[1]M. Srilatha

Dr.N.Srinivasu

Design an Effective, Faster Region-Based Convolutional Neural Network with Optimization for the Human Tracking System

Journal of Electrical Systems

Abstract: - Nowadays, real-time Human Tracking System (HTS) is a crucial topic in computer vision and image processing with applications like robotic perception, scene understanding, video surveillance, image compression, medical image analysis, and augmented reality, among many others. In this paper, we design a Faster Region-based Convolutional Neural Network (FR-CNN) with Crow Search Optimization (FR-CNN-CSO) architecture to improve computational complexity and enhance the performance of HTS. The system is implemented in a Python environment with video input. Remove unnecessary data from the gathered datasets during preprocessing. Next, feature extraction is processed using Histograms of Oriented Gradients (HOG). Then update, the extracted features into a designed FR-CNN model for identifying and tracking a person using crow search fitness. The main goal of the developed approach is to attain accurate prediction results and improve the computational complexity by achieving less execution time. Finally, the experimental outcomes show the reliability of the designed system by other conventional techniques in terms of accuracy, precision, recall, F-measure, and execution time.

Keywords: Human Tracking System, Crow Search Optimization, Faster Region Based Convolutional Neural Network, Surveillance System, and Artificial Intelligence.

1. INTRODUCTION

A complete framework for detecting a crude human model is called "human tracking," It is done using synchronized monocular grayscale image sequences in one or more camera system coordinates [1]. It merely involves separating an interested person from the video scene and constantly monitoring it. Monitoring has recently implemented visual-based tracking and detecting systems to increase human safety, convenience, and security [2]. An effective surveillance system must include subjects like human tracking and detection. Any specific human detection device usually consists of moving object extraction and human recognition [3, 4]. Moving object extraction is used to remove items from the background and calculate the relevant dimensions and location of the object in a video [5]. Human recognition classifies a picture as nonhuman or human. Although the tracked object or person may be obscured by other things while being followed, the tracking device must be able to forecast the position during and after occlusion [6]. Two types of cameras are frequently used in surveillance systems: active and fixed. A fixed camera has the advantage of being inexpensive, but it has a small field of view (FOV), but the active camera has a larger FOV since it can pan and tilt to keep the target item in the frame [7, 8]. Also, the latter offers a superior resolution because it features zoom-in/zoom-out capabilities. A monitoring system on an active camera often considers the temporal difference to extract moving objects. However, the camera must first be stable enough to process the image in this process [9, 10]. In other words, as an item moves, the camera separates background pixels from the pictures it captures [11]. The result is that the active camera acts in a jerky and irregular manner. An optimization technique is used to tackle this issue [12]. The human is first identified as the target model using the deep learning technique. The optimization technique then tracks the person by computing the distance between the target model's color histogram and the subsequent color histogram shape of the selected position [13, 14]. A color histogram has many benefits, including quick computing, partial occlusion resistance, non-rigid object tracking, scale invariance, and rotation [15].

[1] Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Vaddeswaram, AP, India.

[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Vaddeswaram, AP, India. Email: srinivasu28@kluniversity.in.

[1*]Corresponding Authors Email: srilatha.manam@gmail.com

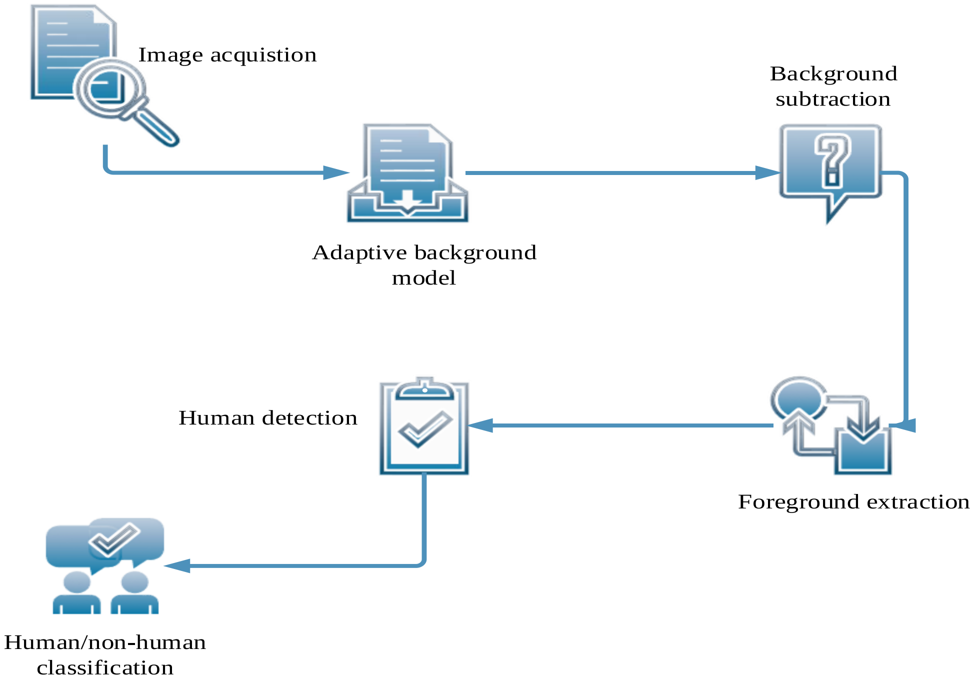

Fig.1 Human tracking system for video surveillance system

The surveillance system of the human tracking system is shown in fig.1. The widespread use of intelligent systems for processing information in contemporary society results from the explosive rise of artificial intelligence applications [16]. The Internet of Things and Artificial Intelligence has lately emerged as one of the hottest study areas; it combines artificial intelligence technology with IoT infrastructure to provide more efficient IoT services [17]. It has demonstrated excellent success for various surveillance applications in different outdoor and indoor complicated industrial contexts [18]. However, the efficacy of human detection techniques may be adversely affected by suddenly cluttered sceneries, changes in motion, camera angles, near interpersonal contact, and occlusion [19]. Furthermore, the individual's outward appearance varies substantially regarding the body's height, poses, scales, articulations, rotation, movements, sizes, and torso. Many applications include robot vision, surveillance, and tennis ball recognition in sports, traffic monitoring, animation, athletic performance analysis, and an interest in tracking motion [20, 21]. Several approaches to human tracking have previously been introduced. Yet it is discovered to be a vital work and does not produce 100% accurate outcomes to recognize the human and track it continually. Furthermore, the dramatic surge in crimes and accidents makes human safety a crucial issue [22, 23].

The most challenging task in human tracking systems is occlusion, less detection accuracy, computational complexity, problem detecting humans because of pose and viewpoint variation, classification, and regression issues [24]. Several deep learning models were developed to enhance the performance of the human tracking system, such as Convolutional Neural Networks (CNN), YOLOv2, and Region-based CNN (R-CNN). However, still suitable solution was not found because of data complexity, error rate, and fewer prediction results [25]. So, design a Faster Region-based Convolutional Neural Network (FR-CNN) with Crow Search Optimization (FR-CNN-CSO) architecture for enhancing the human tracking system by accurately detecting humans from the collected surveillance system. Also designed model accurately monitor and track human using CSO fitness which continuously monitors the tracking system and predict human. The main aim of the developed model is to improve the human tracking system using optimization techniques and attain better prediction results.

Main Objectives

- To design an effective human tracking system using crow search optimization and deep learning.
- To improve the prediction results of a group of videos using faster region-based CNN.

- To prove the efficiency of the designed model by attaining better feature extraction and classification results.

Additionally, the critical contribution of the proposed technique is detailed below,

- Initially, a video dataset was collected from the net source and trained in the system.
- Moreover, design an FR-CNN-CSO framework for accurately detecting and classifying a human tracking system.
- Then, input videos are updated to the preprocessing to remove the redundant information from the collected datasets.
- Hereafter, feature extraction is employed using Histograms of oriented gradients for extracting relevant features.
- Then update, the extracted features into the developed model FR-CNN, which detects the human with high performance using crow search fitness.
- Finally, update the CSO fitness in the classification layer for improving the prediction results, and the gained outcomes are validated with other prevailing models.

The manuscript's organization is as follows: section 2 and section 3 describe related works and problem statements. Section 4 elaborated on the proposed methodology, and section 5 discussed the results and discussion of the proposed technique with other papers. Finally, section 6 is ended with a conclusion and future scope.

2. RELATED WORKS

A DL-based system that combines two object detection techniques, faster mask-based and region-based convolutional neural networks (R-CNN), was proposed by Ming-Chuan et al. [26]. It is applied to collect the segmentation and used to look for the target's motion reference point. The object's location and real rotated angle are determined after integrating the inputs from the two approaches. Data indicate that the detection's accuracy was 96.26%. Increasing production capacity also shortens the processing time.

Imran Ahmed et al. [27] proposed an automated person detection system to track and identify individuals in a complicated industrial facility. Typically, a top view is selected due to its capacity to offer adequate coverage and visibility of a situation. This study provides evidence for the usefulness, efficacy, and feasibility of DL designs using transfer learning. With the highest True Positive Rate (TPR) of 93%, the efficiency of all detection designs has dramatically improved and shown promising results.

Depending on a drone's cameras, Kamel Boudjit et al. [28] developed programs for detecting and recognizing people using convolutional neural networks (CNN) YOLOv2. DL-based computer vision is used to identify the person's status and state. The outcomes of the person detection demonstrate the high degree of object detection and classification accuracy of YOLO-v2. In addition, the tracking algorithm reacts faster for real-time tracking than commonly used methods, effectively tracking the identified person without removing it from view.

Face mask identification technology has been published by Hiten Goyal et al. [29] for static photos and real-time videos, classifying the images as without and with a mask. The Kaggle data set is used to train and assess the model. The collected data set has a performance accuracy rating of 98% and roughly 4,000 images. The suggested model is accurate and efficient in terms of computing.

Hypermetropic CNN was created by Amudhan A.N. et al [30]. to detect small objects. More features are extracted from the shallow levels, and low-level features are transferred to the deeper layers, improving detection. Because it outperforms close items and lags on distant ones, the network is hypermetropic. The suggested approach exhibits remarkable gains in small-size item detection and a 32% rise in false favourable rates.

3. PROBLEM DEFINITION

The practical human tracking system enlarges the production capacity and minimizes processing time. The most challenging task in the human tracking system is improving computational efficiency [31]. In general, confined and controlled environments are superior for training AI-based systems. It can be applied in a different context with good accuracy after being trained in various situations with applicable data variances. Instead of learning the features of the environment, introduce the AI technique to acquire the parts of the target object. DL systems are data-hungry [32]. There are numerous problems with detecting human tracking systems, such as pose variation, viewpoint variation, and occlusion. The most common issues are classification and regression problems, vanishing gradient issues, error rate, computation time, and occlusion. These issues are motivated to enhance the performance of human tracking systems by improving computational efficiency.

4. PROPOSED METHODOLOGY

The investigators have proposed deep learning with a metaheuristic algorithm to enhance prediction accuracy and efficiency. This algorithm can provide better outcomes and effectively solve different image issues. Therefore, input videos are collected and updated for preprocessing in this paper.

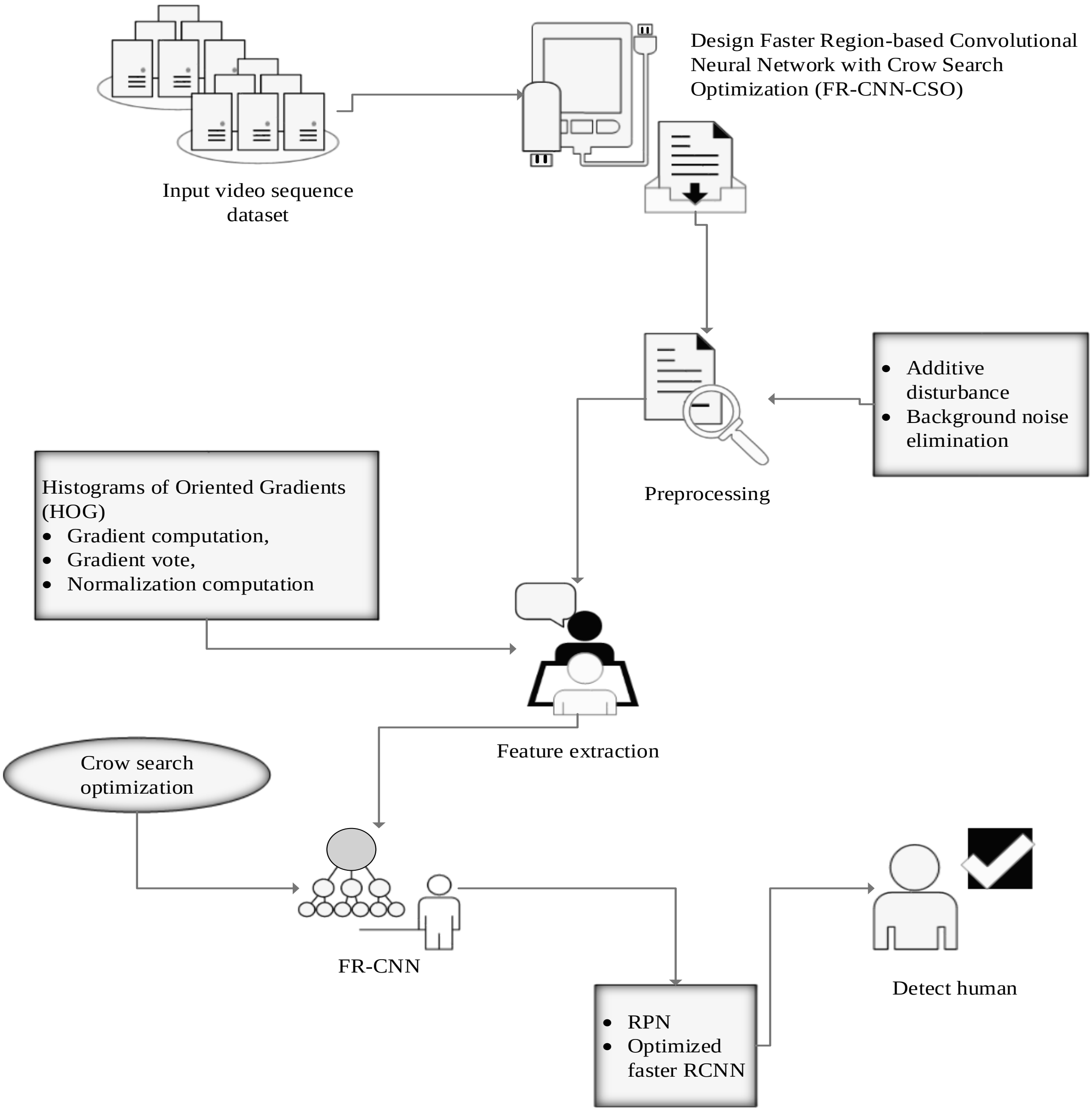
Input video sequence dataset
Design Faster Region-based Convolutional Neural Network with Crow Search Optimization (FR-CNN-CSO)
Additive disturbance
Background noise elimination
Preprocessing
Histograms of Oriented Gradients (HOG)
Gradient computation,
Gradient vote,
Normalization computation
Feature extraction
Crow search optimization
FR-CNN
RPN
Optimized faster RCNN
Detect human

Fig.2 Proposed methodology

The architecture of the proposed model is shown in fig.2. In the preprocessing stage, remove the redundant information from the collected datasets. Then extract the relevant features from the dataset using HOG for improving prediction results. Then design a novel, Faster Region-based Convolutional Neural Network with Crow Search Optimization (FR-CNN-CSO) framework for accurate identification and tracking of humans. After that, it updates Crow Search Optimization (CSO) [33] in the fully connected layer of the FR-CNN model for precise detection and to improve computational complexity. Finally, the simulation outcomes demonstrate the advantage of using FR-CNN-CSO for human tracking systems regarding classification accuracy.

4.1 Dataset description

The study makes use of CEPDOF, or Challenging Events for Person Detection from Overhead Fisheye Images (https://vip.bu.edu/projects/vsns/cossy/datasets/cepdof/download/). 8 videos at a maximum of 13 people visible at once, over 25,000 evaluated frames per second, several novel challenging situations in an extreme body occlusions, crowded room, various body positions, head camouflage images of people on a projection screen, and dim lighting with or without Infra-Red (IR) illumination are all included in the new dataset.

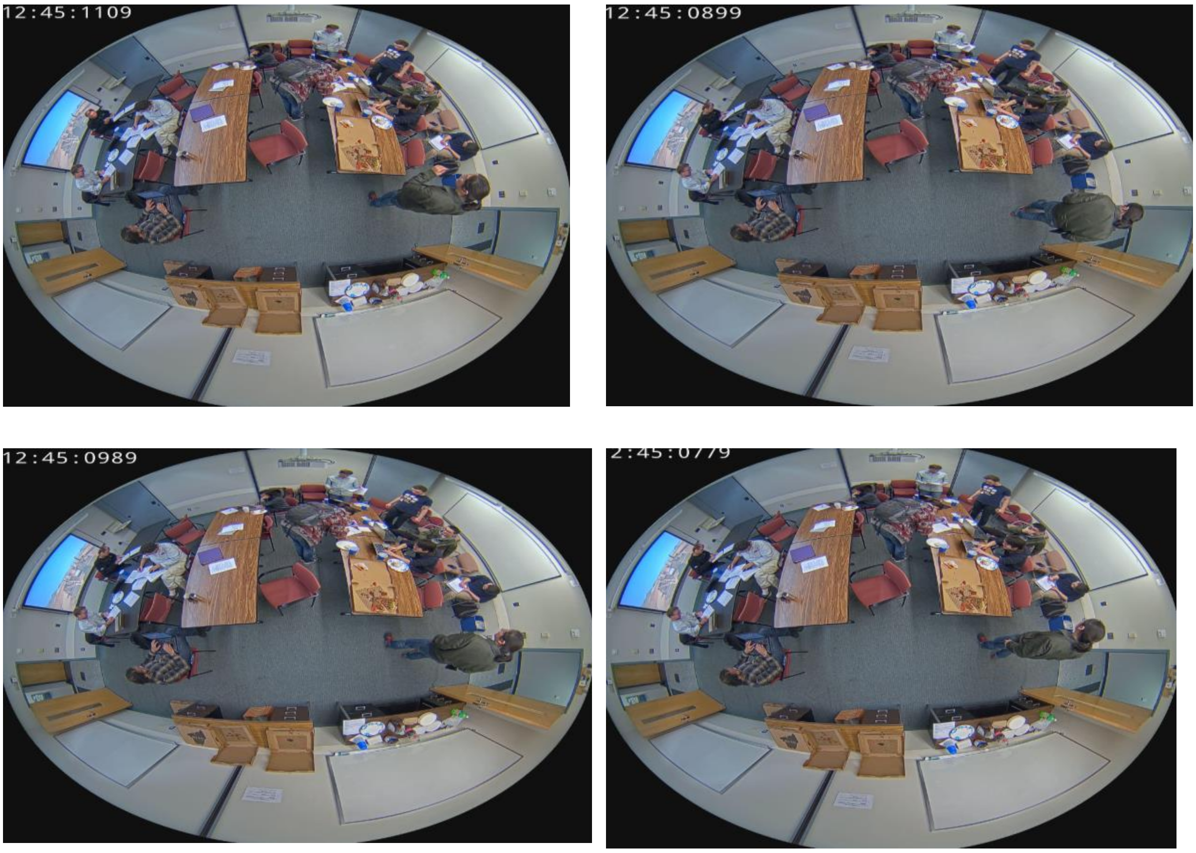

Fig.3 Sample images of the used dataset

Additionally, because CEPDOF is spatiotemporally marked, bounding boxes for the same individual in subsequent frames will carry the same ID. As a result, it can be utilised for further vision tasks like tracking moving objects in films and re-identifying people using fisheye overhead photos. Additionally, 80% of the datasets are utilised for training, while 20% are utilised for testing. The sample images of the dataset are shown in fig.3.

4.2 Preprocessing

Preprocessing is the first stage in creating feature vectors in an HTS and is used to distinguish between noisy and unnoticed datasets. The collected input datasets $I(n)$ are adjusted or modified during preprocessing to make them more suitable for feature extraction examination. Checking for errors and disturbances in the $I(n)$ data is the main thing consider regarding human detection processing. Because $I(n)$ is corrupted by some background noise $F(n)$. Equation (1) is used to determine the additive disturbance.

$$I(n) = v(n) + F(n) \tag{1}$$

Let, $v(n)$ is denoted as a cleaned dataset. Various noise reduction techniques are used to carry out the task on noisy data. Nevertheless, adaptive noise cancellation and spectral subtraction are the two noise reduction techniques widely utilized in human identification systems to develop HTS.

Adjusting the levels of noise in HTS requires consideration of the surrounding noise. When both training and test data are conducted with varying noise levels, the performance metric for human identification systems suffers severely. The noises in real-world situations are typically outside the control of HTS's creators. Preprocessing is a common technique used throughout the training and testing phases to lessen the impact of background noise on human recognition. The filter used in Eqn. (2) is as follows to eliminate background noise.

$$B_{no} = 10 * \log_{10}\left[\theta + \frac{1}{M}\sum_{n=1}^{M} v^2(n)\right] \tag{2}$$

Let, B_{no} is considered as the removal of background noise using M samples, θ is denoted as a small positive constant.

4.3 Feature Extraction

The pre-processed dataset is updated to the feature extraction phase, which extracts the relevant features from the dataset using Histograms of Oriented Gradients (HOG) [34]. It's employed to take features out of image or video data. It is commonly used for object detection in tasks involving computer vision. Moreover, HOG feature extraction is classified into three steps gradient computation, gradient vote, and normalization computation.

The gradient computation is extracted based on the coordinate pixel (a,b), direction $\mu(a,b)$, and magnitude $m(a,b)$ is calculated in this phase. The gradient and luminance values of a and b are measured using Eqn. (3) and (4).

$$f_a(a,b) = f(a+1,b) - f(a-1,b) \tag{3}$$

$$f_b(a,b) = f(a,b+1) - f(a,b-1) \tag{4}$$

Let, $f_a(a,b)$ and $f_b(a,b)$ are considered as the gradient of a and b axes. Moreover, $f(a,b)$ is denoted as a luminance value.

The magnitude $m(a,b)$ is measured using Eqn. (5)

$$m(a,b) = \sqrt{f_a(a,b)^2 + f_y(a,b)^2} \tag{5}$$

The direction $\mu(a,b)$ is computed using Eqn. (6)

$$\mu(a,b) = \arctan\frac{f_b(a,b)}{f_a(a,b)} \tag{6}$$

Every pixel inside the cell generates a gradient vote for an oriented histogram by the orientations of the gradient element centered on it after acquiring the magnitude $m(a,b)$ and direction $\mu(a,b)$. Nine bins are evenly spread out across the orientation's range of 0 to 180. Each pixel's weight, represented by the symbol, can be calculated using Eqn (7).

$$\tau = (j + 0.5) - \frac{c * \mu(a,b)}{\pi} \tag{7}$$

Let, j is represented as the bin to which $\mu(a,b)$ belongs and c is denoted as the total number of bins. A histogram normalization computation is finally produced by collecting all histograms from one block, which has four cells. The result after normalization can be written as Eqn (8).

$$g_i^2 = \frac{g_i}{\sqrt{\|g\|_2^2 + \lambda^2}} \tag{8}$$

Let, i is denoted as several cells and bins, g_i is considered a vector corresponding by combining histogram for block region, and λ is represented as a small constant.

4.4 Human detection and tracking using FR-CNN-CSO

A more traditional deep learning technique called Faster R-CNN (FR-CNN) [35] offers a high detection performance, efficiency, and better recognition rate for a significant target region. The Faster R-CNN technique primarily consists of two components: the extraction module and the detection module, the Fast R-CNN. Region Proposal Network (RPN) acts as an extraction module, and the FR-CNN model is a detection module. The RPN is utilized from the baseline feature map to produce high-quality regions, and the Fast R-CNN immediately recognizes and classifies the targets in the derived suggestions region. The VGG-16 network is used to store photos of any size. Second, the CNN network produced the shared convolutional layer and feature map. After being inserted into the RPN network, the feature map spread to a particular convolutional layer and created a higher-dimensional feature map. Lastly, the characteristics of the recommendation region were retrieved from the higher dimensional feature map using the RoI Pooling. The architecture of the optimized FR-CNN model is shown in fig.4.

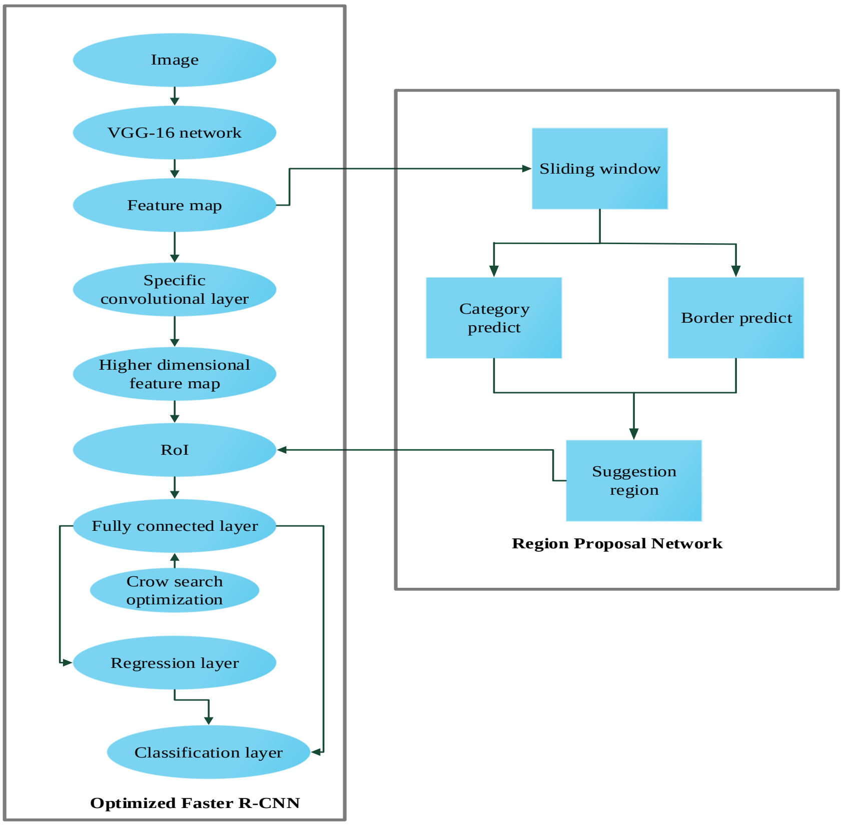

Fig.4 Architecture of optimized FR-CNN model

Following that, the features were added to the classification and regression layers. The prediction outcomes of HTS were improved using CSO optimization. The algorithm then provided the target object category and the Region's coordinates. The Faster R-CNN algorithm has shown promising results in target detection and recognition, and deep learning performance has significantly increased.

Crow search optimization: A new population-based algorithm, the Crow Search Algorithm (CSA), mimics how crows hide food. Crows are intelligent birds that can recognize faces and alert their species to danger. One of the ways they show their cunning is by hiding food and remembering where it is. Each crow is assessed using a fitness function, and its value is stored as the initial memory value. Each crow records where it hides in its memory parameter and updates its location by picking a random crow. This fitness function is employed to locate and follow human participants in the dataset based on their behavior during random selection. That raises the complexity of computing and boosts the effectiveness of the human tracking system.

First, improve computational complexity by implementing Faster R-CNN for person detection. The output includes the rectangle bounding box's (height, coordinates, and width) confidence score value and class label. The information regarding the object's position on the video is contained in this boundary box. The first step produces region anchors using RPN, a two-stage detector. The following stage is used for human classification, which collects bounding box data using discovered anchor regions. Via convolution layers, the attributes for the input video are retrieved. The generated feature maps are created further using the retrieved features. The generation of anchor or region boxes takes place using the sliding window method. These anchor boxes are further refined to identify an object's or person's appearance in the video. The anchors/detected bounding boxes are improved in the final stage, which also involves applying a CSO fitness function in a fully connected layer to improve human prediction using a small network and computing the loss function to determine the best anchor regions.

The RPN generates a series of rectangle object proposals, each with an object score, from an input video (size). Moreover, extract convolutional features from the original photos using the network. The convolutional feature maps are entered into a tiny network that maps them to a lower-dimensional set of features using a $n \times n$ spatial window as input. A box classification and regression layer, which determines the box as a collection of classification tasks or background, is fed this feature. Set reference boxes (anchors) with a size and aspect ratio at every sliding-window location to increase detection precision. The Faster R-CNN anchors share three scales and three aspect ratios that yield $s = 9$ at every sliding position. The RPN function loss is measured using Eqn. (9).

$$A(\{Y_i\},\{Q_i\}) = \frac{1}{n_{cl}} \sum_i A_{cl}\left(Y_i, Y_i^*\right) + \eta \frac{1}{n_{rg}} \sum_i Y_i^* A_{rg}\left(Q_i, Q_i^*\right) + C_s(t) \quad (9)$$

Where, i is considered as the anchor index, Y_i is denoted as anchor predicted probability belonging to i object, Q_i is represented as a vector representation of predicted boundary box, and Q_i^* is called ground truth. Moreover, n_{cl} is denoted as minibatch size, n_{rg} is considered as several anchor locations, and $C_s(t)$ is represented as fitness of the CSO.

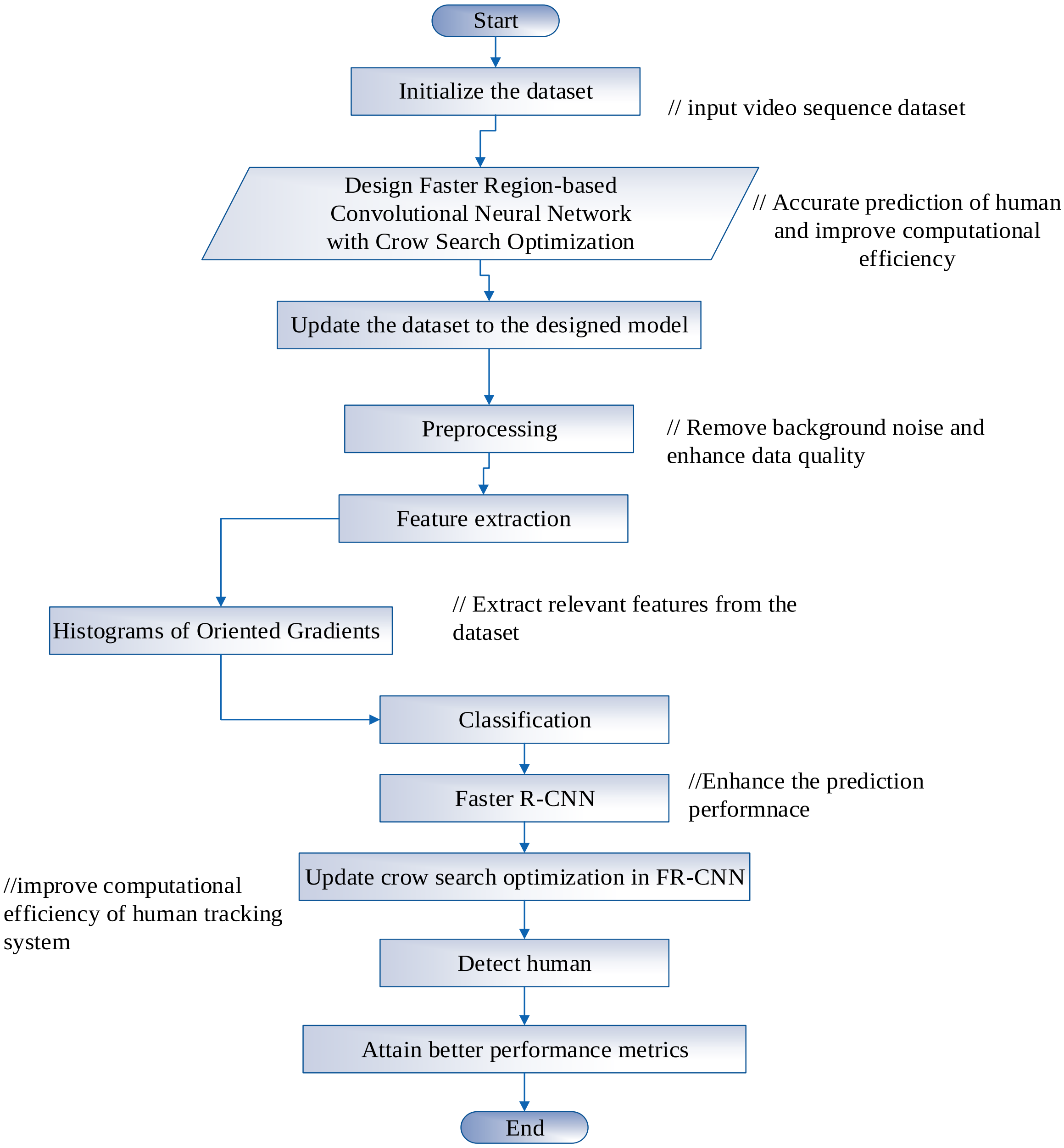
Start
Initialize the dataset
// input video sequence dataset
Design Faster Region-based Convolutional Neural Network with Crow Search Optimization
// Accurate prediction of human and improve computational efficiency
Update the dataset to the designed model
Preprocessing
// Remove background noise and enhance data quality
Feature extraction
Histograms of Oriented Gradients
// Extract relevant features from the dataset
Classification
Faster R-CNN
//Enhance the prediction performnace
Update crow search optimization in FR-CNN
//improve computational efficiency of human tracking system
Detect human
Attain better performance metrics
End

Fig.5 Flow chart of developed FR-CNN-CSO model

The coordinates of the bounding box regression parameterizations is described in Eqn. (10).

$$\begin{aligned} Q_x &= (X - X_b)/D_b, Q_y = (Y - Y_b)/E_b, \\ Q_d &= \log\left(D/D_b\right), Q_e = \log\left(E/E_b\right), \\ Q_x^* &= (X^* - X_b)/D_b, Q_y^* = (Y^* - Y_b)/E_b, \\ Q_d^* &= \log\left(D^*/D_b\right), Q_e^* = \log\left(E^*/E_b\right), \end{aligned} \tag{10}$$

Let, X, and Y are considered as box center coordinates, and E are denoted as the box centre's width and height. Moreover, X is denoted as a predicted box, X_b is called anchor box, and X^* is considered a ground truth box. A human tracking system's objective is to locate and categorize each instance of a human with a bounding box in the input image or video frames. Regression and classification both use the loss function. Finally, the identified bounding box with people and a total class score are produced at the output. The flow chart of the developed model is shown in fig.5.

5. RESULTS AND DISCUSSIONS

The FR-CNN-CSO methodology uses Python software to locate and follow people. The Faster R-CNN updates the CSO fitness to raise computational complexity and boost the results' precision. Also, during the preparation stage, noise from the dataset is eliminated. Moreover, feature extraction removes unneeded features and extracts essential information from the dataset. Ultimately, a built model can quickly and accurately classify and identify people.

5.1 Performance analysis

The obtained findings are tested with other widely used conventional techniques to demonstrate the efficiency of the developed method. Also, the F-score, precision, accuracy, recall, and execution time, performance measures are contrasted. Thus the existing techniques are YOLOv3 [27], YOLO-v2 [28], and CNN [29].

5.1.1 Accuracy

Closed assessments of the acceptable or real value are used to assess accuracy. The actual or desired value of something is determined using accuracy. Eqn. (11) is used to calculate accuracy.

$$A_c = \frac{t^*_{po} + t^*_{ne}}{t^*_{po} + f^*_{po} + t^*_{ne} + f^*_{ne}} \tag{11}$$

Let, t^*_{po} denoted as the actual positive rate of a detected human, t^*_{ne} is considered the correct negative rate of a detected human. Moreover, f^*_{po} and f^*_{ne} are represented as the incorrect favourable and incorrect negative rates of detected humans. The comparison of the accuracy is exposed in fig.6.

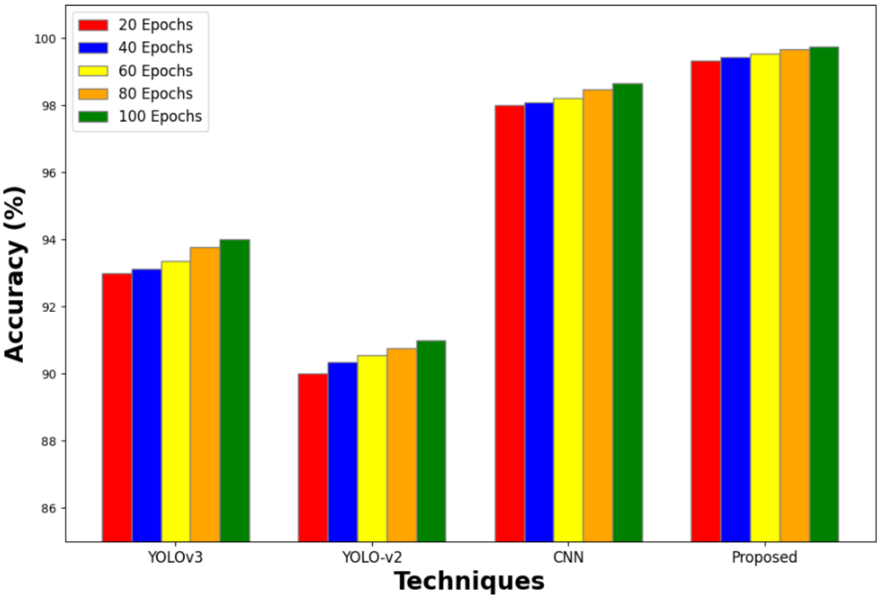

Fig.6 Comparison of accuracy

The proposed model increased accuracy results are compared to popular models like YOLOv3, YOLO-v2, and CNN. Moreover, the accuracy rates for YOLOv3 and YOLO-v2 were 93% and 90%, respectively, for 20 epochs. Similar results were obtained by the CNN model (98% accuracy). Eventually, the created model gained an accuracy of 99.34% for 20 epochs. The accuracy rates for YOLOv3 and YOLO-v2 were 94% and 90.98%, respectively, for 100 epochs. Similar results were obtained by the CNN model (98.66% accuracy). Eventually, the created model gained an accuracy of 99.75% for 1000 epochs. The designed technique achieves higher accuracy ratings when comparing other models.

5.1.2 Precision

Moreover, the closed metrics of the human detecting system are used to determine precision. Precision is distinct from accuracy and is more trustworthy when evaluated by consistent results. Precision is calculated using Eqn. (12).

$$P_r = \frac{t^*_{po}}{t^*_{po} + f^*_{po}} \tag{12}$$

The results of the developed model's increased precision are validated using findings from other popular models like YOLOv3, YOLO-v2, and CNN. The precision rates for YOLOv3 and YOLO-v2 gained 90% and 89.41%, respectively, for 20 epochs. In accordance, the CNN technique achieves 98% precision for 20 epochs. Eventually, the developed model achieves a precision of 99.45% for 20 epochs. Moreover, precision rates for YOLOv3 and YOLO-v2 gained 91.76% and 90.76%, respectively, for 100 epochs. In accordance, the CNN technique achieves 98.47% precision for 100 epochs. Eventually, the developed model achieves a precision of 99.79% for 100 epochs. Compared to other models, the designed one achieves higher precision scores. Fig.7 displays a graphical representation of precision.

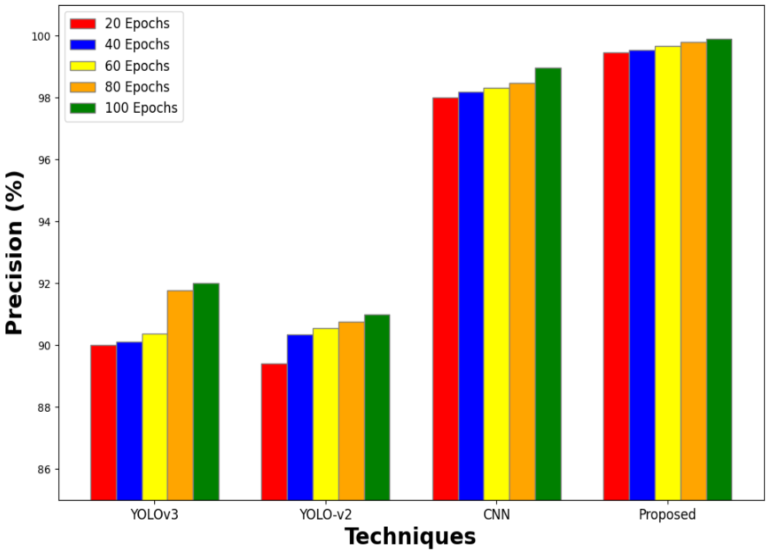

Fig.7 Comparison of precision

5.1.3 Recall

The recall is a total of all relevant data that have been located and are helpful in some way. The ability of the created model to find each appropriate case in the dataset is called recall. Moreover, it is calculated by dividing the total number of true positives by the sum of false negatives and true positives. The recall is calculated using Eqn. (13).

$$R_e = \frac{t^*_{po}}{t^*_{po} + f^*_{ne}} \tag{13}$$

The acquired recall results of the developed model are validated using various widely used models, including YOLOv3, YOLO-v2, and CNN. Moreover, the recall rates for YOLOv3 and YOLO-v2 were 88% and 83.5%, respectively, for 20 epochs. Similar results were obtained by the CNN model (97%) for 20 epochs. Eventually, 99.15% of recall is achieved by the designed model for 20 epochs. In addition, the recall rates for YOLOv3 and YOLO-v2 were gained 89% and 86%, respectively, for 100 epochs. Similar results were obtained by the CNN model (86%) for 100 epochs. Eventually, 99.69% of recall is achieved by the designed model for 100 epochs. Compared to other models, the designed one achieves higher recall scores. Figure 8 displays a graphic illustration of recall.

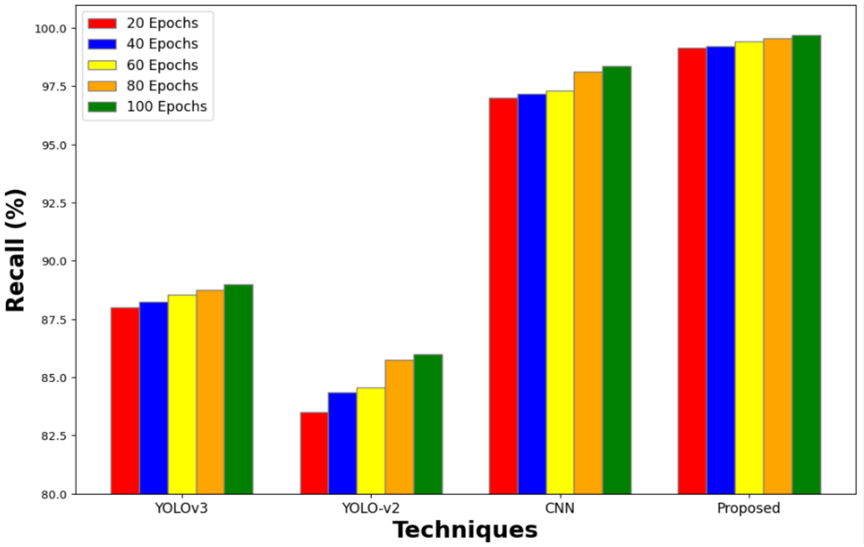

Fig.8 Comparison of recall

5.1.4 F-score

The F-score is created by adding all of the categorization measures together. The F-score perceives the precision and recall of the scoring method. The higher F-score reflects the accuracy of the categorization measures as predictors. Eqn. (14) is used to determine the F-score.

$$F_s = 2 \times \frac{P_r \times R_e}{P_r + R_e} \tag{14}$$

Where, P_r is denoted as precision and R_e is represented as recall.

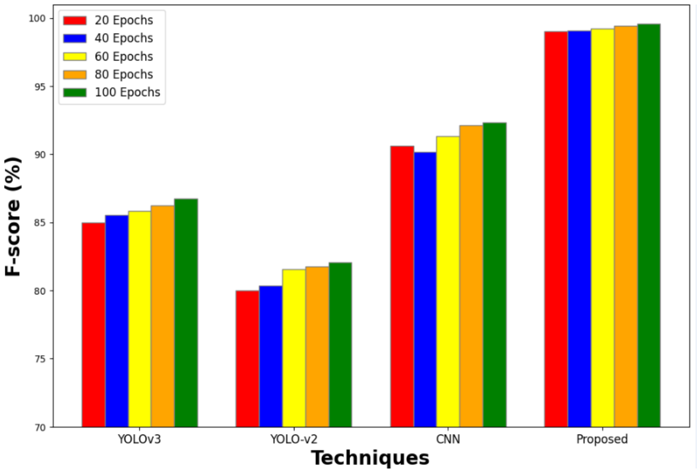

Fig.9 Comparison of F-score

The developed model's F-score results are validated against other widely used models, including YOLOv3, YOLO-v2, and CNN. The F-score rates for YOLOv3 and YOLO-v2 were gained 85% and 80%, respectively, for 20 epochs. In accordance, the CNN technique achieves a 90.6% F-score for 20 epochs. Eventually, 99% of the F-score is achieved by the designed technique for 20 epochs. Additionally, the F-score rates for YOLOv3 and YOLO-v2 were gained 86.76% and 82.05%, respectively, for 100 epochs. In accordance, the CNN technique achieves a 92.35% F-score for 100 epochs. Eventually, 99.57% of the F-score is achieved by the designed technique for 100 epochs. Compared to other models, the designed one achieves higher F-scores. In addition, the designed one performed higher F-scores while comparing other models. Figure 9 shows a graphic representation of the F-score.

5.1.5 Execution time

The execution time, typically independent of the start time but frequently depends on the input data, determines how long a process takes. The period between the start and end times is the execution time. To find the execution time, subtract the starting time from the finishing time. Time complexity is the exponential behavior of execution times as input size approaches infinity. Computational complexity measures how long an algorithm executes with the input's length. It computes the execution time of each algorithm's program statement. A comparison of the execution times is shown in Figure 10.

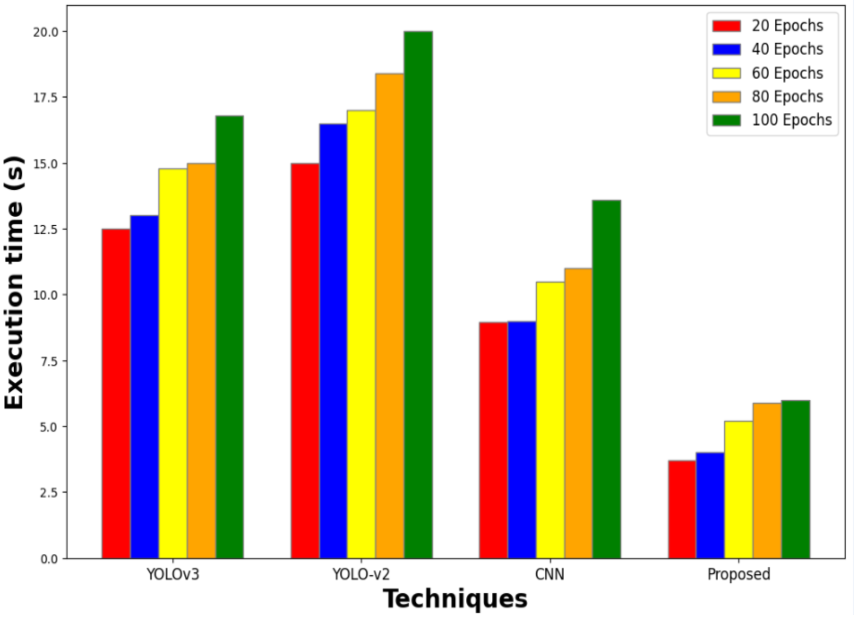

Fig.10 Comparison of execution time

The results of the proposed model's acquired execution time are verified against those of other widely used models like YOLOv3, YOLO-v2, and CNN. The execution time for YOLOv3 and YOLO-v2 was 12.5s and 15s, respectively, for 20 epochs. The CNN technique gained 8.95s execution time for 20 epochs. Eventually, the developed model achieves a 3.7s execution time for 20 epochs. Additionally, the execution time for YOLOv3 and YOLO-v2 gained at 16.8s and 20s for 100 epochs. The CNN technique gained 13.6s execution time for 100 epochs. Eventually, the developed model achieves a 6s execution time for 100 epochs. The planned technique has attained lower execution time while comparing other methods. Moreover, it shows the improved computational complexity of the designed model.

5.2 Discussions

F-score, precision, accuracy, recall, and execution time performance outcomes were all improved by the new technique. Also, the time required to locate and follow the human presence in the collected dataset is reduced compared to other models. However, the developed model increases computational complexity by taking less time to run. The proposed model's performance is eventually validated using various data sizes. Details regarding the results of the model are provided in Table 2.

Table.2 Overall performance

Data sizes (kb)	Performance assessments				
	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	Execution time (s)
100	99.34	99.45	99.15	99	3.7
200	99.44	99.53	99.22	99.09	4
300	99.53	99.66	99.42	99.23	5.2
400	99.68	99.79	99.55	99.44	5.9
500	99.75	99.89	99.69	99.57	6

Also, the FR-CNN-CSO model, which correctly and efficiently tracks the human, has upgraded crow search fitness. Furthermore, 30% of datasets are utilized for testing, while 70% are used for training. As a result, the designed model attained better results in identifying and tracking humans, reaching 99.34%, 99.44%, 99.53%, 99.68%, and 99.75% accuracy, 99.45%, 99.53%, 99.66%, 99.79%, and 99.89% precision, 99.15%, 99.22%, 99.42%, 99.55%, and 99.69% recall, 99%, 99.09%, 99.23%, 99.44%, and 99.57% F-score, and 3.7s, 4s, 5.2s, 5.9s, and 6s execution time for 100, 200, 300, 400, and 500 kb data sizes. The gained performance of accuracy vs. loss is shown in fig.11.

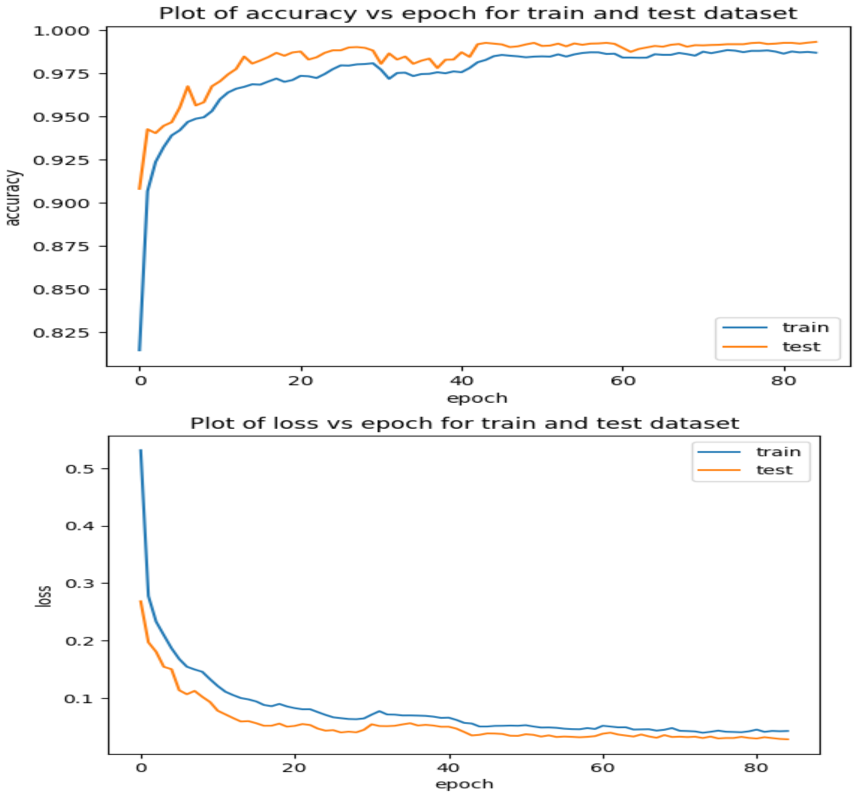

Fig.11 Accuracy Vs. Loss

As a result, the developed model produces improved results for predicting human tracking systems. In addition, the created model improves the computational complexity using an optimized deep learning model and boosts the human tracking system.

6. CONCLUSION

In this work, an approach of Faster Region-based CNN with CSO is developed to improve computation efficiency in human tracking systems. Optimizing-based deep learning with one-class detection is a challenging task in a real-time system where computational time is a considerable parameter and affects the system's efficiency. The developed approach has the advantage of high accuracy by reducing computational time. The proposed method comprises preprocessing, feature extraction, and human detection. The efficiency achieved for human detection is 99.34%. The recall reached for the human target tracking based on several experiments is 99.15%. The results have proved that using an FR-CNN-CSO model can achieve more accurate results within a reasonable computational time. Implementing such kind of strategy in a complex system like a drone is very feasible. In the future, hybrid optimization with enhanced DL models will improve the performance of the human tracking system and enhance computational complexity.

Compliance with Ethical Standards

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent

Informed consent does not apply as this was a retrospective review with no identifying patient information.

Funding: Not applicable

Conflicts of interest Statement: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material:

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Code availability: Not applicable

REFERENCES

1. Rohei, Muhammad Sadiq, et al. "Design and testing of an epidermal RFID mechanism in a smart indoor human tracking system." IEEE Sensors Journal 21.4 (2020): 5476-5486.
2. Nguyen, Van-Truong, et al. "A real-time human tracking system using convolutional neural network and particle filter." Intelligent Systems and Networks: Selected Articles from ICISN 2021, Vietnam. Springer Singapore, 2021.
3. Xue, Kunxi, et al. "Robotic seam tracking system based on vision sensing and human-machine interaction for multi-pass MAG welding." Journal of Manufacturing Processes 63 (2021): 48-59.
4. Motroni, Andrea, et al. "An RFID tracking system for Agricultural Safety." 2021 IEEE International Conference on RFID Technology and Applications (RFID-TA). IEEE, 2021.
5. Cetinkaya, Osman Tarik, et al. "A Fuzzy Rule Based Visual Human Tracking System for Drones." 2019 4th International Conference on Computer Science and Engineering (UBMK). IEEE, 2019.
6. Xu, Jianpei, et al. "Vitality-Enhanced Dual-Modal Tracking System Reveals the Dynamic Fate of Mesenchymal Stem Cells for Stroke Therapy." Small 18.47 (2022): 2203431.
7. Revilla-León, Marta, Jonathan M. Zeitler, and John C. Kois. "Digital maxillomandibular relationship and mandibular motion recording by using an optical jaw tracking system to acquire a dynamic virtual patient." The Journal of Prosthetic Dentistry (2022).
8. Tian, Li-Ping, et al. "Wits: An efficient Wi-Fi based indoor positioning and tracking system." Remote Sensing 14.1 (2022
9. Azimjonov, Jahongir, and Ahmet Özmen. "A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways." Advanced Engineering Informatics 50 (2021): 101393.
10. Antonanzas, J., M. Arbeloa-Ibero, and J. C. Quinn. "Comparative life cycle assessment of fixed and single axis tracking systems for photovoltaics." Journal of Cleaner Production 240 (2019): 118016.

11. Hempel, Thorsten, and Ayoub Al-Hamadi. "Slam-based multistate tracking system for mobile human-robot interaction." Image Analysis and Recognition: 17th International Conference, ICIAR 2020, Póvoa de Varzim, Portugal, June 24–26, 2020, Proceedings, Part I 17. Springer International Publishing, 2020.
12. Zou, Yanbiao, et al. "Robotic seam tracking system combining convolution filter and deep reinforcement learning." Mechanical Systems and Signal Processing 165 (2022): 108372.
13. Liu, Meng, Youfu Li, and Hai Liu. "3D gaze estimation for head-mounted eye tracking system with auto-calibration method." IEEE Access 8 (2020): 104207-104215.
14. Younis, Ola, et al. "A hazard detection and tracking system for people with peripheral vision loss using smart glasses and augmented reality." International Journal of Advanced Computer Science and Applications 10.2 (2019): 1-9.
15. Yang, Fei, Rong Zhang, and Youpeng Zhao. "Research on a visual sensing and tracking system for distance education." International Journal of Emerging Technologies in Learning (Online) 14.8 (2019): 181.
16. Xie, Jing, Erik Stensrud, and Torbjørn Skramstad. "Detection-based object tracking applied to remote ship inspection." Sensors 21.3 (2021): 761.
17. Biroju RaviKiran, P., et al. "Implementation and Optimization of Human Tracking System using Beagle Boneblack Embedded Platform."
18. Liu, Ying, et al. "Improvement of robot accuracy with an optical tracking system." Sensors 20.21 (2020): 6341.
19. Zhao, Peijun, et al. "Human tracking and identification through a millimeter wave radar." Ad Hoc Networks 116 (2021): 102475.
20. Larumbe-Bergera, Andoni, et al. "Accurate pupil center detection in off-the-shelf eye tracking systems using convolutional neural networks." Sensors 21.20 (2021): 6847.
21. Pandiyan, P., et al. "Real-time monitoring of social distancing with person marking and tracking system using YOLO V3 model." International Journal of Sensor Networks 38.3 (2022): 154-165.
22. Hao, Qian, Zhifang Wang, and Lele Qin. "Design of BeiDou satellite system in ocean logistics real-time tracking system." Journal of Coastal Research 94.SI (2019): 204-207.
23. Liu, Chang, and Tamás Szirányi. "Real-time human detection and gesture recognition for on-board UAV rescue." Sensors 21.6 (2021): 2180.
24. Shu, Francy, and Jeff Shu. "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box." Scientific reports 11.1 (2021): 2471.
25. Zhang, Kevin, et al. "Double anchor R-CNN for human detection in a crowd." arXiv preprint arXiv:1909.09998 (2019).
26. Chiu, Ming-Chuan, Ho-Yen Tsai, and Jing-Er Chiu. "A novel directional object detection method for piled objects using a hybrid region-based convolutional neural network." Advanced Engineering Informatics 51 (2022): 101448.
27. Ahmed, Imran, Marco Anisetti, and Gwanggil Jeon. "An IoT-based human detection system for complex industrial environment with deep learning architectures and transfer learning." International Journal of Intelligent Systems 37.12 (2022): 10249-10267.
28. Boudjit, Kamel, and Naeem Ramzan. "Human detection based on deep learning YOLO-v2 for real-time UAV applications." Journal of Experimental & Theoretical Artificial Intelligence 34.3 (2022): 527-544.
29. Goyal, Hiten, et al. "A real time face mask detection system using convolutional neural network." Multimedia Tools and Applications 81.11 (2022): 14999-15015.
30. Amudhan, A. N., and A. P. Sudheer. "Lightweight and computationally faster Hypermetropic Convolutional Neural Network for small size object detection." Image and Vision Computing 119 (2022): 104396.
31. Yamazaki, Yuki, et al. "Victim detection using UAV with on-board voice recognition system." 2019 Third IEEE International Conference on Robotic Computing (IRC). IEEE, 2019.
32. Chander, Harish, et al. "Wearable stretch sensors for human movement monitoring and fall detection in ergonomics." International journal of environmental research and public health 17.10 (2020): 3554.
33. Hussien, Abdelazim G., et al. "Crow search algorithm: theory, recent advances, and applications." IEEE Access 8 (2020): 173548-173565.
34. Matsumura, Ryo, and Akitoshi Hanazawa. "Human detection using color contrast-based histograms of oriented gradients." International Journal of Innovative Computing, Information and Control 15.4 (2019
35. Singh, Sunil, et al. "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment." Multimedia Tools and Applications 80 (2021): 19753-19768.