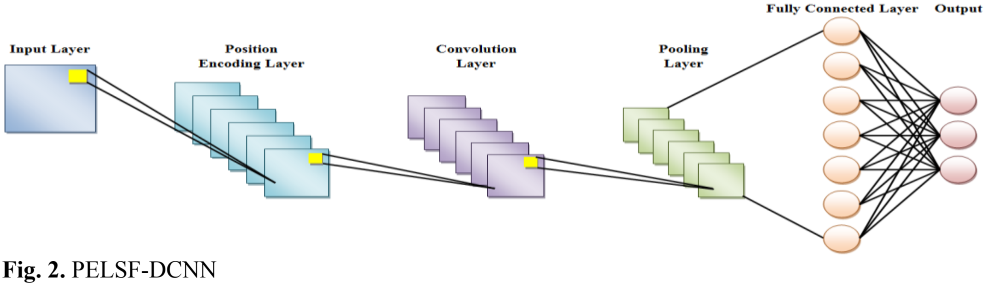


Fig. 2. PELSF-DCNN

Initially, the input is fed into the position encoding layer Ψ_{pos}. This layer is given by sine and cosine variations as,

$$\Psi_{pos}(2\gamma) = \sin\left(\frac{\delta_{best}.\aleph}{n^{\frac{2\gamma}{q}}}\right) \tag{16}$$

$$\Psi_{pos}(2\gamma+1) = \cos\left(\frac{\delta_{best}.\aleph}{n^{\frac{2\gamma}{q}}}\right) \tag{17}$$

Where, q is the dimension of the embedding space and γ is used to map the column indices.

Convolution layer (CL): CL consists of a set of learnable filters called kernels. The convolution operation Ψ_{conv} is expressed as,

$$\Psi_{conv} = \Psi_{pos} * \psi(\alpha,\alpha).\lambda \tag{18}$$

Where, $\psi(\alpha,\alpha)$ denotes the kernel size. The Swishsoft activation function λ is defined as,

$$\lambda(\delta_{best}.\aleph) = \frac{1}{1+sigmoid\left(e^{-\delta_{best}.\aleph}\right)} \tag{19}$$

Pooling Layer (PL): The PL scales the dimensionality of input (Ψ_{conv}) using the max-pooling $(\tau_{\max})$ function as,

$$\Psi_{pool} = \tau_{\max}(\Psi_{conv} * \lambda) \tag{20}$$

Where, Ψ_{pool} denotes the pooling layer output. The output of the final CL and PL is transformed into a one-dimensional array (Ψ_v).

Fully Connected Layer (FCL): Here, every input is connected to every output by a learnable weight. The output of the FCL Ψ_{full} is calculated as,

$$\Psi_{full} = \psi(\alpha,\alpha)\cdot\Psi_v + \kappa \tag{21}$$

Where, κ represents the bias value, and Ψ_{full} is the output. Finally, the proposed PELSF-DCNN classifies the weapons as guns, grenades, knives, and other classes.

3.8 Confidence Score

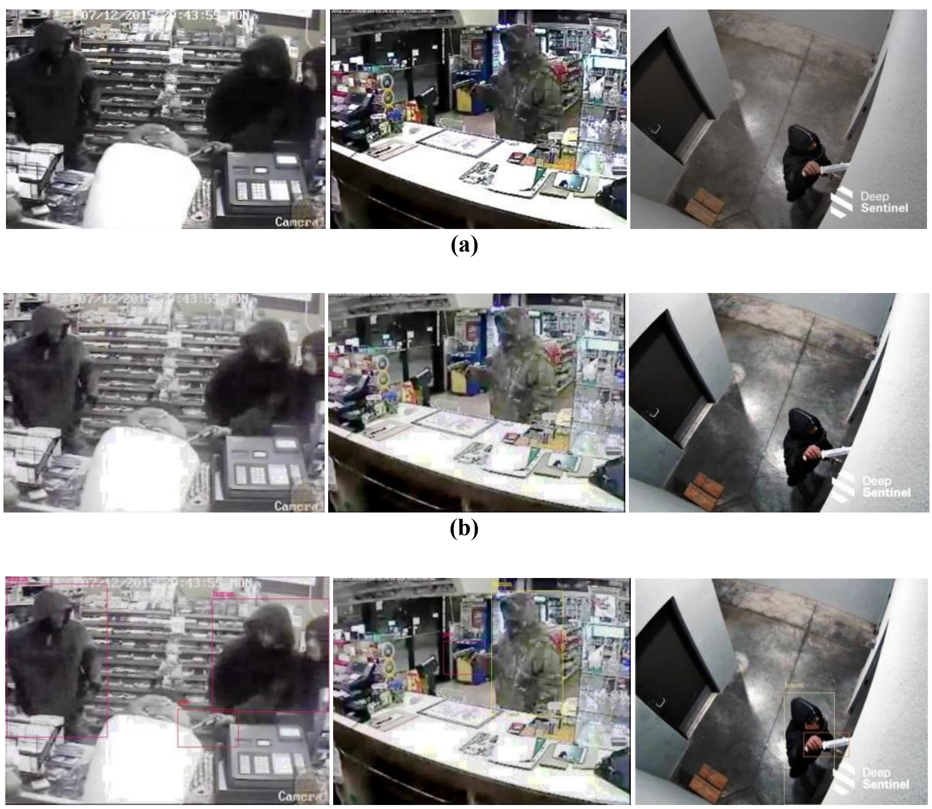
Here, the confidence score is calculated for the classified frames to define the number of weapons. The confidence score (η) is computed with sample mean ω and sample size φ as,

$$\eta = \omega \pm d\left(\frac{\Psi_{full}}{\sqrt{\varphi}}\right) \tag{22}$$

The efficiency of the suggested model is evaluated in the following result section.

4. RESULTS AND DISCUSSION

This section analyzes the performance of the proposed model which is implemented in the working platform of PYTHON. The proposed work used data gathered from publicly available sources. Sample inputs and their corresponding preprocessed and object-detected images are shown in Figure 3.



(a) (b) (c)

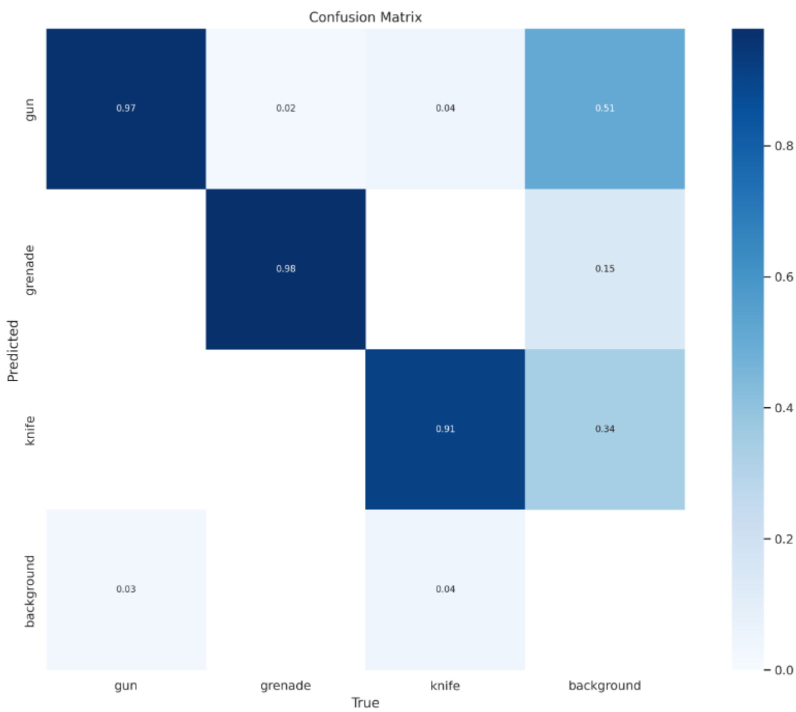
Gun **Gun** **Knife**

Fig. 3. (a) sample input images (b) Preprocessed images (c) objected detected images

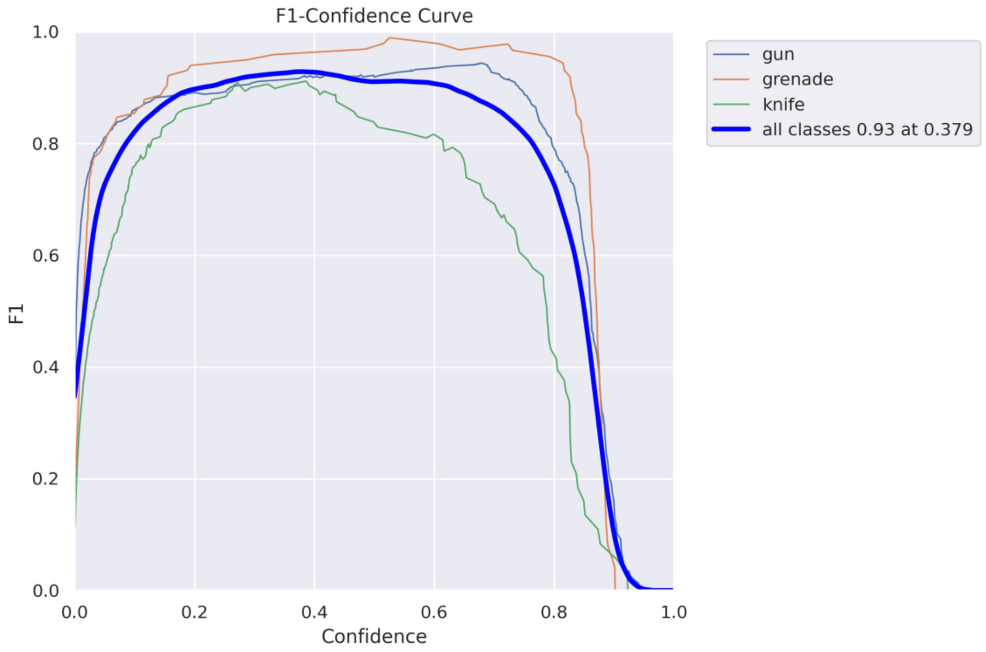
In Figure 3(a), the sample input images are shown; in Figure 3(b), the preprocessed imagesare illustrated; in Figure 3(c), the object-detected frames using YOLO v8 are shown.The proposed PELSF-DCNNhas accurately detected the weapons in the frame (gun, knife).

4.1 Performance Evaluation of YOLO v8

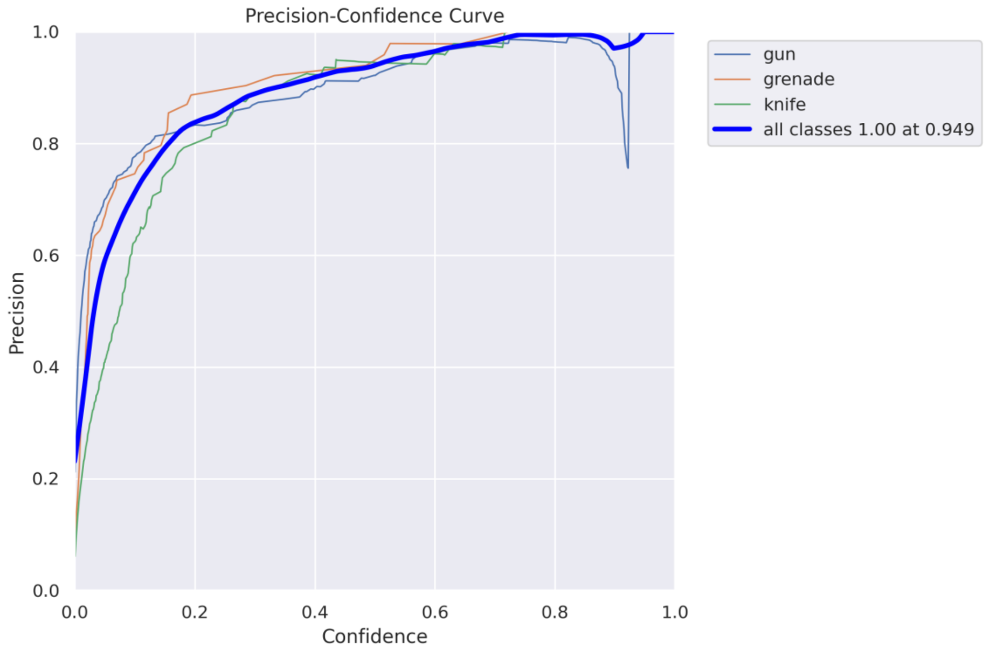
This section evaluates the performance of the proposed YOLO v8 model based on its precision, recall, f1-score, and confusion matrix.



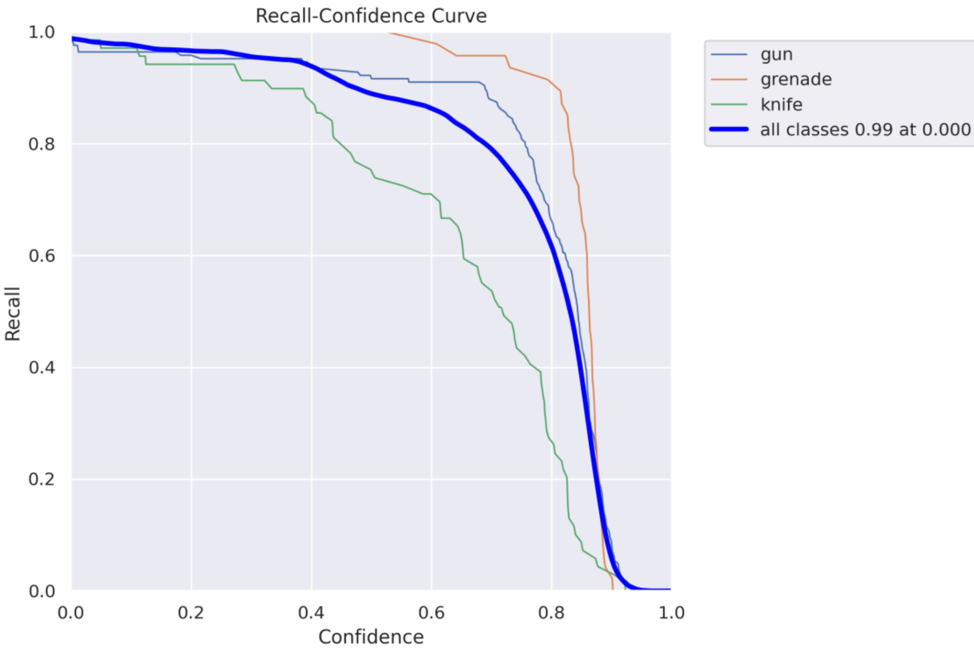
(a)


F1-Confidence Curve
1.0
0.8
0.6
0.4
0.2
0.0
F1
0.0
0.2
0.4
0.6
0.8
1.0
Confidence
gun
grenade
knife
all classes 0.93 at 0.379

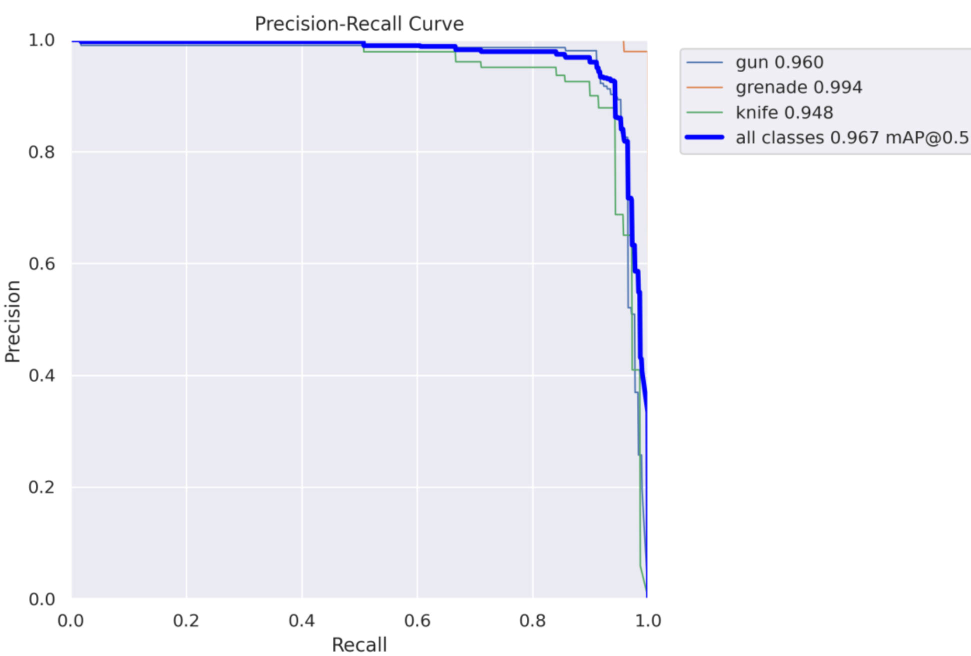
(b)


Precision-Confidence Curve
1.0
0.8
0.6
0.4
0.2
0.0
Precision
0.0
0.2
0.4
0.6
0.8
1.0
Confidence
gun
grenade
knife
all classes 1.00 at 0.949

(c)



(d)



(e)

Fig. 4. (a) Confusion Matrix (b) F1-score (c) Precision (d) recall (e) Precision-recall curve

Figure 4 illustrates the performance of the YOLO v8 model based on various metrics. In Figure 4 (a) confusion matrix is plotted for true and predicted classes. Figure 4 (b) indicates that YOLO v8 has obtained an average F1-score of 0.93 for all classes. Likewise, the YOLO v8 model has obtained better recall and precision of 0.99 and 1, respectively for all classes. This proves the effectiveness of YOLO v8 for object detection.

4.2 Performance Analysis of Classifier

Here, based on several quality criteria, the performance of the proposed PELSF-DCNN is compared with the existing DCNN, Deep Learning Neural Network (DLNN), Recurrent Neural Network (RNN), and Artificial Neural Network (ANN).

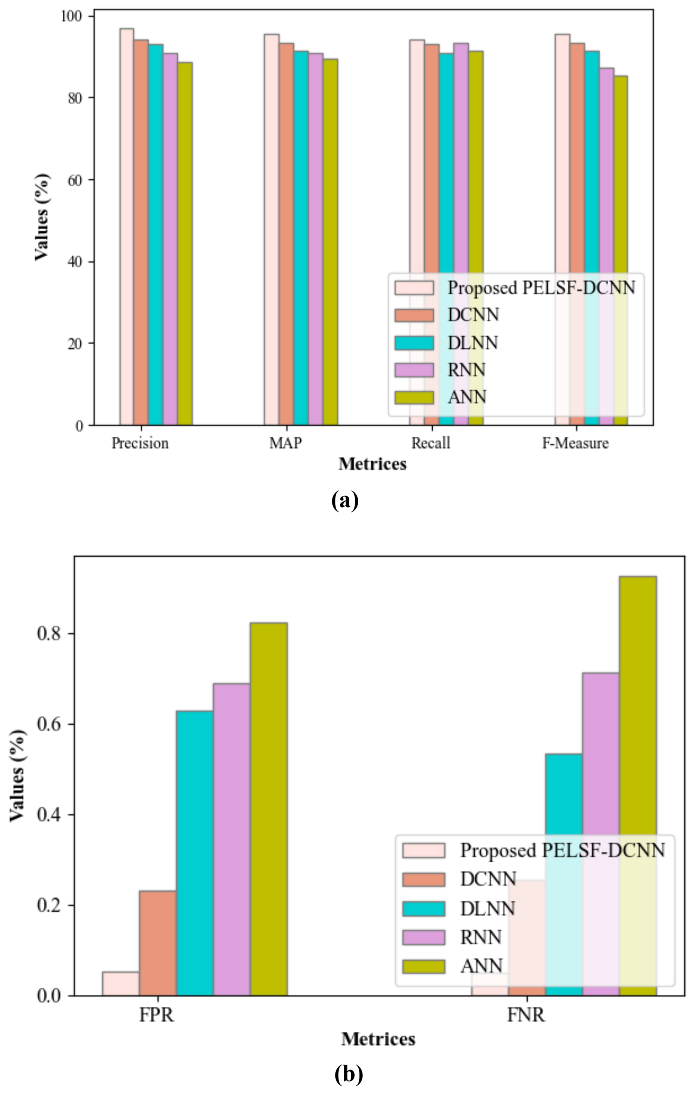


Fig. 5. Performance analysis of proposed PELSF-DCNN based on (a) precision, recall, mAP, f-measure (b)False Positive Rate (FPR),False Negative Rate (FNR)

In above figure 5, the performance of the proposed and the existing methods are analyzed. The proposed method achieves higher performance values of 96.8%% (precision), 95.42% (mAP), 95.48% (f-measure), and 94.2% (recall). But, the conventional DCNN, DLNN, RNN, and ANN methods offer lower values. The proposed model has neglected the false prediction to obtain lower FNR and FPR. Overall, the extra layer in the proposed Model has enhanced the classification process.

Table 1: Performance comparison of Proposed PELSF-DCNN

Techniques	Accuracy (%)
Proposed PELSF-DCNN	97.5
DCNN	95.34
DLNN	91.34
RNN	90.56
ANN	86.95

The performance of the proposed PELSF-DCNNbased on accuracy is shown inTable 1. The proposed model attains 97.5% of accuracy, which is higher when compared to the existing models. The inclusion of an extra layer and Swishsoft activation in DCNN proved the superiority of the proposed model.

4.3 Performance Analysis of CSBO

The performance of the proposed CSBO algorithm is compared with the techniques like BO, White Shark Optimization (WSO), Jellyfish Optimization (JO),and Salp Swarm Optimization (SSO) based on fitness vs. iteration.

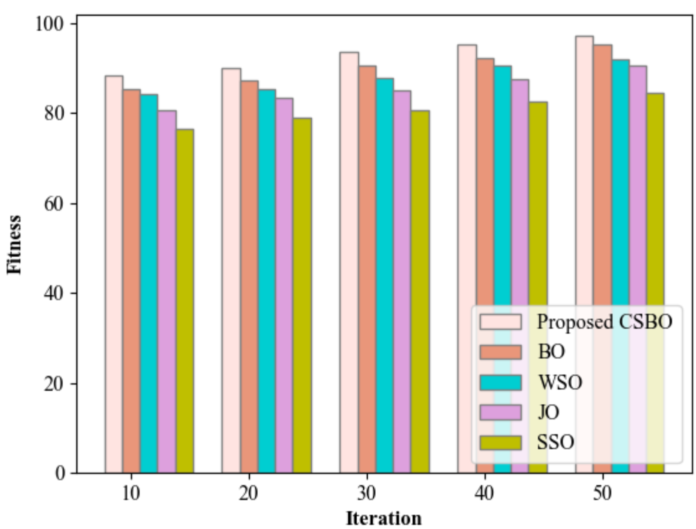


Fig. 6. Performance analysis of Proposed CSBO

The fitness vs. iteration of the proposed and existing models is shown in Figure 6. The fitness is based on the accuracy of the classifier. So, fitness should be high for an efficient model. The proposed model obtained a fitness value of 97.2 for 250 iterations. Likewise, figure 6 clearly indicates that the proposed model has obtained high fitness values in all iterations.

4.4 Performance Analysis of Pre-processing

With the use of existing techniques like MSR, Contrast-limited Adaptive Histogram Equalisation (CLAHE), AHE, and HE based on Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), and Structural Similarity Index (SSIM), the performance of the proposed PDFMSR is evaluated.

Table 2: Performance analysis of proposed PDFMSR

Techniques/Metrics	PSNR	MSE	SSIM
Proposed PDFMSR	43.12	9.8	0.84
CLAHE	39.42	13.67	0.77
AHE	38.92	14.23	0.72
HE	32.45	16.7	0.65

Table 2 analyses the performance of the proposedPDFMSRand existing methods. The PSNR value of the proposed is 3.7dB higher than the existing CLAHE method. The SSIM value shows a greater difference of 0.77 compared with the existing CLAHE. Likewise, the proposed PDFMSRexhibits better performance by obtaining a lower MSE value than the existing methods.

4.5 Comparative Analysis with Previous Approaches

Here, the performance of the proposed weapon detection methodology is compared with traditional approaches suggested by (Ahmed et al., 2022), (Fathy & Saleh, 2022), and (Bhatti et al., 2021) based on its mAP.

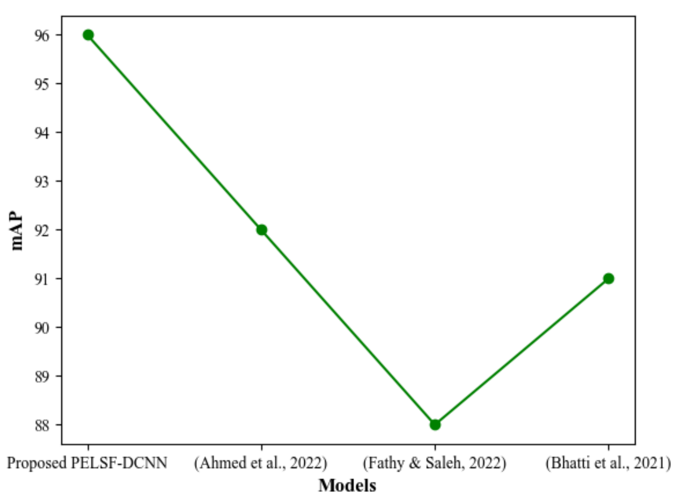


Fig. 7. Comparative Analysis

Figure 7 shows the performance of the proposed framework based on its mAP. The proposed model has obtained a mAP value of 95.42%, whereas the existing models attained lower values of 92.1 % (Ahmed et al., 2022), 88% (Fathy & Saleh, 2022), and 91.7% (Bhatti et al., 2021). Thus, it is clear that the overall performance of the proposed methodology is better.

5. CONCLUSION

This paper proposes a PELSF-DCNN model for Weapon detection. The proposed method undergoes 8 phases. After executing all the steps, the experimental analysis is performed to prove the effectiveness of the proposed model. The final outcomes reveal that the proposed model achieves an accuracy of 97.5%, and a precision of 96.8%. YOLO v8 has also achieved better results. Similarly, the proposed model achieves the best result for all other metrics. Thus, it is concluded that the proposed model is highly efficient for WD. The work detected guns but it could differentiate between real and dummy guns. So, in the future, the work will be extended by providing a new ensemble classifier to differentiate real guns and detect other dangerous weapons.

REFERENCES

1. Ahmed, S., Bhatti, M. T., Khan, M. G., Lövström, B., & Shahid, M.*, Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos*, Proceedings of Applied Sciences (Switzerland), 12(12). (2022).
2. Ashraf, A. H., Imran, M., Qahtani, A. M., Alsufyani, A., Almutiry, O., Mahmood, A., Attique, M., & Habib, M. *Weapons detection for security and video surveillance using CNN and YOLO-V5s*. Proceedings of Computers, Materials and Continua, 70(2), 2761–2775. (2022).
3. Baig, M. S., & Khan, P. A., *Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications.* Proceedings of International Journal of Advanced Research in Science and Technology, 12(10), 127–135. (2020)
4. Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J., *Weapon Detection in Real-Time CCTV Videos Using Deep Learning*. Proceedings of IEEE Access, 9, 34366–34382. (2021).
5. Fathy, C., & Saleh, S. N., *Integrating Deep Learning-Based IoT and Fog Computing with Software-Defined Networking for Detecting Weapons in Video Surveillance Systems*. Sensors, 22(14). (2022).
6. Galab, M. K., Taha, A., & Zayed, H. H., *Adaptive Technique for Brightness Enhancement of Automated Knife Detection in Surveillance Video with Deep Learning,* Arabian Journal for Science and Engineering, 46(4), 4049–4058. (2021)
7. Gawade, S., Vidhya, R., & Radhika, R., *Automatic Weapon Detection for Surveillance Applications*. Proceedings of the International Conference on Innovative Computing and Communication, 1–6. (2022)
8. Hashmi, T. S. S., Haq, N. U., Fraz, M. M., & Shahzad, M. *Application of Deep Learning for Weapons Detection in Surveillance Videos*. Proceedings of International Conference on Digital Futures and Transformative Technologies, October. (2021)

9. Jain, H., Vikram, A., Mohana, Kashyap, A., & Jain, A *Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications*, Proceedings of the International Conference on Electronics and Sustainable Communication Systems, 193–198. (2020)
10. Salido, J., Lomas, V., Ruiz-Santaquiteria, J., & Deniz, O. *Automatic handgun detection with deep learning in video surveillance images*. Applied Sciences, 11(13), 1-17 (2021)
11. Hnoohom, N., Chotivatunyu, P., Maitrichit, N., Sornlertlamvanich, V., Mekruksavanich, S., & Jitpattanakul, A. *Weapon Detection Using Faster R-CNN Inception-V2 for a CCTV Surveillance System*. Proceedings of the 25th International Computer Science and Engineering Conference, 400–405. (2021).
12. Ekmal, M., Quyyum, E., Haris, M., & Abdullah, L. *Proceedings of the Multimedia University Engineering Conference.* Atlantis Press International BV. (2023).
13. Kaya, V., Tuncer, S., & Baran, A. *Detection and classification of different weapon types using deep learning*. Applied Sciences, 11(16), 1-13. (2021)
14. Xu, S., & Hung, K, *Development of an AI-based System for Automatic Detection and Recognition of Weapons in Surveillance Videos*, proceedings of the IEEE 10th Symposium on Computer Applications and Industrial Electronics, 48–52. (2020).
15. T Hamsini, Lokhande, H. V, NithisiriS, & L, R., *A Review on Weapon Detection and Alert System Using Deep Neural Networks*, proceedings of International Research Journal of Modernization in Engineering Today and Science, 4(06), 410–413. (2022).