Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.037738
Article

Automated Video-Based Face Detection Using Harris Hawks Optimization with Deep Learning

Latifah Almuqren[1], Manar Ahmed Hamza[2,*], Abdullah Mohamed[3] and Amgad Atta Abdelmageed[2]

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[3]Research Centre, Future University in Egypt, New Cairo, 11845, Egypt

*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa
Received: 15 November 2022; Accepted: 23 February 2023

Abstract: Face recognition technology automatically identifies an individual from image or video sources. The detection process can be done by attaining facial characteristics from the image of a subject face. Recent developments in deep learning (DL) and computer vision (CV) techniques enable the design of automated face recognition and tracking methods. This study presents a novel Harris Hawks Optimization with deep learning-empowered automated face detection and tracking (HHODL-AFDT) method. The proposed HHODL-AFDT model involves a Faster region based convolution neural network (RCNN)-based face detection model and HHO-based hyperparameter optimization process. The presented optimal Faster RCNN model precisely recognizes the face and is passed into the face-tracking model using a regression network (REGN). The face tracking using the REGN model uses the features from neighboring frames and foresees the location of the target face in succeeding frames. The application of the HHO algorithm for optimal hyperparameter selection shows the novelty of the work. The experimental validation of the presented HHODL-AFDT algorithm is conducted using two datasets and the experiment outcomes highlighted the superior performance of the HHODL-AFDT model over current methodologies with maximum accuracy of 90.60% and 88.08% under PICS and VTB datasets, respectively.

Keywords: Face detection; face tracking; deep learning; computer vision; video surveillance; parameter tuning

1 Introduction

Face recognition is considered a hot research topic in computer vision (CV) and human-computer interaction (HCI) [1]. It is the fundamental step for a system which deals with face analysis. Several studies had completed pointing to automatic face detection [2]. Face detection exists in everyone's lives, but many technology users may not think in depth. If not using a digital camera, the social media

app Snapchat, the phone camera, tagging feature of Facebook, chances are you have experienced face detection earlier [3]. Face detection can be the implementation of computer technology for detecting faces in digital images. The latest research in CV mainly aims at face detection in uncontrolled environments due to changes in face appearances (e.g., pose changes and illuminations) that can result in the worst sturdiness of the system. Recently, convolutional neural networks (CNNs) have become the most typically utilized techniques for representing features and reached good outcomes in face detection issues [4]. Face detection is of 2 categories; one is face verification, in which 2 faces are presented, and the system should be verified if these two faces belong to the same individual, and another one is face identification, in which a face image can be represented with an unknown identity and the system should determine this identity [5].

The CNN needs an additional computational period for computing complicated features for improvising accuracy [6]. But the added computational load is normalized by reducing several cascade stages. The decrease in cascade stages minimizes the computational load without affecting the performance. A decline in cascade phases will make the entire computation stay unmodified. This monitoring can be encouraged with the help of advanced features-related techniques for face detectors. The CNN-related techniques contrasted with the hand-engineered feature-related approaches [7]. It deals with hard visual differences by using a large trained dataset. CNN functions similarly to the typical artificial neural network (ANN). Still, the neuron in a CNN layer was associated with a particular subarea of the former layers. Conversely, every neuron was completely linked in ANN [2]. The neuron in a CNN layer has been organized in 3 dimensions: height, width, and depth. Many interesting researchers concentrating on face detection have attained higher success [8]. But whether the pose was altered or the face was presented at an angle, the individual could not identify. Earlier techniques for face detection were dependent upon the discriminative classifying method were well-trained on a dataset of familiar identities, and an intermediate bottleneck layer was utilized as a depiction for detection. This technique specifies a huge representation for every face, but only some studies attempted to reduce dimensionality using principal component analysis (PCA) [9].

Though several models are available in the literature, most of the works have not focused on hyperparameter tuning process. Therefore, this study presents a novel Harris Hawks Optimization with deep learning-empowered automated face detection and tracking (HHODL-AFDT) model. The proposed HHODL-AFDT model involves Faster region-based convolutional neural network (RCNN) based face detection model and HHO based hyperparameter optimization process. The presented optimal Faster RCNN model precisely recognizes the face and is passed into the face-tracking model using a regression network (REGN). The face tracking using the REGN model uses the features from neighboring frames and foresees the location of the target face in succeeding frames. The experimental validation of the presented HHODL-AFDT algorithm is conducted using two datasets, and the outcomes are inspected using various aspects.

2 Related Works

Wang et al. [10] developed a methodology for face detection in real-time surveillance video using the deep learning (DL) technique. Firstly, target real-time video surveillance data is created automatically and increasingly with face purifying, recognition, tracking, and labelling. Next, a CNN using the labelled data is finetuned. In [11], the researchers aim to utilize DL to detect face masks in the video automatically. The presented architecture comprises two mechanisms. Face detection and tracking are initially designed using machine learning and OpenCV; the facial frame is later processed to the presented deep transfer learning (TL) model MobileNetV2 for identifying the mask region. The

presented architecture was tried under dissimilar images and videos using the smartphone camera. Lei et al. [12] developed a hybrid module-based DL and visual tracking system to accomplish face detection. Firstly, a video sequence is classified into reference frame (RF) and non-reference frame (NRF). Next, the DL-based model in RF recognizes the target face. Meanwhile, the Kernelized-correlation-filter-based visual tracing methodology is utilized for accelerating FR. In [13], the authors proposed a toddler tracking system using a deep neural network (DNN). The presented technique depends on object tracking algorithms and faces recognition and is generated by a pretrained neural network. The system depends on recognizing the toddler's face and follows each toddler's movement in the house. Cárdenas et al. [14] developed a new methodology for face recognition in lower-resolution video depending on the morphology of the upper body of persons and the usage of CNN. Saypadith et al. [15] designed face detection architecture that is carried out on the embedded graphical processing unit (GPU) method.

3 The Proposed Model

In the study, a new HHODL-AFDT model has been established to detect and track faces in videos. The suggested HHODL-AFDT model follows two stages: Faster RCNN-based face detection model and HHO based hyperparameter optimization process. At the initial stage, the presented optimal Faster RCNN model precisely recognizes the face. Secondly, the tracking of faces takes place using the REGN model, which utilizes the features from neighbouring frames and foresees the face location in succeeding frames.

3.1 Face Detection: Faster RCNN Model

This work employs the Faster RCNN architecture to effectively identify faces in the input video frames [16]. R-CNN is a simple and novel technique as a pioneer advanced, providing above thirty percent mean average precision (mAP) compared to the earlier studies on PASCAL Visual Object Classes (VOC). The RCNN structure comprises four major stages. In the initial phase, the RCNN network resizes the image into 227×227 and takes them as input. The selection search technique for an image produces two thousand candidates of the presented bounding box as the warped region utilized for the input data. The network extracts a 4096-dimension vector from all the regions and later calculates the feature for all the regions. At last, utilizing the linear classification behindhand, the final layer categorizes the region for considering if any objects exist. In RCNN, the lower-level image feature (HOG) is substituted by the CNN feature, which is a discriminatory representation. But the evaluation of images is wasteful and highly expensive since RCNN should apply the convolution network two thousand times. In addition, resizing the input might create a problem that affects smaller objects that it easier to lose or deform data as the resolution changes from their original size. The region proposal overlapped, thus leading to the computation of notable features several times. With all the region proposals, it should be saved to disk beforehand implementing the feature extraction. Besides, numerous bounding boxes over-lapped will leads to a drop of mAP when the smaller object is closer to the big object since there is a bias to select the bounding box that comprises big and smaller objects.

Fast R-CNN [17] is a new technique that presents different innovations for improving the time of testing and training phases and effectively categorizing object proposals while improving the accuracy rate with the help of a deep convolution network. The structure of Faster RCNN is trained with a multi-task loss. Especially the convolution networks take images of any size as input and region of interest (RoI). Rather than RoI on an input, Faster RCNN employs the RoI on a feature map afterwards the convolution layer of the base network. The network comprises output vectors for each RoI: *softmax*

probability and bounding-box regression offset. The main characteristic of RoI is memory and shared computation in the backward and forward passes from a similar image. The great involvement of Faster RCNN is that it presents a novel training methodology that fixes the drawback of SPP-net and RCNN while improving the accuracy rate and running time. The benefit is that the mean average precision (mAP) is high compared to SPP-net and RCNN. The training stage is a single stage, with a multi-task loss, and could upgrade the network layer.

Fast R-CNN [18] is an innovative technique enhanced from Fast RCNN. Different from the two preceding techniques, rather than producing a bounding box with the external algorithm involved, Fast RCNN runs its technique named the region proposal network (RPN). Fig. 1 depicts the structure of RPN. Afterward, getting in-depth features from the previous convolution layer, RPN is considered and window slide over the feature maps for extracting features for all the region proposals. RPN is considered a full convolution network (FCN) that concurrently predicts the bounding box of objects and object score at every location. The intermediate layer is fed into 2 distinct subdivisions, one for object score (defines either the region is stuff or thing) and another for regression. The RPN enhances running time and accuracy besides preventing the generation of excess suggestion boxes since the RPN minimizes the cost by sharing computations on convolution features. Faster RCNN and RPN are combined into an individual network by sharing the convolution feature. This mixture assists Faster RCNN to have superior performance on accuracy; however, it results in the framework as a two-phase network that decreases the processing speed. Fig. 2 illustrates the overview of the RPN method.

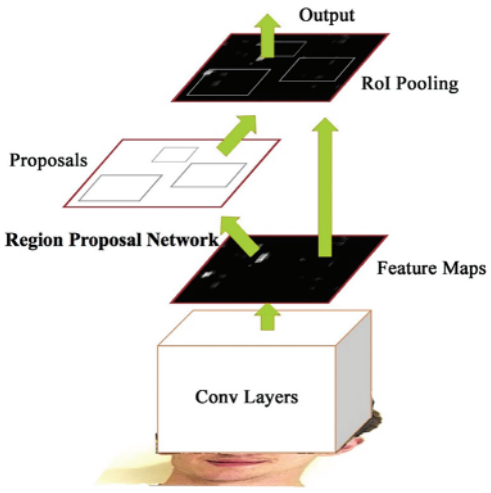

Figure 1: Structure of faster RCNN

3.2 Hyperparameter Tuning: HHO Algorithm

The HHO algorithm was exploited to tune the hyperparameters related to the Faster RCNN model. The HHO algorithm introduces many updating disciplines for individuals in the swarms for updating the position [19]. Four special conditions have been taken into account, and four kinds of updating methods are included, individual in the HHO swarm choose a method from the four according to the randomness and the escaping energy E.

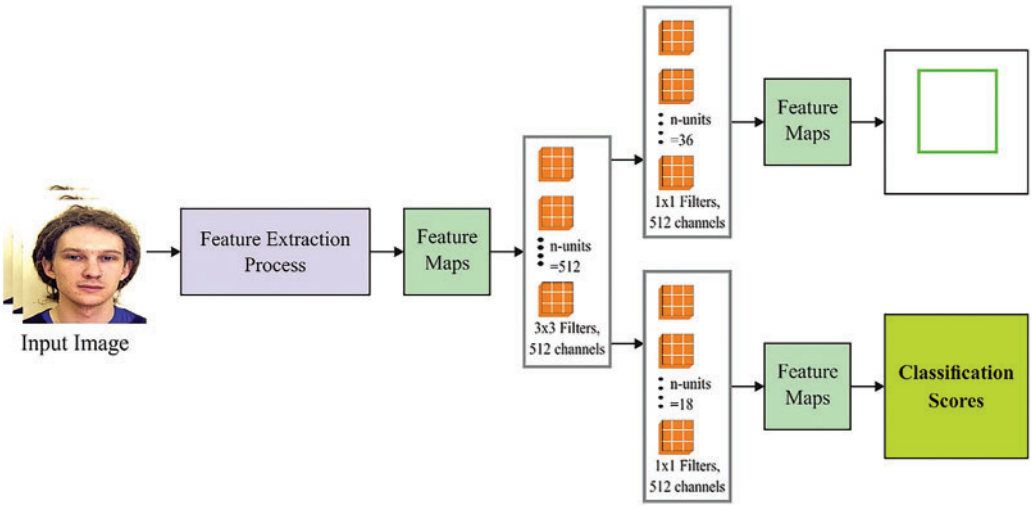

Figure 2: Overview of the RPN model

a) Exploration procedure

If the E escaping energy of the rabbit is smaller than -1 or larger than 1, then the individual in the HHO swarm explores the entire area very fast; two approaches are used:

$$X(t+L) = \{X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| \quad q \geq 0.5 \; P_b(t) - X_m(t) - r_3 (LB + r_4 (UB - LB)) \quad q < 0.5 \tag{1}$$

In Eq. (1), $X_{rand}(t)$ refers to a randomly chosen candidate at the current iteration, $X_m(t)$ indicates the average location of each individual at the existing iteration and is computed as follows. q refers to a random integer that lies between zero and one.

$$X_m(t) = \frac{1}{N} \sum_i^n X_i(t) \tag{2}$$

b) Exploitation process

Once the rabbit finds prey, the individual in the HHO swarm performs a procedure based on the status of the rabbit with smart action. The escaping energy of the rabbit controls such behaviors:

$$E = 2E_0 \left(1 - \frac{t}{maxIter}\right) \tag{3}$$

In Eq. (3), E_0 denotes the primary energy of the rabbit. If $|E| \geq 1$, the individual in the HHO swarm performs exploration; on the other hand, if $|E| < 1$, the individual implements exploitation and chooses one way to update the position according to a random number and the real-time escaping energy value.

i) Soft besiege if $r \geq 0.5$ and $|E| \geq 0.5$, individuals in the HHO swarm are aware that the rabbit keeps stronger and runs fast to escape, thus, flying nearby to the prey and attacking them, which is expressed as

$$X(t+1) = P_g(t) - X(t) - E \left| J \cdot P_g(t) - X(t) \right| \tag{4}$$

In Eq. (4), J indicates the capability of the prey to jump arbitrarily as follows:

$$J = 2\left(1 - r_5\right) \tag{5}$$

In Eq. (5), r_5 is another random integer between zero and one.

ii) Soft besiege with progressive fast dives. If $r < 0.5$ and $|E| \geq 0.5$, the prey energy is larger for escaping the capture, hence the HH should dive around the prey, and it is formulated by

$$X\left(t+1\right) = \{Y = P_g\left(t\right) - E\left|J \cdot P_g\left(t\right) - X\left(t\right)\right| \; f\left(Y\right) < f\left(X\left(t\right)\right)$$
$$Z = Y + r_6 \times LF\left(D\right) \; f\left(Z\right) < f\left(X\left(t\right)\right) \tag{6}$$

Here, r_6 denotes a randomly generated integer, and LF(D) indicates the Levy flight and is computed by:

$$LF\left(x\right) = 0.01 \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma\left(1+\beta\right) \times sinsin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2\left(\frac{\beta-1}{2}\right)}\right)^{\frac{1}{\beta}} \tag{7}$$

In Eq. (7), μ and v denote random values lying between zero and one, and β denotes a constant.

iii) Hard besiege. If $r \geq 0.5$ and $|E| < 0.5$, an individual in the HHO swarm performs using the lower escaping energy of a rabbit, they are eager to catch the prey and update the position related to the optimal global location:

$$X\left(t+1\right) = P_g\left(t\right) - E\left|P_g - X\left(t\right)\right| \tag{8}$$

iv) Hard besiege with progressive fast dives. If $r < 0.5$ and $|E| < 0.5$, the escape energy of the prey is lower escape. Hence the HH conducts a hard besiege and, lastly, grabs the prey, and the following equation formulates it.

$$X\left(t+1\right) = \{Y = P_g\left(t\right) - E\left|J \cdot P_g\left(t\right) - X_m\left(t\right)\right| \; f\left(Y\right) < f\left(X\left(t\right)\right)$$
$$Z = Y + r_6 \times LF\left(D\right) \; f\left(Z\right) < f\left(X\left(t\right)\right) \tag{9}$$

Algorithm 1: Pseudo-code of HHO algorithm

Inputs: N population size and T maximal quantity of iterations
Outputs: The rabbit position and its fitness values
Set the arbitrary population $X_i\,(i = 1,\ 2,\ \ldots\ ,\ N)$
while (ending state was not seen) perform
    Calculate the hawk fitness value
    Set X_{rabbit} as the rabbit location (optimal site)
    for (every hawk (X_i)) do
    Upgrade the jump strength J and primary energy E_0
    Upgrade the E
    if $(|E| \geq 1)$ after
    Upgrade the position vectors
    If $(|E| < 1)$, then
        If $(r \geq 0.5\ and\ |E| \geq 0.5)$ after

(Continued)

Algorithm 1: Continued

Upgrading the location vector
else if ($r \geq 0.5$ *and* $|E| < 0.5$) after
Upgrading the location vectors
else if ($r < 0.5$ *and* $|E| \geq 0.5$) after
Upgrading the location vectors
else if ($r < 0.5$ *and* $|E| < 0.5$) after
Upgrading the location vectors
Return X_{rabbit}

3.3 Face Tracking: REGN Model

In order to track the faces, the REGN model has been utilized in this work. The presented REGN model derives facial characteristics of the two nearby frames and recognizes the face location in the next frame [20]. In the REGN face tracking method, the place of moving face from the provided video sequences isn't altered much in 2 neighboring frames that also get prior data to the trained model. The connection between the present $\left(c_x^{'},\ c_y^{'}\right)$ and the preceding $\left(c_x,\ c_y\right)$ frame target center location coordinates are computed as:

$$c_x^{'} = c_x + w \cdot \Delta x,\ c_y^{'} = c_y + h \cdot \Delta y, \tag{10}$$

whereas w and h indicate the width and height of rectangular boxes, Δx and Δy were arbitrary variables, and it is given as follows:

$$f\ (\mu,\ b) = \frac{1}{2b}\ exp\ \left[-\frac{|x-\mu|}{b}\right]. \tag{11}$$

Get $\mu = 0$, $b = 1/5$ as to Eq. (11), and it is attained:

$$f\left(\mu = 0,\ b = \frac{1}{5}\right) = \frac{5}{2}\ exp\ [-1\,|x|]\,. \tag{12}$$

The connection amongst $(w',\ h')$ and $(w,\ h)$ target rectangle variables of present and preceding frames were determined:

$$h' = h \cdot \gamma_h,\ w' = w \cdot \gamma_w, \tag{13}$$

whereas γ_w and γ_h fulfil the Laplace distribution ($\mu = 1$ and $b = 1/15$). During this approach, this condition is suitable for our assumption of the motion of objects in effect. According to the trained REGN approach, it can be input the video orders as to the REGN approach, and afterwards, input the target face from the initialized window. It identifies the location $\left(c_x,\ c_y\right)$ of target faces from the primary frame of video sequences, whereas w and h were utilized as the initializing window. To face the tracking procedure on all the frames (that is, the t^{th} frame), the cropped image of the t^{th} frame and the preceding $t-1^{th}$ frame were utilized as the input of the REGN method. Next, the REGN approach forecasts the place of the target face on the $t+1^{th}$ frame and utilizes the forecasted place as the initialized window on the succeeding frames.

During the training stage of the method, the REGN system was trained on the provided video sequence. If the error of the REGN approach converges afterwards trained, this method is utilized for predicting a novel video sequence. During the face-tracked procedure of the novel video, the trained REGN approach was utilized to input the face place of the preceding frame and the searching area of

the present frame. Lastly, the REGN approach outcomes the face location of the following frame and recognizes the face track.

4 Experimental Results and Discussion

In these subsections, the experimental results analysis of the HHODL-AFDT model is tested using two benchmark datasets psychological image collection at stirling (PICS) and Visual tracker benchmark (VTB) dataset. The result analysis of the HHODL-AFDT algorithm on two benchmark datasets is elaborated in the following sections. Fig. 3 illustrates the sample face detection results of the HHODL-AFDT approach. The figure illustrates that the HHODL-AFDT algorithm has effectually recognized the faces under different conditions.

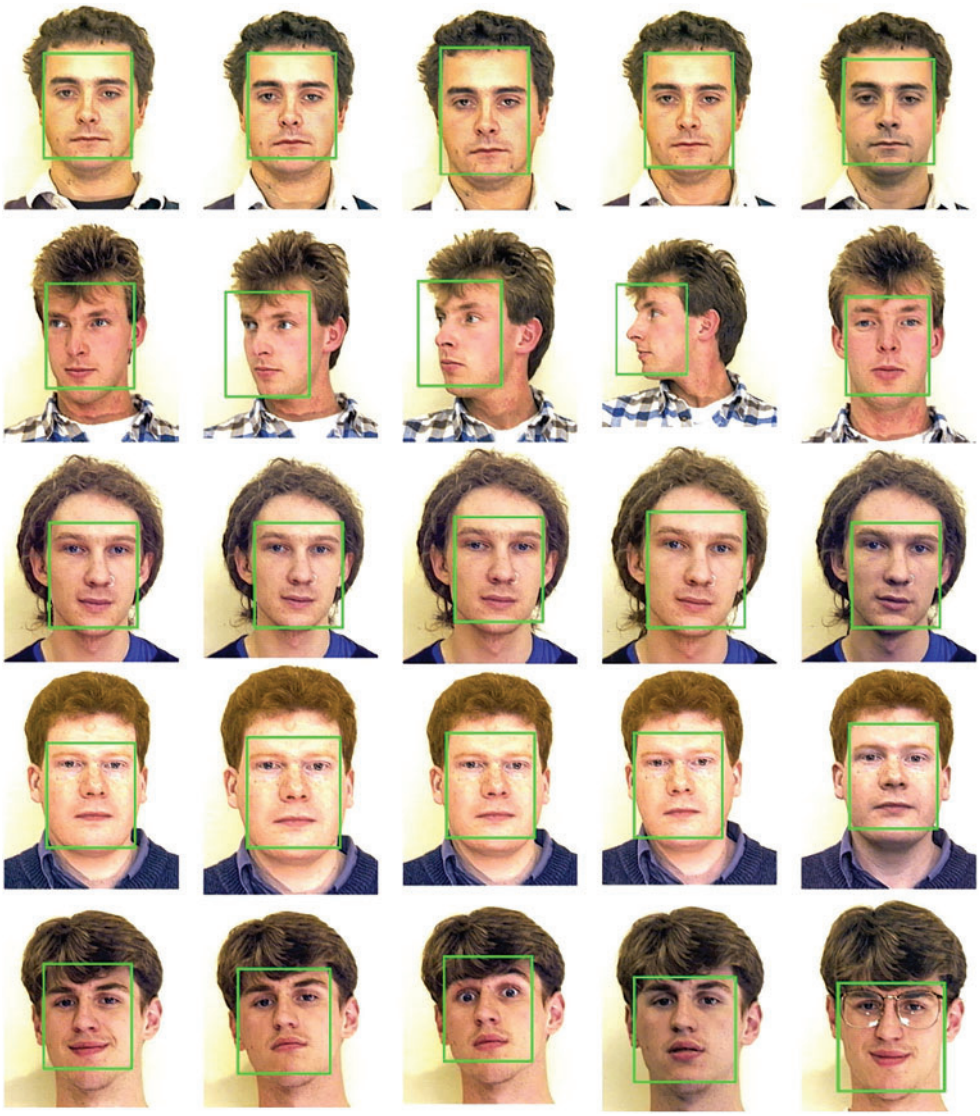

Figure 3: Visualization of detected faces by the proposed model

4.1 Result Analysis on PICS Dataset

The PICS [21] comprises a set of face images exploited for psychological experiments. In this work, we have chosen four facial image subsets that feature ambient light variations, face deflection, and expression modifications. The dataset contains 1698 images, as demonstrated in Table 1.

Table 1: PICS dataset details

Datasets	Images
Aberdeen	687
Pain	599
Stirling	312
Nottingham	100
Total	**1698**

Table 2 and Fig. 4 show the overall face recognition and tracking performance of the HHODL-AFDT technique on the test PICS dataset. The experimental outcome implied that the HHODL-AFDT approach had illustrated effective results with maximum accuracy values under all sub-datasets. For example, on the Aberdeen dataset, the HHODL-AFDT model has offered increased accuracy of 91.28% while the Ada-boost, YOLO, ResNets, discriminative correlation filters (DCF), convolutional support vector machines (CSVM), head-cascade RCNN (HRCNN), and SEN-ResNet models have shown reduced accuracy of 80.13%, 79.76%, 84.90%, 80.07%, 80.22%, 83.53%, and 88.51% correspondingly. Also, on the Stirling dataset, the HHODL-AFDT approach has increased accuracy by 92.68%. In contrast, the Ada-boost, YOLO, ResNets, DCF, CSVM, HRCNN, and SEN-ResNet methodologies have revealed decreased accuracy of 76.70%, 80.18%, 80.03%, 77.72%, 79.35%, 80.78%, and 88.90% correspondingly.

Table 2: Accuracy analysis of the HHODL-AFDT method with distinct classes under the PICS dataset

	Accuracy (%)							
Class name	Ada-boost	YOLO	ResNet	DCF	CSVM	HRCNN	SEN-ResNet	HHODL-AFDT
Aberdeen	80.13	79.76	84.90	80.07	80.22	83.53	88.51	91.28
Pain	78.34	81.98	83.28	78.18	79.85	86.76	85.87	89.06
Stirling	76.70	80.18	80.03	77.72	79.35	80.78	88.90	92.68
Nottingham	77.64	79.68	82.31	76.95	79.89	82.24	85.96	89.36
Average	**78.20**	**80.40**	**82.63**	**78.23**	**79.83**	**83.33**	**87.31**	**90.60**

A comparative execution time (ET) evaluation of the HHODL-AFDT model with current techniques is made in Table 3 and Fig. 5. The experimental outcomes implied that the HHODL-AFDT model had reached better results with minimal values of ET under all sub-datasets. For instance, on the Aberdeen dataset, the HHODL-AFDT method has a reduced ET of 2.13 s. In contrast, the Ada-boost, YOLO, ResNets, DCF, CSVM, HRCNN, and SEN-ResNet models have exhibited higher ET of 23.39, 22.18, 21.76, 13.36, 13.35, 6.65, and 4.16 s respectively. Eventually, on the Stirling dataset, the HHODL-AFDT approach resulted in a lower ET of 3.44 s. In contrast, the Ada-boost, YOLO,

ResNets, DCF, CSVM, HRCNN, and SEN-ResNet techniques have exhibited superior ET of 23.66, 19.70, 14, 12.84, 9.92, 6.96, and 5.19 s correspondingly.

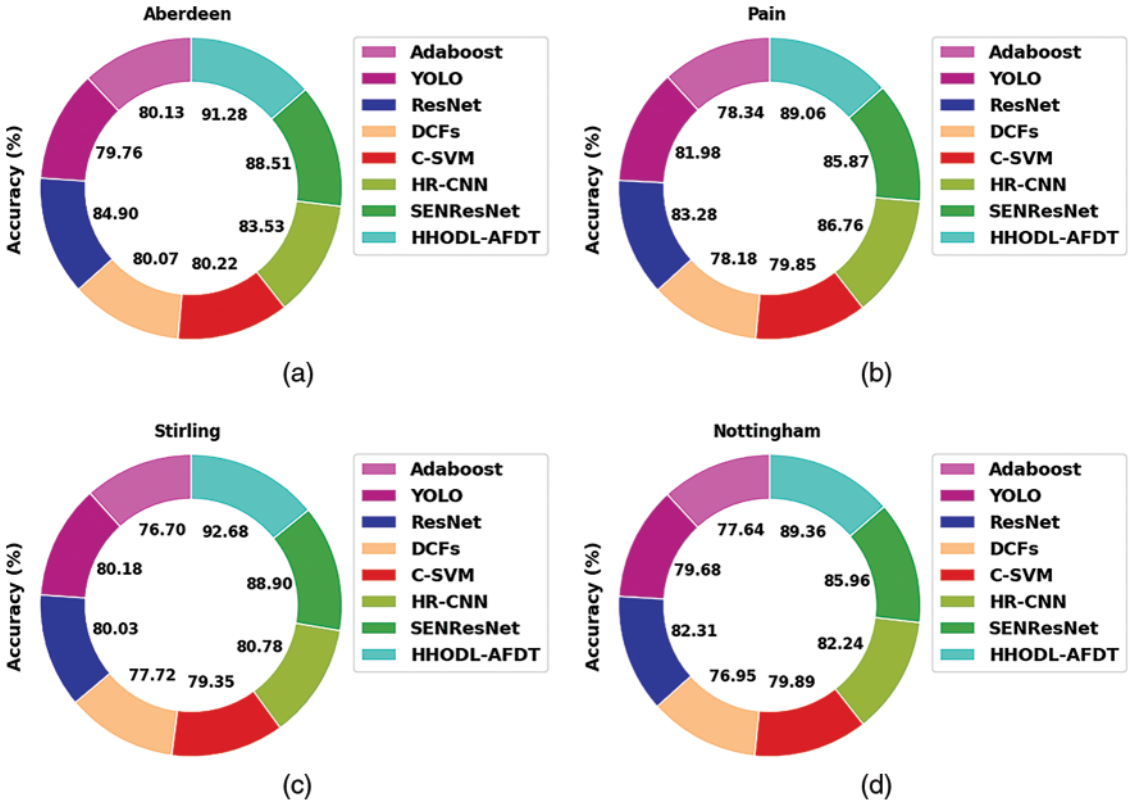

Figure 4: Accuracy analysis of HHODL-AFDT approach under PICS dataset (a) aberdeen, (b) pain, (c) stirling, and (d) nottingham

Table 3: Execution time analysis of HHODL-AFDT approach with distinct classes under the PICS dataset

	Execution time (sec)							
Class name	Ada-boost	YOLO	ResNets	DCF	CSVM	HRCNN	SEN-ResNet	HHODL-AFDT
Aberdeen	23.39	22.18	21.76	13.36	13.35	6.65	4.16	2.13
Pain	26.92	24.59	17.89	16.14	15.94	15.39	4.59	2.35
Stirling	23.66	19.70	14.00	12.84	9.92	6.96	5.19	3.44
Nottingham	26.99	24.68	20.55	17.97	15.18	13.00	3.59	2.22
Average	**25.24**	**22.79**	**18.55**	**15.08**	**13.60**	**10.50**	**4.38**	**2.54**

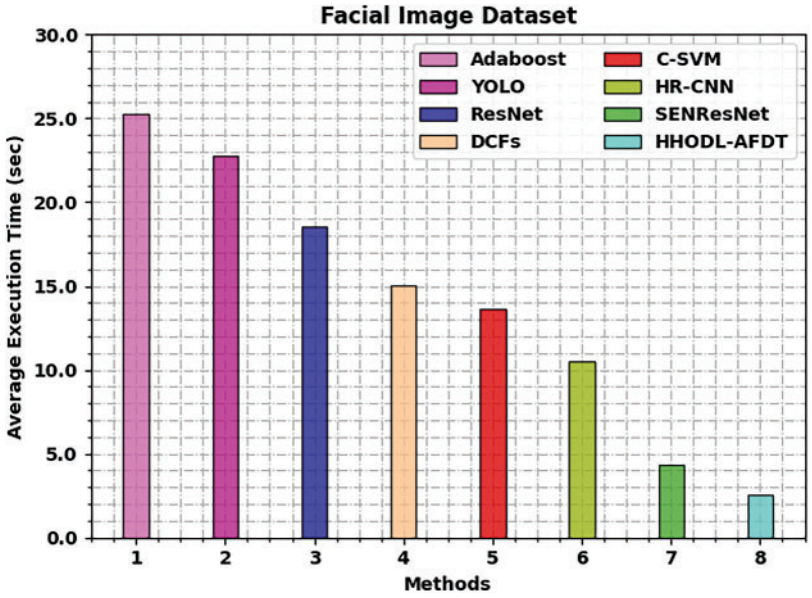

Figure 5: Average ET analysis of HHODL-AFDT technique under the PICS dataset

The average accuracy analysis of the HHODL-AFDT with existing models on the PICS dataset is described in Fig. 6. The figure exhibited that the HHODL-AFDT model has reached improved accuracy values under all training iterations compared to other models. In addition, it is observed that the Ada-boost and YOLO models have shown lower average accuracy values.

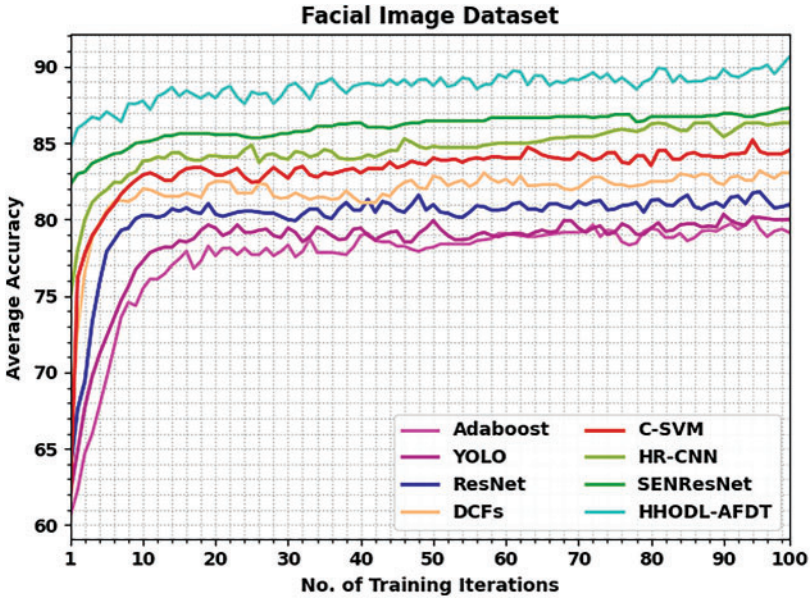

Figure 6: Average accuracy analysis of HHODL-AFDT approach on the PICS dataset

Fig. 7 provides a comprehensive loss graph inspection of the HHODL-AFDT and existing techniques on the test PICS dataset. The figure denoted that the Ada-boost and YOLO techniques have demonstrated poor performance with higher values of average loss. Followed by the HRCNN and

SEN-ResNet models have resulted in certainly reduced values of average loss. However, the HHODL-AFDT model has accomplished effectual performance over other models with lower average loss.

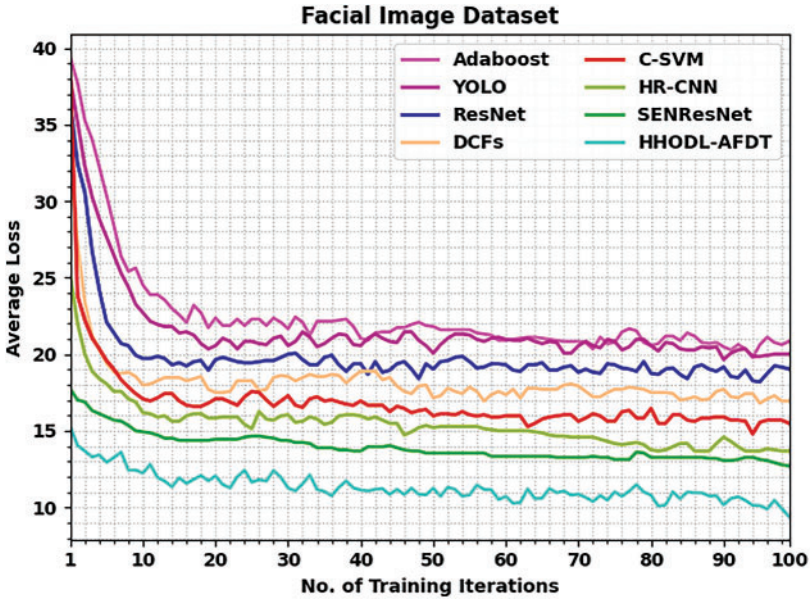

Figure 7: Average loss analysis of HHODL-AFDT approach on the PICS dataset

Fig. 8 demonstrates the receiver operating characteristic (ROC) curve results offered by the HHODL-AFDT methodology on the test PICS dataset. The results show that the HHODL-AFDT system has enhanced performance with maximum ROC values over the other models.

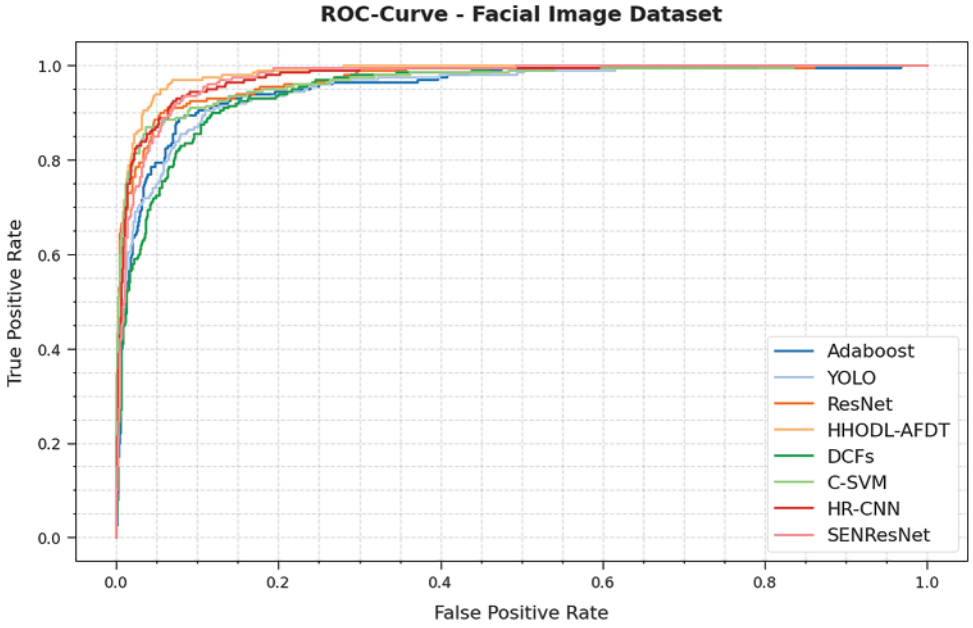

Figure 8: ROC analysis of HHODL-AFDT approach on the PICS dataset

4.2 Result Analysis of VTB Dataset

VTB [22] is a visual tracker benchmark for online visual tracking. Five facial video subsets can be selected that contain the features of changes in ambient light, variations from the human pose, and occlusion. The dataset contains 4014 frames, as shown in Table 4.

Table 4: VTB dataset details

Datasets	Frames
Dudek	1145
David	770
FaceOcc1	892
FleetFace	707
Girl	500
Total	**4014**

Table 5 offers the overall face recognition and tracking performance of the HHODL-AFDT methodology on the test VTB dataset. The outcome exposed that the HHODL-AFDT methodology has revealed effectual outcomes with maximal accuracy values under sub-datasets. The HHODL-AFDT technique has improved accuracy by 87.13% for the sample on the Dudek dataset. In contrast, the MeanShift, CamShift, Kernelized Correlation Filters (KCF), HRCNN, CSVM, Contour-motion feature (CMF), and Regression Network-based Face Tracking (RNFT) systems have outperformed reduced accuracy of 80.60%, 79.52%, 82.93%, 84.25%, 84.72%, 79.88%, and 83.49% correspondingly. Followed by the Girl dataset, the HHODL-AFDT approach has obtainable enhanced accuracy of 80.51%. In contrast, the MeanShift, CamShift, KCF, HRCNN, CSVM, CMFs, and RNFT techniques have outperformed lesser accuracy of 78.12%, 78.61%, 82.35%, 84.69%, 81.99%, 80.43%, and 85.75% correspondingly.

Table 5: Accuracy analysis of HHODL-AFDT approach with distinct classes under the VTB dataset

	Accuracy (%)							
Dataset	MeanShift	CamShift	KCF	HRCNN	CSVM	CMFs	RNFT	HHODL-AFDT
Dudek	80.60	79.52	82.93	84.25	84.72	79.88	83.49	87.13
David	78.91	80.66	82.85	87.19	81.51	82.07	86.97	90.15
FaceOcc1	76.51	75.31	81.31	85.47	80.95	79.82	85.37	88.30
FleetFace	76.81	80.02	82.15	83.28	80.00	79.90	85.99	91.50
Girl	78.12	78.61	82.35	84.69	81.99	80.43	85.75	80.51
Average	**78.19**	**78.82**	**82.32**	**84.98**	**81.83**	**80.42**	**85.51**	**87.52**

Table 6 and Fig. 9 depict an average accuracy performance of the HHODL-AFDT approach on the test VTB dataset with a distinct training set (TRS). The outcome implied that the HHODL-AFDT method had exhibited effectual results with higher average accuracy values under all TRS. For instance, on 500 TRS, the HHODL-AFDT approach has offered increased average accuracy of 81.40%. In contrast, the MeanShift, CamShift, KCF, HRCNN, CSVM, CMFs, and RNFT models

have shown lower average accuracy of 76.52%, 77.56%, 79.17%, 79.74%, 78.28%, 79.22%, and 80.72% respectively. Besides, on 2000 TRS, the HHODL-AFDT methodology has an obtainable increased average accuracy of 88.08%. In contrast, the MeanShift, CamShift, KCF, HRCNN, CSVM, CMFs, and RNFT methodologies have shown decreased average accuracy of 80.72%, 81.97%, 83.88%, 87.20%, 85.49%, 83.78%, and 87.41% correspondingly.

Table 6: Average Accuracy analysis of HHODL-AFDT approach with distinct training set under VTB dataset

	Average accuracy							
Size of training set	MeanShift	CamShift	KCF	HRCNN	CSVM	CMFs	RNFT	HHODL-AFDT
500	76.52	77.56	79.17	79.74	78.28	79.22	80.72	81.40
100	77.40	78.65	80.36	83.00	81.29	80.57	83.37	84.35
1500	79.89	80.36	82.28	84.35	83.11	81.66	85.18	86.11
2000	80.72	81.97	83.88	87.20	85.49	83.78	87.41	88.08

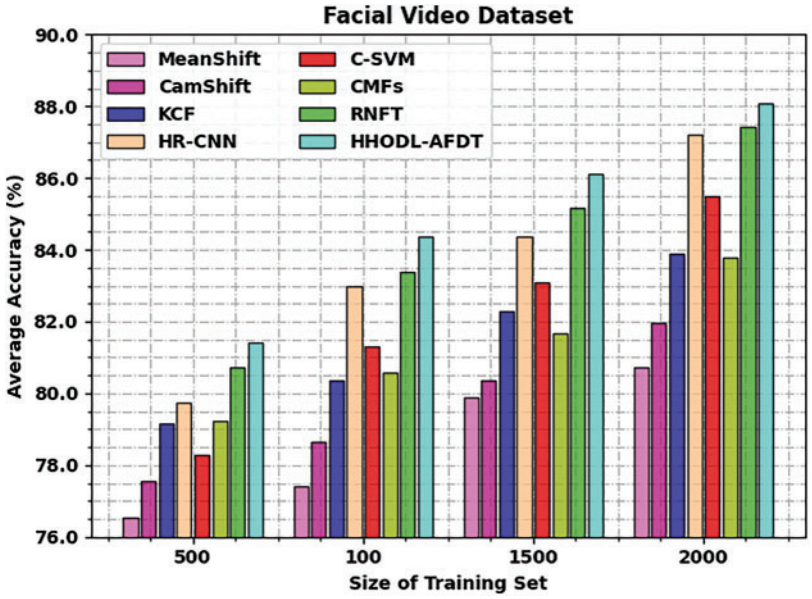

Figure 9: Average accuracy analysis of HHODL-AFDT approach under the VTB dataset

A comparative ET estimation of the HHODL-AFDT model algorithm with recent approaches under the VTB dataset is demonstrated in Table 7. The outcome revealed that the HHODL-AFDT method had reached optimum outcomes with decreased values of ET under all sub-datasets. For sample, on the Dudek dataset, the HHODL-AFDT technique has resulted in a minimal ET of 2.15 s. In contrast, the MeanShift, CamShift, KCF, HRCNN, CSVM, CMFs, and RNFT methodologies have exhibited higher ET of 27.78, 22.11, 17.64, 10.61, 9.42, 9.16, and 4.33 s correspondingly. Finally, on the Girl dataset, the HHODL-AFDT technique has resulted in a reduction of ET of 3.72 s. In contrast, the MeanShift, CamShift, KCF, HRCNN, CSVM, CMFs, and RNFT methodologies have exhibited

superior ET of 23.35, 19.86, 17.92, 17.78, 16.08, 13.27, and 5.37 s correspondingly. From the detailed discussion and outcomes, it is ensured that the HHODL-AFDT algorithm has illustrated enhanced performance over other techniques.

Table 7: Execution time analysis of HHODL-AFDT approach with distinct classes under the VTB dataset

	Execution time (sec)							
Dataset	MeanShift	CamShift	KCF	HRCNN	CSVM	CMFs	RNFT	HHODL-AFDT
Dudek	27.78	22.11	17.64	10.61	9.42	9.16	4.33	2.15
David	27.65	27.12	13.88	13.68	13.38	6.29	5.76	3.44
FaceOcc1	27.96	22.67	17.73	11.81	9.65	5.55	3.92	1.61
FleetFace	27.02	26.70	22.17	20.34	17.94	5.62	3.83	1.96
Girl	23.35	19.86	17.92	17.78	16.08	13.27	5.37	3.72
Average	**26.75**	**23.69**	**17.87**	**14.84**	**13.29**	**7.98**	**4.64**	**2.58**

5 Conclusion

In the study, a new HHODL-AFDT model has been established to recognise and track faces in videos. The presented HHODL-AFDT model follows a two-stage process: Faster RCNN-based face detection model and HHO based hyperparameter optimization process. At the initial stage, the presented optimal Faster RCNN model precisely recognizes the face. Secondly, the tracking of faces takes place using the REGN model, which utilizes the features from neighboring frames and foresees the location of the target face in succeeding frames. The experimental validation of the HHODL-AFDT technique is conducted under two datasets, and the outcome is inspected using various aspects. The experiment results highlighted the superior performance of the HHODL-AFDT model over current approaches. In the future, an ensemble learning process can be introduced to improve face recognition performance by using multiple DL models.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R349), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University Project Number (PSAU/2023/R/1444).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Zheng and Y. Xu, "Efficient face detection and tracking in video sequences based on deep learning," *Information Sciences*, vol. 568, pp. 265–285, 2021.

[2] A. Das, M. Pukhrambam and A. Saha, "Real-time robust face detection and tracking using extended haar functions and improved boosting algorithm," in *Int. Conf. on Green Computing and Internet of Things (ICGCIoT)*, Greater Noida, India, pp. 981–985, 2015.

[3] K. Goyal, K. Agarwal and R. Kumar, "Face detection and tracking: Using OpenCV," in *Int. Conf. of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, vol. 1, pp. 474–478, 2017.
[4] D. Schofield, A. Nagrani, A. Zisserman, M. Hayashi, T. Matsuzawa *et al.,* "Chimpanzee face recognition from videos in the wild using deep learning," *Science Advances*, vol. 5, no. 9, pp. eaaw0736, 2019.
[5] N. Prabakaran, S. S. S. Kumar, P. K. Kiran and P. Supriya, "A deep learning based social distance analyzer with person detection and tracking using region based convolutional neural networks for novel coronavirus," *Journal of Mobile Multimedia*, vol. 18, no. 3, pp. 541–560, 2022.
[6] A. Kwaśniewska, J. Rumiński and P. Rad, "Deep features class activation map for thermal face detection and tracking," in *10th Int. Conf. on Human System Interactions (HSI)*, Ulsan, Korea (South), pp. 41–47, 2017.
[7] V. Arulkumar, S. J. Prakash, E. K. Subramanian and N. Thangadurai, "An intelligent face detection by corner detection using special morphological masking system and fast algorithm," in *2021 2nd Int. Conf. on Smart Electronics and Communication (ICOSEC)*, Trichy, India, pp. 1556–1561, 2021.
[8] J. Dong and X. Xie, "Visually maintained image disturbance against deepfake face swapping," in *2021 IEEE Int. Conf. on Multimedia and Expo. (ICME)*, Shenzhen, China, pp. 1–6, 2021.
[9] M. Chowdhury, I. Hossain, N. M. Sakib, S. M. M. Ahmed, M. Zeyad *et al.,* "Human face detection and recognition protection system based on machine learning algorithms with proposed ar technology," in *Advances in Augmented Reality and Virtual Reality, Studies in Computational Intelligence Book Series*, Singapore: Springer, vol. 998, pp. 177–192, 2022.
[10] Y. Wang, T. Bao, C. Ding and M. Zhu, "Face recognition in real-world surveillance videos with deep learning method," in *2nd Int. Conf. on Image, Vision and Computing (ICIVC)*, Chengdu, China, pp. 239–243, 2017.
[11] S. Asif, Y. Wenhui, Y. Tao, S. Jinhai and K. Amjad, "Real time face mask detection system using transfer learning with machine learning method in the era of COVID-19 pandemic," in *4th Int. Conf. on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, pp. 70–75, 2021.
[12] Z. Lei, X. Zhang, S. Yang, Z. Ren, O. F. Akindipe *et al.,* "RFR-DLVT: A hybrid method for real-time face recognition using deep learning and visual tracking," *Enterprise Information Systems*, vol. 14, no. 9–10, pp. 1379–1393, 2020.
[13] H. GÜney, M. Aydin, M. TaŞkiran and N. Kahraman "Toddler tracking system with face recognition and object tracking using deep neural network," in *Int. Conf. on Innovations in Intelligent Systems and Applications (INISTA)*, Novi Sad, Serbia, pp. 1–6, 2020.
[14] R. J. Cárdenas, C. A. Beltrán and J. C. Gutiérrez, "Small face detection using deep learning on surveillance videos," *Environment*, vol. 2, no. 5, pp. 14, 2019.
[15] S. Saypadith and S. Aramvith, "Real-time multiple face recognition using deep learning on embedded GPU system," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC)*, Honolulu, HI, USA, pp. 1318–1324, 2018.
[16] N. -D. Nguyen, T. Do, T. D. Ngo and D. -D. Le, "An evaluation of deep learning methods for small object detection," *Journal of Electrical and Computer Engineering*, vol. 2020, pp. 1–18, 2020.
[17] R. Girshick, "Fast R-CNN," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 1440–1448, 2015.
[18] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[19] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, M. Hassaballah *et al.,* "A novel hybrid harris hawks optimization and support vector machines for drug design and discovery," *Computers & Chemical Engineering*, vol. 133, pp. 106656, 2020.
[20] Y. Li, J. Wan, Q. Miao, S. Escalera, H. Fang *et al.,* "Cr-net: A deep classification-regression network for multimodal apparent personality analysis," *International Journal of Computer Vision*, vol. 128, no. 12, pp. 2763–2780, 2020.

[21] U. of Stirling, Psychological image collection at stirling (pics), 2022. Website, http://pics.stir.ac.uk
[22] Google, Visual tracker benchmark, 2022. Website, http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.