Received: February 8, 2022. Revised: April 19, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Weapon Detection in Surveillance Videos Using Human Inspired Particle Swarm Optimization Algorithm and Support Vector Machine

Kiran Kalla[1]* **Gogulamanda Jaya Suma[2]**

[1]*Ramachandra College of Engineering, Eluru, India*
[2]*JNTUK-University College of Vizianagaram, Vizianagaram, India*
* Corresponding author's Email: Kallakiran1974@gmail.com

Abstract: In recent decades, automatic control systems are becoming the essential need for security forces, due to the increase in the number of criminal activities. The fast and precise automatic weapon detection system is useful to mitigate or avoid risks in public spaces. In this manuscript, a new automated model is implemented for effective weapon detection in closed circuit television videos. After collecting the data from YouTube and Gun movies databases, the Gaussian Mixture Model (GMM) is applied to detect the weapons in the video sequences. Then, the feature extraction is performed using deep learning models: AlexNet and ResNet 18, and a descriptor: Scale Invariant Feature Transform (SIFT) for extracting the feature vectors from the segmented regions. Whereas, the combination of deep and texture features reduces the semantic space between the feature sub-sets that helps in enhancing the classification performance. In addition, the feature optimization is accomplished by Human Inspired Particle Swarm Optimization (HIPSO) algorithm to select active feature vectors that decrease the system complexity and training time of the classifier. In the conventional PSO algorithm, the Human Group Optimization (HGO) algorithm is utilized to influence the particles, and then the adaptive uniform mutation is utilized to improve the convergence rate and makes the implementation simple. Finally, the selected active feature vectors are fed to the Support Vector Machine (SVM) classifier for weapon and non-weapon classification. The experiment results confirmed that the HIPSO-SVM model has achieved high accuracy of 95.34% and 98.60% on the YouTube and Gun movies databases, which are better compared to the existing models.

Keywords: AlexNet, Gaussian mixture model, Particle swarm optimization algorithm, ResNet 18, Support vector machine, Weapon detection.

1. Introduction

The rapid and precise detection of weapons in public places is necessary to mitigate or avoid risks. In this application, the closed-circuit television is widely used for recognizing dangerous situations, where the closed-circuit television is considered as the effective operational requirement in terms of safety aspects [1-3]. The main purpose of closed circuit television is to provide security, crime investigation, deterrence, and reduction in insurance costs [4, 5]. However, the deterrence effects of closed circuit television cameras vary from the different time periods and crime categories. Usually, the human operator handles the weapon detection task, which is ineffective, due to visual distraction or fatigue [6, 7]. In addition, the increasing number of areas controlled by video cameras and the factors inherent to human conditions like loss of attention and fatigue make these systems inefficient [8]. Therefore, intelligent systems are developed by researchers for the automatic detection of risk situations or threats involving firearms [9, 10]. The intelligent systems are effective in the situations such as terrorist attacks, gunfire incidents on school grounds, mass shooting, and handgun attacks [11, 12]. This article uses HIPSO-SVM model for improving the performance of weapon detection and the major contributions are listed below:

- After collecting the video sequences from YouTube and Gun movies databases, the

weapon detection is performed utilizing the GMM technique.

- Then, the feature extraction is carried out utilizing AlexNet, ResNet 18, and SIFT models for extracting the deep and textual feature vectors from the segmented regions, and further, HIPSO algorithm is introduced for diminishing the dimensions of extracted feature vectors that enhances the system complexity and training time of the classification technique. As mentioned earlier, the HGO algorithm effectively influences the particles of PSO algorithm by performing the adaptive uniform mutation, which enhances the convergence rate and makes implementation simple.
- At last, the selected active feature vectors are given as the input to the SVM classifier for classifying the weapon and non-weapon classes. Additionally, the proposed HIPSO-SVM model's performance is examined in terms of f-score, recall, precision, and classification accuracy.

This manuscript is structured as follows: a few articles on the research topic "weapon detection" are surveyed in Section 2. Theoretical explanation and the experimental evaluations of the HIPSO-SVM model are represented in Section 3 and 4. The summary of this manuscript is denoted in Section 5.

2. Related works

Narejo [13] developed the You Only Look Once (YOLO) V3 model for weapon detection in surveillance videos. The presented YOLO V3 model superiorly detects the unsafe assets and weapons in the high-end security and surveillance videos compared to the conventional pre-trained Convolutional Neural Network (CNN) model named YOLO V2. However, the developed YOLO V3 model requires high computation resources and intensive graphics processing units for training the data, which was considered a major issue in this literature. Kaya [14] implemented the Visual Geometry Group (VGG)-19 model for detecting and classifying seven weapon types in the surveillance videos. The presented VGG-19 model obtained a superior performance in weapon detection compared to other deep learning models such as ResNet-101, ResNet-50, and VGG-16. In addition, González [15] integrated faster R-CNN with ResNet-50 model for real-time gun detection in closed-circuit television videos. However, the VGG-19 and ResNet-50 models were computationally expensive, because it needs an enormous amount of data for model training.

el den Mohamed [16] integrated GoogLeNet and AlexNet models for detecting the guns and pistols in the closed-circuit television videos. The usage of transfer learning and deep learning techniques effectively improves the over-all detection speed and accuracy. Similar to the prior literature, Salido [17] integrated YOLO V3, RetinaNet, and faster R-CNN models for an effective handgun detection in the video surveillance images. As mentioned earlier, the computational complexity of the hybrid deep learning model was higher compared to existing machine learning methods. Olmos [18] implemented faster R-CNN model for automatic handgun detection in the videos. By investigating the obtained results, the presented faster R-CNN achieved satisfactory results in the low quality YouTube video sequences. However, the developed surveillance and control systems still need human intervention and supervision.

Velasco-Mata [19] integrated YOLO V3 detector with individual subjects' pose information to enhance over-all performance of handgun detection. In this literature, the developed model integrates handgun detector output and heat-map-like images for detecting the handguns in the video sequences. The presented model showed improvement in the handgun detection related to the original handgun detector. As stated previously, the YOLO V3 detector needs high intensive graphics processing units and more computation resources for data training. J. Ruiz-Santaquiteria [20] combined both weapon appearance and human pose information for handgun detection. However, the developed model showed only comparable results in the factors like camera distance, poor occlusions, and lighting conditions. Grega [21] used canny edge detector to segment knives and firearms in closed-circuit television videos. Secondly, the MPEG-7 homogeneous texture descriptor was employed to extract feature vectors from the segment regions and then an SVM classifier was employed for weapon classification. The MPEG-7 homogeneous texture descriptor consists of the standard deviation, energy, and mean value of an image. However, the extracted feature vectors were multi-dimensional increasing the system complexity and running time of the SVM classifier. For highlighting the aforementioned concerns, a new HIPSO-SVM model is introduced in this manuscript for effective weapon detection.

3. Methodology

In the lethal weapon detection, the proposed HIPSO-SVM model includes five major phases such as data collection: YouTube and Gun movies

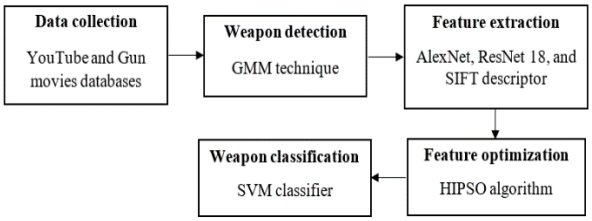

Figure. 1 Flowchart of the proposed HIPSO-SVM model

Figure. 2 Sample frames of YouTube database

Figure. 3 Sample frames of gun movies database

databases, weapon detection: GMM, feature extraction: AlexNet, ResNet 18, and SIFT feature descriptor, feature optimization: HIPSO, and classification: SVM classifier. The flowchart of the proposed HIPSO-SVM model is specified in Fig. 1.

3.1 Data collection

In lethal weapon detection, the proposed HIPSO-SVM model's effectiveness is validated on YouTube and gun movies databases. In the YouTube database, the video sequences are captured during shooting practice sessions, and it comprises 12 video sequences with 952 frames/images of pixel size 1920×1080 . In this database, the video sequences are recorded at different shooting poses, camera locations, lighting conditions, and background scenarios. Further, the Gun movies database is a video database that is captured from security and surveillance closed-circuit television cameras. This database mimics the gun shooting conditions, due to the non-availability of real time videos. The Gun movies database comprises seven video sequences with 24,000 frames of pixel size 640×480. The sample frames of YouTube and gun movies databases are stated in Fig. 2 and 3.

3.2 Lethal weapon detection

After data collection, lethal weapon detection is accomplished by utilizing the GMM technique. It is fundamentally used as a parametric technique that calculates the probability density function on different object features. In the computer vision application, it is hard to find the moving objects in the dynamic scene changes and severe occlusion. For identifying the moving objects in the video sequences, the background subtraction method is undertaken in this article, where the background modeling is done utilizing GMM. To achieve the ideal outcome, every frame in the video sequences is subtracted from a reference frame. Next, match the dissimilarity between the reference and incoming frame to segment the foreground regions from the background regions. The GMM technique effectively identifies and tracks the objects, because it includes intensity and color based methods for background subtraction from the foreground regions. The steps associated with GMM are listed as follows:

Step 1: Distinguish each input pixel for mean μ of components. If the pixel value is nearer to the mean of selected component, the specific component is considered as the compatible component. To be a compatible component, the difference of pixel and mean obtained should be less and it is matched with the standard deviation of scaling factor D.

Step 2: Next, the mean, Gaussian weights, and standard deviation (variance) variables are updated to replicate the obtained new pixel values. Further, the components that are non-matched decrease to weight w and the mean and standard deviation will not change that relied on learning component p to state the instant alterations.

Step 3: Categorize the components, which are the portions of background model. To accomplish this task, a threshold value is utilized as a component weight w.

Step 4: Regulate the pixels of foreground regions. Here, the recognized pixels as foreground will not be suitable with any other components from the background regions.

3.2.1. Background modelling

The GMM technique is parameterized by mixture component weight, mean, and variance. The GMM with $k^{th}the$ component has a variance of σ_k and covariance matrix of $\sum_k$ for the multivariate cases and a mean of μ_k for the univariate cases. For component C_k, the mixture component weights are indicated as ϕ_k with the constraint $\sum_{i=1}^{k} \phi_i = 1$, so the total probability distribution normalizes to 1, and ϕ_k is considered as a priori distribution over the components, if the component weights are not learned. Each pixel of the background region is modeled by a separate mixture of k Gaussians, as represented in the Eqs. (1) and (2).

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} \times \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

Where,

$$\sum_{i=1}^{k} \omega_{i,t} = 1 \quad (2)$$

Mean: The mean of mixtures is indicated in Eq. (3).

$$\mu_t = \sum_{i=1}^{k} \omega_{i,t}, \mu_{i,t} \quad (3)$$

Where, k represents a number of Gaussians, which generally ranges between 3 to 5, $\Sigma_{i,t}$ represents covariance matrix, $\mu_{i,t}$ states mean value of i^{th} Gaussian in the time instant t, X_t represents present pixel value, and $\omega_{i,t}$ estimates the weight of i^{th} Gaussian.

Variance: The variance of a random variable X is considered as the expected variable of the square deviation from the mean of X, $\mu = E[X]$, which is mathematically represented in Eq. (4). The variance is also considered as the covariance of a random variable X, as mentioned in Eq. (5).

$$Var(X) = E[(X - \mu)^2] \quad (4)$$

$$Var(X) = Cov(X) \quad (5)$$

3.2.2. Parameter estimation of k-Gaussian distribution

The k-Gaussian distribution parameters include variance, weight and mean, which are needed to be estimated. The weight and mean are initialized to zero, and the variance is set to a large value V_0. At every time instant t, each new pixel X_t is matched with the existing k pixels, until a match is found. A match is distinct as a pixel value X_t inside 2.5 standard deviations of a distribution. In the unmatched Gaussian distributions, the μ and Σ parametric values are similar, and the matched parameters of Gaussian G_i in the mixture X_t is updated as mentioned in the Eqs. (6) to (8).

$$\mu_{i,t} = (1 - \rho) \times \mu_{i,t-1} + \rho \times X_t \quad (6)$$

$$\sum_{i,t} = (1 - \rho) \times \sum_{i,t-1} + \rho \times diag[(X_t - \mu_{i,t}) \quad (7)$$

Where

$$\rho = \alpha \times \eta(X_t | \mu_{i,t-1}, \sum_{i,t-1}) \quad (8)$$

Where, G_j represents probable distribution, $diag|x|$ denotes diagonal matrix, and α states learning rate. The present pixel value X_t is re-assigned, if none of the k Gaussians is matched.

3.2.3. Classification of foreground and background regions

The Gaussians generated by the background process are determined after the parameters of every pixel model are updated. Initially, the Gaussians are ordered by ω, so the background distributions remain on top and the less background distributions moves towards the bottom, and are then replaced by new distributions. Then, the B distributions are selected as the background model as mentioned in Eq. (9).

$$B = arg\, min_b(\omega > T) \quad (9)$$

Where, T indicates threshold value, which ranges between $0.5 < T < 1$. After estimating the k-Gaussian distribution, the background and foreground pixel classification is carried out with some confidence interval of its distribution's mean. The formulas of foreground and background regions are mentioned in Eqs. (10) and (11).

$$\frac{|(I_t) - \mu_t|}{\sigma_t} > k \quad \text{Foreground} \quad (10)$$

$$\frac{|(I_t) - \mu_t|}{\sigma_t} \leq k \quad \text{Background} \quad (11)$$

Where, k represents a free threshold value, and a small k upsurges the probability of transition from foreground to the background regions, due to subtle change. Whereas, a large k value allows a more dynamic background. In another method, a pixel distribution is updated, if the pixel is classified as background that prevents foreground objects from fading into the background. The updated formula is mentioned in Eq. (12).

$$\mu_t = M\mu_{t-1} + (1 - M)(I_t\rho + (1 - \rho)\mu_{t-1}) \quad (12)$$

In this scenario, $M = 1$, when the pixel is foreground and $M = 0$ when the pixel is background. When the pixel is detected as foreground, the mean value remains the same, and the pixel is considered as a background pixel only when the intensity value gets closer to the value before the pixel became foreground pixel. In addition, the unmatched image pixels are considered as foreground pixels that are grouped using 2D component analysis, either using

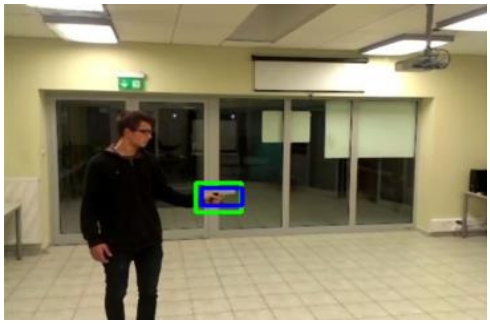

Figure. 4 Sample lethal weapon detected frame

eight-pixel or four-pixel connectivity. At any time t, a particular pixel (x_0, y_0) is given in Eq. (13).

$$X_1, \ldots\ldots, X_t = k(x_0, y_0, i): 1 \leq i \leq t \qquad (13)$$

The history is modeled by a mixture of k Gaussian distributions, as stated in Eqs. (14) and (15).

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} N(X_t \left| \mu_{i,t}, \sum_{i,t} \right.) \qquad (14)$$

Where,

$$N(X_t \left| \mu_{i,t} \sum_{i,t} \right.) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{\left| \sum \quad_{i,t} \right|^2}$$
$$exp\ (-\frac{1}{2}(X_t - \mu_{i,t})^T \sum_{i,t}^{-1} (X_t - \mu_{i,t})) \qquad (15)$$

Where, the classification of foreground and background regions is performed through k-Gaussian distribution. The Lethal weapon in the video sequences is detected, since it is the foreground. The sample lethal weapon detected frame is indicated in Fig. 4.

3.3 Feature extraction

After the detection of lethal weapons, the feature extraction is carried out by utilizing the deep learning techniques: AlexNet and ResNet 18, and feature descriptor: SIFT. The ResNet 18 has 1 fully connected layer with softmax classifier, 5 convolutional layers, and 1 average pooling layer that superiorly extracts the deep feature vectors for better convergence behaviors, and compelling accuracy. Hence, the ResNet 18 and AlexNet are pre-trained convolutional neural networks, where the segmented images are resized to the pixel size of 227×227. The AlexNet model comprises of 8 layers like 5 convolutional layers and 3 fully connected layers, where every convolutional layer is followed by a max-pooling operation, and every fully connected layer is followed by a rectifier linear unit for extracting deep feature vectors from the segmented regions. In the AlexNet model, the rectifier linear unit is used for replacing the negative feature maps by zero, and the max pooling operation is used for decreasing the feature maps.

Correspondingly, the SIFT feature descriptor is to identify the key points and locations of the frame at which feature is invariant to rotation and scale. The SIFT feature descriptor comprises four phases: key point descriptor, assignment of orientation, extrema detection in scale space, and location of key points. The scale-space extrema detection is utilized for extracting the multi-scale feature vectors from the segmented images. Here, the SIFT feature is accomplished by scale space function based on the Gaussian function. Further, Gaussian difference is estimated by localizing scale space extrema, identifying the locations of key points and calculating the dissimilarity between the two successive images/frames. In the key point localization, the key points are located by choosing the local extrema and the candidate points are stable under the Gaussian space. For identifying the orientation using the key points, select the Gaussian smoothed images by computing the gradient magnitude. Finally, the descriptor is generated based on the scale, orientation, and location of the key points, once the key points are located. The extracted feature vectors of the AlexNet=512, ResNet 18=3040, and SIFT=1027 models are integrated by utilizing feature level fusion, and then the total extracted feature vectors are fed to the HIPSO for feature optimization.

3.4 Feature optimization

After extracting the feature vectors using AlexNet, ResNet 18, and SIFT feature descriptor, the feature optimization is performed using the HIPSO algorithm for selecting the discriminative feature vectors that reduce the system complexity and running time of the classification method. The traditional PSO algorithm mimics the behavior of birds, and it is a population-based searching optimization algorithm. The Eqs. (16) and (17) are used for updating the velocity and position of the particles.

$$v_{id}(t+1) = I_w \times v_{id}(t) + c_1 \times r_1 \times [p_{id}(t) - x_{id}(t)] + c_2 \times r_2 \times [p_{gd}(t) - x_{id}(t)] \qquad (16)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (17)$$

Where, I_w represents inertia weight that balances global and local search, t states iteration number, c_1 and c_2 indicates acceleration coefficients, $r_1\ and\ r_2$ specifies two random numbers, p_{id} denotes particles current best position, and p_{gd}

indicates global best position. In the HIPSO algorithm, the human group optimization algorithm is employed for influencing the particles and further, the adaptive uniform mutation is applied to enhance the convergence rate and to make implementation simple.

At first, the human group optimization algorithm is applied to convert the discrete multiple labels into continuous labels. The HIPSO algorithm optimizes the extracted feature vectors based on d_i, where the feature vectors of the position of the particle are represented as $p_i(t) = (p_{i,1}, p_{i,2}. p_{i,D})$. Further, the adaptive uniform mutation (fitness function) is applied to extend the capability of feature optimization in the exploration. In this process, a non-linear function p_m is applied to control the decisions and range of the mutation on every particle. At every iteration, p_m is updated by the Eq. (18).

$$p_m = 0.5 \times e^{(-10 \times \frac{t}{T})} + 0.01 \quad (18)$$

Where, T denotes maximum iteration, and the p_m value tends to reduce, while the iteration number increases. If the p_m value is larger than the random number between [0,1], the mutation randomly picks the discriminative feature vectors (3451) that are fed to the SVM for weapon classification. The parameter setting of HIPSO algorithm is given as follows: maximum number of iterations is 100, population size is equal to total extracted feature vectors, social constant c_1 is 3, and cognitive constant c_2 is 2.

3.5 Weapon/non-weapon classification

The selected discriminative feature vectors are used for classification by employing SVM to classify the weapon/non-weapon classes. The SVM classifier is a supervised classifier, where it has a discriminative characteristic of hyperplane for image classification. The vapnik-chervonenkis and the structure principles resolve the two-class limitations in the SVM classification method. The discriminant function is linear and the formula is specified as $W_f \times x_f + a_f = 0$. The optimal hyper-plane distinguishes the classes without noise, and it is expressed in Eq. (19).

$$H_{pi}[W_f \times x_f + a_f] - 1 \geq 0, i = 1,2, \ldots \quad (19)$$

Then, reduce $\|W_f\|^2$ in Eq. (19), so that the problem of optimization is solved using Lagrange function ϑ_i, which is mathematically expressed in Eq. (20).

$$f(x_f) = sign\{(W_f{}^* x_f) + a_f{}^*\} = sign\{\sum_{i=1} \vartheta_i^* \times H_{pi}(x_{fi}^* - x_f) + a_f{}^*\} \quad (20)$$

Finally, change the interior-product $(x_{fi}^* - x_f)$ obtained from the linear function $K(x_f, x_f{}')$ in Eq. (20) that decreases the computational complexity in high dimensional data. The sample obtained from the discriminant function is separable, which is rewritten in Eq. (21). The radial basis function is used as a kernel function in SVM that is mathematically denoted in Eq. (22).

$$f(x_f) = sign\{\sum_{i=1} \vartheta_i^* \times H_{pi} \times K(x_f, x_f{}') + a_f{}^*\} \quad (21)$$

$$K(x_f, x_f{}') = exp\left[-\frac{\|x_f, x_f{}'\|^2}{2\sigma^2}\right] \quad (22)$$

4. Experimental results

In this manuscript, the HIPSO-SVM model is simulated using Python software tool on the system configuration with Intel Core i9 processor, Linux operating systems, 8 TB hard disk, and 16 GB random access memory. The proposed HIPSO-SVM model effectiveness is evaluated on two online databases such as YouTube and gun movies databases using f-score, accuracy, recall, and precision. The performance measures: f-score, accuracy, recall, and precision are used as a regular measurement of results that generates reliable data on the efficiency and effectiveness of the HIPSO-SVM model. The mathematical representation of the undertaken performance measures is depicted in the Eqs. (23) to (26).

$$F - score = \frac{2TP}{FP+2TP+FN} \times 100 \quad (23)$$

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (24)$$

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (25)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (26)$$

Where, FP, FN, TP, and TN are indicated as a false positive, false negative, true positive, and true negative.

4.1 Quantitative evaluation

In Table 1, the effectiveness of the HIPSO-SVM model is validated on YouTube database that

Table 1. Experimental results of HIPSO-SVM model on the YouTube database

Classifiers	Optimizers	F-score (%)	Recall (%)	Precision (%)	Accuracy (%)
Naïve Bayes	DOA	80.89	80.90	80.80	80.08
	GOA	82.35	87.37	83.69	82.40
	ACO	87.96	87.58	87.77	87.50
	PSO	89.88	90.30	90.24	88.58
	HIPSO	90.38	90.53	90.70	90.82
Decision tree	DOA	77.30	68.68	78.77	82.83
	GOA	77.80	78.38	79.50	85.26
	ACO	85.48	79.60	79.90	86.60
	PSO	86.09	88.20	86.50	87.88
	HIPSO	87.10	89.99	87.80	88.80
Random forest	DOA	78.54	70.98	78.90	88.30
	GOA	76.40	80.78	79.82	88.40
	ACO	87.09	82.44	80.58	90.67
	PSO	88.88	88.76	84.87	91.20
	HIPSO	90.28	90.03	90.50	91.78
SVM	DOA	88.49	86.40	90.97	89.50
	GOA	87.90	88.90	93.42	90.55
	ACO	89.74	90.36	94.20	92.28
	PSO	92.76	92.30	96.87	92.90
	HIPSO	95.40	96.27	98.58	95.34

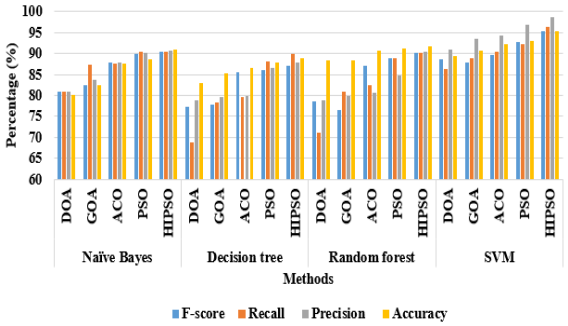

Figure. 5 Comparison results of the HIPSO-SVM model on the YouTube database

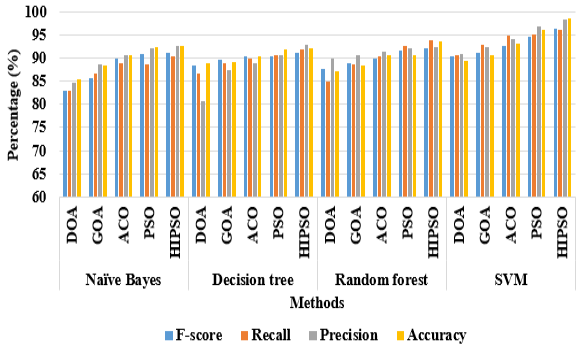

Figure. 6 Comparison results of the HIPSO-SVM model on the Gun movies database

includes 952 frames/images in that 80:20% of data are used model training and testing. In addition, the fivefold cross-validation is applied for further estimating the effectiveness of the HIPSO-SVM model. In this article, the experimentation is performed with several machine-learning classifiers: Naïve Bayes, decision tree, random forest, and SVM, and feature optimization algorithms: HIPSO, PSO, Dragonfly Optimization Algorithm (DOA), Ant Colony Optimization (ACO) algorithm, and Grasshopper Optimization Algorithm (GOA). As represented in Table 1, the combination: HIPSO with SVM achieved better performance in the weapon detection compared to other classifiers, and feature optimization algorithms. In the YouTube database, the HIPSO-SVM model has achieved 95.34% of accuracy, 95.40% of f-score, 96.27% of recall, and 98.58% of precision in weapon detection. A graphical presentation of the HIPSO-SVM model on the YouTube database is depicted in Fig. 5.

In Table 2, the effectiveness of the HIPSO-SVM model is investigated on Gun movies database using f-score, recall, precision, and accuracy. By investigating Table 2, the proposed HIPSO-SVM model obtained superior performance in the weapon detection compared to other classifiers, and feature optimization algorithms. As seen in Table 2, the proposed HIPSO-SVM model obtained 98.60% of accuracy, 96.45% of f-score, 96.20% of recall, and 98.56% of precision in weapon detection. Graphical presentation of the HIPSO-SVM model on the Gun movies database is stated in Fig. 6.

4.2 Comparative evaluation

The comparison result of the HIPSO-SVM model and the existing models is depicted in table 3. Velasco-Mata [19] combined the YOLO V3 detector with the subject's pose information for enhancing handgun detection. The experimental analysis showed that the presented model attained 76.60% of

Table 2. Experimental results of HIPSO-SVM model on the gun movies database

Classifiers	Optimizers	F-score (%)	Recall (%)	Precision (%)	Accuracy (%)
Naïve Bayes	DOA	82.92	82.98	84.85	85.58
	GOA	85.78	86.60	88.67	88.40
	ACO	89.90	88.96	90.70	90.68
	PSO	90.86	88.80	92.26	92.58
	HIPSO	91.30	90.53	92.70	92.83
Decision tree	DOA	88.50	86.60	80.70	88.90
	GOA	89.78	88.90	87.57	89.20
	ACO	90.40	89.93	88.98	90.50
	PSO	90.48	90.78	90.70	91.87
	HIPSO	91.20	91.90	92.88	92.14
Random forest	DOA	87.78	84.90	89.98	87.30
	GOA	88.90	88.78	90.80	88.40
	ACO	90.06	90.40	91.50	90.69
	PSO	91.80	92.70	92.14	90.72
	HIPSO	92.20	94.03	92.50	93.78
SVM	DOA	90.40	90.78	90.90	89.50
	GOA	91.20	92.98	92.49	90.75
	ACO	92.78	94.87	94.26	93.29
	PSO	94.76	95.30	96.88	96.20
	HIPSO	96.45	96.20	98.56	98.60

Table 3. Comparison result of HIPSO-SVM model and the existing models

Models	Database	Recall (%)	Precision (%)
YOLO V3 with pose information [19]	YouTube	76.60	96.40
	Gun movies	44.10	98.40
Weapon appearance and human pose information [20]	YouTube	83.83	97.30
Canny edge detector with SVM [21]	Gun movies	81.80	-
HIPSO-SVM model	YouTube	96.27	98.58
	Gun movies	96.20	98.56

recall and 96.40% of precision on YouTube database, and 44.10% of recall and 98.40% of precision on Gun movies database. Further, Ruiz-Santaquiteria [20] integrated weapon appearance and human pose information for effective handgun detection. Experimental results confirmed that the presented model obtained 83.83% of recall and 97.3% of precision on YouTube database. Grega [21] integrated canny edge detector and MPEG-7 homogeneous texture descriptor for object detection and feature extraction. Next, the SVM classifier was applied for the weapon classification. The extensive experimental investigation showed that the presented model obtained 81.80% of recall value on Gun movies database.

The HIPSO-SVM model achieved better performance in weapon detection related to the comparative models. The HIPSO algorithm selects discriminative feature vectors that effectively decrease the computational complexity and training time of the SVM classifier, which are the major problems highlighted in the literature section. The computational complexity of the proposed model is linear, and the training time of the classifier is 33.56 and 48.22 seconds on the YouTube and Gun movies databases, which are limited compared to other machine learning classifiers.

5. Conclusion

In this manuscript, a new HIPSO-SVM model is introduced for effective weapon detection. The proposed HIPSO-SVM model comprises two important steps such as weapon detection and weapon and non-weapon classification. After detecting the weapon in the video sequences utilizing the GMM technique, the feature extraction is carried out using AlexNet, ResNet 18, and SIFT models for extracting feature vectors from the segmented images. The extracted multi-dimensional feature vectors are optimized by proposing a HIPSO algorithm that superiorly reduces the training time and computational complexity of the model. Lastly, the optimized feature vectors are given as the input to the SVM classification methodology for weapon and non-weapon classification. The conducted extensive experiment showed that the proposed HIPSO-SVM model achieved a higher classification accuracy of 95.34% and 98.60% on the YouTube and Gun movies databases, which are effective compared to other

Parameter	Notation
k	Number of Gaussians
$\Sigma_{i,t}$	Covariance matrix
$\mu_{i,t}$	Mean value of i^{th} Gaussian in the time instant t
X_t	Present pixel value
$\omega_{i,t}$	Weight of i^{th} Gaussian
G_j	Probable distribution
$diag\|x\|$	Diagonal matrix
α	Learning rate
T	Threshold value
I_w	Inertia weight
$c_1\ and\ c_2$	Acceleration coefficients
$r_1\ and\ r_2$	Two random numbers
p_{id}	Particles current best position
p_{gd}	Global best position
FP	False positive
FN	False negative
TP	True positive
TN	True negative

classifiers (naïve Bayes, decision tree and random forest) and optimizers (PSO, DOA, ACO, and GOA). As a future enhancement, a new ensemble classifier can be included in the proposed model to further improve weapon detection.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning", *IEEE Access*, Vol. 9, pp. 34366-34382, 2021.

[2] A. Egiazarov, V. Mavroeidis, F. M. Zennaro, and K. Vishi, "Firearm detection and segmentation using an ensemble of semantic neural networks", In: *Proc. of European Intelligence and Security Informatics Conference*, pp. 70-77, 2019.

[3] F. Gelana and A. Yadav, "Firearm detection from surveillance cameras using image processing and machine learning techniques", In: *Proc. of Smart Innovations in Communication and Computational Sciences, Springer*, pp. 25-34, 2019.

[4] M. Ghazal, N. Waisi, and N. Abdullah, "The detection of handguns from live-video in real-time based on deep learning", *Telkomnika*, Vol. 18, No. 6, pp. 3026-3032, 2020.

[5] N. Vallez, A. V. Mata, and O. Deniz, "Deep autoencoder for false positive reduction in handgun detection", *Neural Computing and Applications*, Vol. 33, No. 11, pp. 5885-5895, 2021.

[6] R. Olmos, S. Tabik, A. Lamas, F. P. Hernandez, and F. Herrera, "A binocular image fusion approach for minimizing false positives in handgun detection with deep learning", *Information Fusion*, Vol. 49, pp. 271-280, 2019.

[7] R. Debnath and M. K. Bhowmik, "A comprehensive survey on computer vision based concepts, methodologies, analysis and applications for automatic gun/knife detection", *Journal of Visual Communication and Image Representation*, p. 103165, 2021.

[8] J. Rose, T. Bourlai, and J. A. Loudermilk, "Assessment of Data Augmentation Techniques for Firearm Detection in Surveillance Videos", In: *Proc. of the IEEE International Conference on Big Data*, pp. 1838-1846, 2020.

[9] M. T. Ağdaş, M. Türkoğlu, and S. Gülseçen, "Deep neural networks based on transfer learning approaches to classification of gun and knife images", *Sakarya University Journal of Computer and Information Sciences*, Vol. 4, No. 1, pp. 131-141, 2021.

[10] J. Li, C. Ablan, R. Wu, S. Guan, and J. Yao, "Preprocessing Method Comparisons for VGG16 Fast-RCNN Pistol Detection", *EPiC Series in Computing*, Vol. 76, pp. 39-48, 2021.

[11] R. Olmos, S. Tabik, F. P. Hernandez, A. Lamas, and F. Herrera, "MULTICAST: MULTI Confirmation-level Alarm SysTem based on CNN and LSTM to mitigate false alarms for handgun detection in video-surveillance", *arXiv Preprint arXiv:2104.11653*, 2021.

[12] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning", *Neurocomputing*, Vol. 330, pp. 151-161, 2019.

[13] S. Narejo, B. Pandey, C. Rodriguez, and M. R. Anjum, "Weapon Detection Using YOLO V3 for Smart Surveillance System", *Mathematical Problems in Engineering*, 2021.

[14] V. Kaya, S. Tuncer, and A. Baran, "Detection and Classification of Different Weapon Types

Using Deep Learning", *Applied Sciences*, Vol. 11, No. 16, p. 7535, 2021.
[15] J. L. S. González, C. Zaccaro, J. A. Á. García, L. M. S. Morillo, and F. S. Caparrini, "Real-time gun detection in CCTV: An open problem", *Neural Networks*, Vol. 132, pp. 297-308, 2020.
[16] M. K. E. D. Mohamed, A. Taha, and H. H. Zayed, "Automatic gun detection approach for video surveillance", *International Journal of Sociotechnology and Knowledge Development*, Vol. 12, No. 1, pp. 49-66, 2020.
[17] J. Salido, V. Lomas, J. R. Santaquiteria, and O. Deniz, "Automatic handgun detection with deep learning in video surveillance images", *Applied Sciences*, Vol. 11, No. 13, p. 6085, 2021.
[18] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning", *Neurocomputing*, Vol. 275, pp. 66-72, 2018.
[19] A. V. Mata, J. R. Santaquiteria, N. Vallez, and O. Deniz, "Using human pose information for handgun detection", *Neural Computing and Applications*, Vol. 33, No. 24, pp. 17273-17286, 2021.
[20] J. R. Santaquiteria, A. V. Mata, N. Vallez, G. Bueno, J. A. Á. García, and O. Deniz, "Handgun detection using combined human pose and weapon appearance", *IEEE Access*, Vol. 9, pp. 123815-123826, 2021.
[21] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image", *Sensors*, Vol. 16, No. 1, p. 47, 2016.