Deep-Learning Approach for Efficient Eye-blink Detection with Hybrid Optimization Concept

Rupali Gawande, Sumit Badotra*
Department of Computer Engineering, Lovely Professional University
Phagwara, Punjab India

Abstract*—In this research work, a novel eye-blink detection model is developed. The proposed eye blink detection model is modeled by following seven major phases: (a) video–to-frame conversion, (b) Pre-processing, (c) face detection, (d) eye region localization, (e) eye landmark detection and eye status detection, (f) eye blink detection and (f) Eye blink Classification. Initially, from the collected raw video sequence (input), each individual frames are extracted in the video –to-frame conversion phase. Then, each of the frames is subjected to pre-processing phase, where the quality of the image in the frames is improved using proposed Kernel median filtering (KMF) approach. In the face detection phase, the Viola-Jones Model has been utilized. Then, from the detected faces, the eye region is localization within the proposed eye region localization phase. The proposed Eye region localization phase encapsulates two major phases: Feature extraction and landmark detection. The features like improved active shape models (I-ASMs), Local Binary pattern are extracted from the detected facial images. Then, the eye region is localization by using a new optimized Convolution neural network framework. This optimized CNN framework is trained with the extracted features (I-ASM and LBP). Moreover, to enhance the classification accuracy of eye localization, the weight of CNN is fine-tuned using a new Seagull Optimization with Enhanced Exploration (SOEE), which is the improved version of standard Seagull Optimization Algorithm (SOA). The outcome from optimized CNN framework is providing the exact location of the eye region. Once the eye region is detected, it is essential to detect the status of the eye (whether open or close). The status of the eye is detected by computing the eye aspect ratio (EAR). Then, the identified eye blinks are classified based on the computed correlation coefficient as long and short blinks. Finally, a comparative evaluation has been accomplished to validate the projected model.

Keywords*—Eye localization; CNN; Seagull Optimization with Enhanced Exploration (SOEE); improved active shape model (I-ASM); eye aspect ratio (EAR); eye-blink detection

I. Introduction

Recently, the Eye-tracking and/or -blinking detection algorithms are playing a significant role in the all the spheres of the world, more particularly in the smartphones that are revalorizing the entire globe. The smart phones have become ubiquities in our routing life style. We rely upon the smartphones for storing our confidential information. In the smartphones, the spoofing attacks can be avoided by utilizing the eye blink detection techniques within the face recognition systems [1-4] Moreover, the eye- blink detection systems in the smartphones provides information regarding the eye-blinking habits of the users, and provides advices [5-8]. In fact, eye blinking is said to be the partly subconscious quick closing as well as reopening of the eyelids. The eye blinking is carried out with the aid of multiple muscles. The eye opening as well as closing is managed by the *orbicularis oculi* and *levator palpebrae superioris* [9-11]. In fact, the eye corona moistens by the eye blinking process. The frequency and duration are the prime factors for blinking. For an adult, the Average blinking frequency is 15-20 blinks/min, while the blinking frequency of the children is lower [12-15].

In literature, massive count of researchers has been devoted for eye blink detection. The outcome from them is either blink detection or eye status reorganization (open/closed eye). Most of the eye-tracking and eye-blinking detection algorithms rely upon the video input [16]. However, the techniques developed for desktop environments are not suitable for mobile/smartphone environments, owing towards the major shortcoming in terms of computational resources. A promising solution to this problem is to make use of the region of interest (ROI), wherein a smaller region is alone examined rather than searching for the location of eye in the whole image. For real-time processing, the eye localization is indeed a suitable approach for ROI-based solution. Once, the eye region is localized, its status is tracked using the deep learning models like convolutional neural network (CNN) and Artificial Neural Network (ANN) as well. Among the entire deep learning model utilized for eye blink detraction, the CNN is a renowned one [17]. Although the CNN is successful in eye blink detection, it is till been considered to be a computational complexity model, owing towards its higher time consumption in training the parameters.

On the basis of this research gap is identified objective are identified as below:

- To propose an enhanced model for face localization from video frame.
- To propose an effective technique for eye pair localization from captured frame.
- To propose eye blink detection technique for left, right and double click instance.
- To evaluate effectiveness of the proposed technique.

The major contribution of this research work is: The research work Introduces a Kernel median filtering (KMF) approach for de-noising the image frames. Work also Introduces an improved active shape models (I-ASMs) in the Eye region localization phase to exactly localize the eye

*Corresponding Author.

region. It introduces an optimized CNN model for precise Eye region localization. The weight of CNN is fine-tuned using the newly projected Seagull Optimization with Enhanced Exploration (SOEE) model. This Seagull Optimization with Enhanced Exploration (SOEE) is the improved version of standard Seagull Optimization Algorithm (SOA) . At the end of the research the accuracy of the system is compared with previously present technologies.

The leftover parts of this paper are arranged as: The literature review on eye blink detection is manifested in Section II. The proposed eye-blink detection framework: an overview is addressed in Section III. The video-to-frame conversion and pre-processing via proposed kernel median filtering, face detection via viola Jones model and eye localization via optimized CNN is portrayed in Section IV, Section V and Section VI, respectively. Moreover, eye status detection and eye blink detection and classification are portrayed in Section VII and Section VIII, respectively. The results acquired as well are discussed comprehensively in Section IX. This paper is concluded in Section X.

II. Related Work

In 2019, Sharmila et al. [1] have proposed a new technique for Eye Blink Detection model for preventing Computer Vision Syndrome (CVS). The proposed work uses Viola Jones algorithm for detecting the eyes, eye blink using background subtraction, gradient based corner detection and it is capable of detecting common cases of fatigued behaviour linked with prolonged computer use by tracking the eye blink rate. Hence, this proposed system could significantly reduce the symptoms among regular computer users leading to improved health habits.

In 2021, Rajamohana et al. [2] have proposed a novel automated, real-time driver's drowsiness detection framework using the proposed hybrid framework. The proposed model has formulated by blending the CNN and Bidirectional Long Term Dependencies (BiLSTM). The driver's facial image and eye blinks have been tracked using the Video camera. The projected model has encapsulated three major phases: identification of the driver's face image using a web camera, extraction of eye image features using the Euclidean algorithm and continuous eye blinks monitoring. As a n outcome, the status of the eye is exhibited as either open or close.

In 2021, Quddu et al. [3] have proposed a novel eye-tracking system using the RNN and LSTM. The authors have utilized the RNN to detect the drowsiness. In addition, the eye movements were modelled using the LSTM. Two types of LSTMs: 1-D LSTM (R-LSTM) and convolutional LSTM (C-LSTM) has been utilized.

In 2020, Liu et al. [4] have proposed a fatigue detection algorithm on the basis of the analyzed deeply-learned facial expression. Initially, the multi block local binary patterns (MB-LBP) and Adaboost classifier has been utilized for training the face key point detection model. In addition, the 24 facial points were identified using the trained model. Moreover, the proportion of the closed eye time within the unit time (PERCLOS) and yawning frequency were computed. The driver's fatigue state was deduced using the fuzzy inference system. The projected model could detect the degree of the driver fatigue accurately as well as quickly.

In 2019, Wang et al. [5] have presented a robust fatigue detection system based on the binocular consistency using the deep learning. The eye gaze has been detected using the dual-stream bidirectional convolutional neural network (BCNN). The GP-BCNN was constructed by including the vectorized local integral projection features within the Gabor filters and the projection vectors for overcoming the orientation and scale changes. Further based on the pupil distance, the eye screening mechanism (ESM) was projected with the intention of eliminating the detected errors. The projected model has yielded a higher accuracy rate.

In 2019, Selvathi et al. [6] have proposed a fatigue and drowsiness detection system in FPGA using the deep neural network. Initially, the collected images were pre-processed via median filtering. Then, the facial regions were identified using the Viola Jones face detection algorithm. Further, the authors have extracted the Local Binary Pattern features from the detected faces, and have employed the Max pooling to lessen the level of complexity. The final detection was carried out using the SVM classifier. The results acquired had exhibited that the projected model is applicable for real time driver's vigilance monitoring.

In 2020, Lamba et al. [7] have utilized the feature level fusion (MmERMFLF) for precise multimodal eye blink recognition. The status of the eye was computed using the eye-eyebrow facet ratio (EEBFR), and eyebrow to nose facet ratio. From eye blinks, pulse rate as well as behavioral patterns (emotions), the authors have sensed the emergency state using the improved intellectual framework. Moreover, the count of eye blinks was detected using the novel multimodal method (MmERMFLF). The projected model had achieved improved reorganization accuracy.

In 2019, Zemblys et al. [8] have developed gazeNet for event detector's generation. The projected model has encapsulated an end-to-end deep learning approach, which was trained using the raw eye-tracking data and have classified it as fixations, saccades and post-saccadic oscillations. The projected model hasn't been utilized in large scale owing to its expanded training time. Table I summarizes the comparison between the existing approaches.

TABLE I. COMPARISON OF EXISTING APPROACHES

Ref No	Published Year	Approach	Advantages	Disadvantages
[2]	2021	Convolution neural Network	Convolution neural network is useful in facial recognition CNN is having higher accuracy	It is not more useful in speech and voice recognition CNN requires lots of data
[4]	2019	Facial Feature	It reduces computational cost	Vector used for facial feature finding is high
[15]	2017	Template Marching	It is easier to implement	If template used is not proper then accuracy of output is hampered
[22]	2016	Component Based	It is faster and more accurate face detection	For more accuracy larger vector should be used.
[28]	2015	Knowledge based	It reduces computational complexity	Detection rate of this technique is slower

III. PROPOSED EYE-BLINK DETECTION FRAMEWORK: AN OVERVIEW

A. Architectural Description

The proposed eye blink detection model will be modeled by following seven major phases: (a) video–to-frame conversion, (b) Pre-processing, (c) face detection, (d) eye region localization, (e) eye landmark detection and eye status detection, (f) eye blink detection and (f) Eye blink Classification. The architecture of the proposed work is manifested in Fig. 1.

- Initially, from the collected raw video sequence (input), each individual frames are extracted in the video –to-frame conversion phase.
- Then, each of the frames is subjected to pre-processing phase, where the quality of the image in the frames is improved using proposed Kernel median filtering (KMF) approach. At the end of the pre-processing phase, noiseless and higher quality video frames are acquired. These pre-processed frames are subjected to face detection phase.
- In the face detection phase, the facial region (ROI) in the pre-processed frames is detected by suppressing the rest of the regions (Non- ROI). The face detection is accomplished via Viola-Jones Model.
- Then, from the detected faces, the eye region is localization within the pro-posed eye region localization phase. The proposed Eye region localization phase encapsulates two major phases: Feature extraction and landmark detection. The features like improved active shape models (I-ASMs), LBP are extracted from the detected facial images. And then the eye region is localization by using a new optimized CNN framework. This optimized CNN framework is trained with the extracted features (I-ASM and LBP). Moreover, to enhance the classification accuracy of eye localization, the weight of CNN is fine-tuned using a new Seagull Optimization with Enhanced Exploration (SOEE), which is the improved version of standard Seagull Optimization Algorithm (SOA) [9].

The outcome from optimized CNN framework is providing the exact location of the eye region. Once the eye region is detected, it is essential to detect the status of the eye (whether open or close).

Eye status detection: The status of the eye is detected by computing the eye aspect ratio (EAR). The aspect ratio is calculated between the points of eye to check whether eyes are opened or closed.

Eye-blink detection: It is performed onto the computed eye landmarks using the normalized cross-correlation method. Then, the identified eye blinks are classified based on the computed correlation coefficient as long and short blinks.

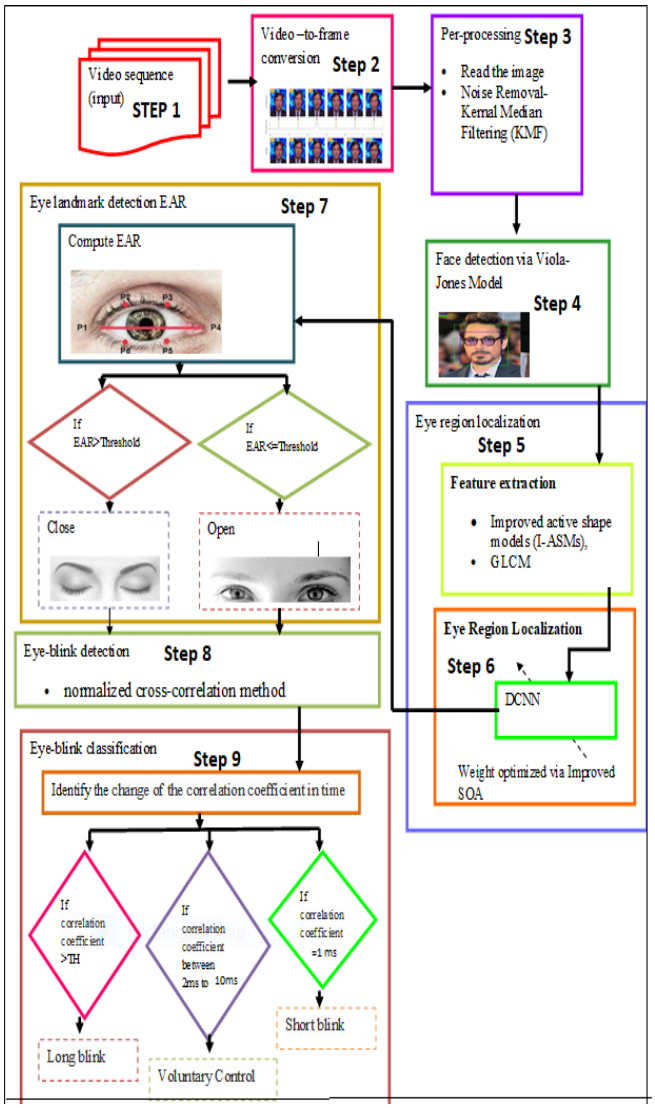


Fig. 1. Architecture of the Proposed Work.

IV. Video-to-Frame Conversion and Pre-Processing Via Proposed Kernel Median Filtering

A. Video-to-Frame Conversion

Let the collected input video sequence be denoted as, which includes count of frames. The video is indeed a medium that has become more important in today's society. It is made up of a series of frames, each of which is divided into shots and scenes, with the video being made up of a mixture of these scenes. To evaluate any video, we must first examine the features of the frames and then assess the video's attributes, which is accomplished using python' (OpenCv library) video to frame converter. Let the extracted frames be denoted as. The noise in is removed in the pre-processing stage using the proposed Kernel Median Filtering (KMF). Here, are the pixels.

B. Pre-Processing via Kernel Median Filtering

Preprocessing is a set of processes that aim to rectify or compensate for systematic mistakes in input before it is analyzed further. Owing to low collected integrity, every sort of information need pre-processing to improve it for future processing.

Pre-processing is required for the following reasons:

1) Light propagation qualities like scattering and absorption cause image deterioration.

2) Video captures with a high level of specificity, such as an unknown stiff scene, un-known hue, or poor light sensitivity. As a consequence, an initiative has been taken to discover the perfect filtering for pre-processing the frames. Filtering reduce noise by reducing statistical variances. It contributes to the reduction of visual noise. Noise affects the acquisition or capture of an image. Filtering methods are used to process the image. The ultimate focus of the filter has always been to minimize picture noise and improve the input image. Prior to further analysis and processing, it's indeed vital to optimize image quality.

The pre-processing stages utilized in this research work are

a) Read the image in the frames

b) De-Noise using proposed KMF

Each of the frames in is read using the OpenCV Library Cv2.imread() function From this , the noise is removed in the pre-processing stage using the proposed Kernel Median Filtering (KMF).

Median filtering is a nonlinear procedure that may be used to reduce impulsive noise, sometimes known as salt-and-pepper noise. It also is beneficial for retaining picture borders despite decreasing random noise. A random bit error in a line of communication may cause spontaneous or salt-and-pepper noise. Across an image frame, a window is moved in the median filter, and then for each pixel inside the window, its median intensity value is computed. This median intensity value is output intensity of the pixel being processed. The median filter has a significant drawback in that every out-come has always been bound by specification to have been the window's median value. The centre value substituted also isn't evaluated to see whether it's an impulse or not. Whenever the intensity of the noise is considerable, the median filter works miserably. Therefore, a new Kernel Median Filtering (KMF) is introduced in this research work.

The steps followed in the Kernel Median Filtering (KMF) are manifested below:

1) The Kernel Median Filtering (KMF) detects the noisy pixels at first and then filters it as shown in Fig. 2.

2) Once, the noise is filtered; the pixels are validated to check whether it is a thin line or an edge pixel.

3) The input frame is convoluted with 4 convolutional kernels (sigmoid, RBF, Lapla-cian, Chi2). The minimal difference of these four convolutions is used for edge detection. (1)

$$\text{Cij } \min\{V^{F_i(a,b)} \otimes W_k \,|\, K = 1-4\} = \quad (1)$$

4) Compare the value of with Threshold (say 1000). If the minimal value of the pixel is less than the threshold value, then the current pixel value is the same as it, else the standard median filtering is applied, and the median value is returned as the pixel value. This is mathematically shown in Eq. (2).

$$V^{KMF} = \begin{cases} Y^{MF} & ;If\ C_{ij} \geq T \\ V^{F_i(a,b)} & Otherwise \end{cases} \quad (2)$$

The pre-processed outcome is subjected to face detection phase, wherein the facial region is detected using the standard Viola-Jones Model.

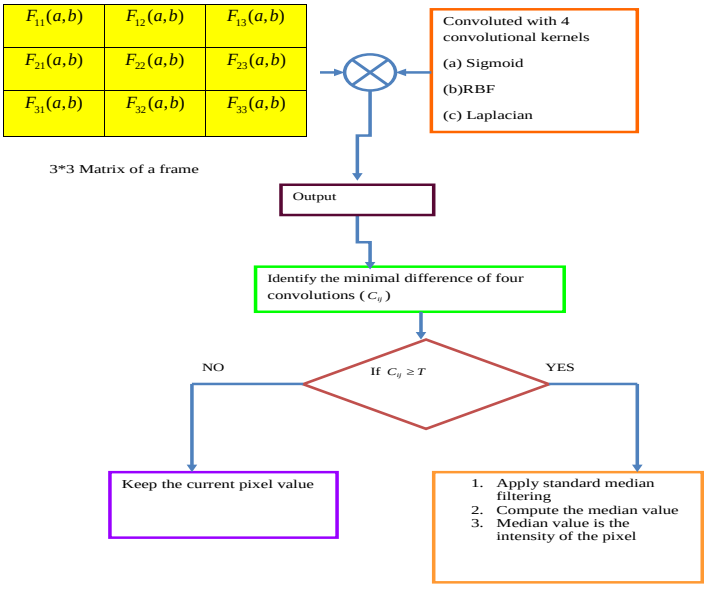


Fig. 2. Proposed Kernel Median Filtering.

V. Face Detection via Viola Jones Model

A. Viola-Jones Model

The pre-processed image is subjected to viola-jones model for face region detection. The Viola-Jones algorithm, created by Paul Viola and Michael Jones in 2001, is indeed an object-

recognition framework that enables for real-time identification of visual properties. Although becoming an obsolete paradigm, Viola-Jones seems to be remarkably efficient, as well as its applicability in real-time face identification has conclusively demonstrated to be particularly noteworthy [10].

The Viola-Jones Algorithm has two stages:

1) Training
2) Detection

1) Training: Training data is the initial dataset used to train machine learning algorithms. Models create and refine their rules using this data. It's a set of data samples used to fit the parameters of a machine learning model to training it by example.

2) Detection: Viola-Jones had been created for frontal face images; therefore this detects them superior over face images that are oriented sideways, upwards, or downwards. The image is transformed to grayscale before its being used to recognize a face since it is necessary to focus with less data to analyze. The Viola-Jones method discovers the position on the coloured picture after detecting the face in the grayscale image. Viola-Jones constructs a box and explores for a face within it. It's obviously looking for haar-like characteristics, which would be detailed later. After moving through each and every square inside the picture, the box advances a step to the right. A multitude of boxes recognize face-like characteristics (Haar-like features) employing smaller steps, and the information from all of those boxes aggregated supports the algorithm towards determining in which the face resides.

3) Haar-like features: The characteristics underneath depict a box with such a dark and bright side, which the machine helps to define whatever the feature represents. As in the edge of a brow, one side may well be brighter than another at sometimes. The central area of the box may be brighter than that of the neighboring boxes, which might be misinterpreted for a nose. Viola and Jones observed three sorts of Haar-like characteristics in their research: • Four-sided features • Edge features • Line features [11]

4) Integral images: The integral picture aids us in performing these time-consuming computations fast enough that we may determine if a feature of a group of features meets the requirements [12].

5) Training classifiers: We're training the machine to recognize these characteristics. We're feeding it data and then training this to understand from it in order to make predictions. In the end, the algorithm produces whether any of the frames may be categorized as a feature or not by defining a minimal threshold [13-16].

6) Adaptive boosting (AdaBoost): In general, the boosting is an Ensemble method that is formulated by means of combing several weak learners. As a resultant a strong learner is formulated. The most popularly utilized boosting models is the Adaptive Boosting approach. The adaboost solves the problem of “curse in dimensionality” issue. Even though, the formulated integral image has reduced the dimensionality of the image, still its features remain quite higher, and this tends to lessen the training accuracy. Therefore, the adaboost has been implied [17].

7) Cascading: The cascading takes place next to the adaboosting model. In the cascading model, each of the sub-windows is validated to check whether the most important features are available within it or not. If the corresponding sub-window has the most important feature, then the second important features are explored, else the sub-window is discarded. This process is referred as cascading. By using this approach, the amount of computational time spends on the false windows.

From the detected facial images V^{face}, the eye region is localized in the proposed eye region localization phase.

VI. Eye Localization via Optimized CNN

The proposed Eye Region Localization phase includes two major phases: (a) feature extraction and (b) face region identification. Initially, from the identified faces using viola-jones model, the facial features like improved ASM (I-ASM) and LBP (texture features) are extracted.

A. I-ASM Features

In ASM local texture feature and shape models are used. The deformation begins with the models shape. The shape is designated as a "point landmark". Eq. (3) is obtained by rotating, scaling and translating and as shown in Eq. (4).

$$G = \left(g_1, o_1, \ldots . g_n, o_n\right)^H \tag{3}$$

$$g = H_{wg,wo,d,\upsilon}\left(G\right) \tag{4}$$

For ordered shapes, the average shape is formulated as in Eq. (5).

$$\overline{g} = \sum_{i=1}^{m} g_i$$

The differences of each shape are indicated as Eq. (6).

$$dg_i = g_i - \overline{g}$$

The covariance matrices are indicated as Eq. (7). Here, points to the count of histogram curves.

$$S = \frac{1}{m}\sum_{i=1}^{m} dg_i dg_i^T \tag{7}$$

The variable indicates major axes that provides divergence modes of all points of shape, and signifies the unit eigenvectors. The principal component analysis of provides eigen vectors and eigen values, and these eigen values forms the mode of variation. During the training process, the histogram curve can be modelled as the sum of the mean model and weighted modes of variation. As shown in Eq. (9), a new shape is generated using the mean of the weighted matrix sum, the mean shape of the eigenvector, and the weight vector.

$$g = \overline{g} + Eb \tag{8}$$

$$b = E^T(S - \overline{S}) \tag{9}$$

The initial deformation weight is set as zero.

Find a new preferred curve shape: the pixel in that is nearest to the corresponding pixel on the histogram curve of the test image is a new preferred choice of landmark.

Compute new values for the weight changes: $\Delta g = E.\Delta b$ Here, $\Delta g = V^{face} - V^{F}$. Moreover, we acquire $b_{new} = b_{old} + \Delta b$

Limit the formulation: b is limited with the intention of guarantying the additional deformation. The weight can be chosen such that Euclidean distance is less than maximal distance.

Calculate the new shape

Repeat the process until the convergence is reached.

The extracted I-ASM feature is denoted as f^{I-ASM}

B. LBP

LBP is a measure of the correlation between pixels in a local region, which mostly represents local data. Many studies and research been published that combine more global or more local data with LBP in order to provide a more discriminative description from various feature levels. The LBP is less difficult to programme and has a higher discriminative potential. The image pixel, including the decimal numbers, is also labelled by the LBP operator. Each image pixel is computed with its surrounds during the labelling procedure by subtracting the value of the centre pixel. Furthermore, negative values are encoded as 0, and positive and zero values are encoded as 1. All of the binary codes are concatenated clockwise from the top-left to form a binary number, and these binary numbers are referred to as LBP codes [18]. The texture descriptor is used to create the local descriptions, which are then combined to create the global description. Furthermore, the distinguishable capability is used to extract characteristics from these texture objects. S_{pl} and S_{cl} in Eq. (10), respectively, denote the centre pixel intensities and the image centre pixel from the adjacent pl . The pixel's LBP descriptor is written as $LBP(\bullet)$, with NE_{Pl} denoting the number of neighbours. Eq. (11) gives the LBP descriptor function f^{LBP} .

$$LBP(S_{cl}) = \sum_{pl=0}^{NE_{pl}} FE_{LBP(pl,cl)} 2^{pl-1} \tag{10}$$

$$f^{LBP} = \begin{Bmatrix} 1, if\ S_{pl} - S_{cl} \geq 0 \\ 0, otherwise \end{Bmatrix} \tag{11}$$

The final detection takes place in CNN< which is trained with the extracted features $F = f^{LBP} + f^{I-ASM}$

C. CNN

The CNN is a deep learning model with three layers: "fully-connected layers, convolutional layer, and pooling layer". A lot of convolution kernels together makeup the convolution layer, which assist in computing the feature maps. A significant number of convolution kernels are included in the convolution layer, which aid in the calculation of various feature maps. The surrounding neurons of the previous layer are linked to each of the neurons in the feature map [19]. Finally, several kernels are employed in order to acquire the whole feature maps. At location (d,e) of the B^{th} feature map residing in the A^{th} layer is $Z^{A}_{d,e,B}$, which is computed as per Eq. (I2). In this same layer and location, the nonlinear activation function $act(\bullet)$ is computed as per Eq. (13). “Sigmoid, tanh, and ReLU” are the activation functions.

$$Z^{L}_{d,e,B} = W_B^{A^T} Q^{A}_{d,e} + bias^{A}_{B} \tag{12}$$

$$act^{A}_{d,e,B} = act\left(Z^{A}_{d,e,B}\right)$$

W and the $bias$ term denotes the weight and bias of CNN, respectively. This weight function is fine-tuned using SOEE model [20]. The patched input is denoted as $Q^{A}_{d,e}$. Shift-invariance has been achieved successfully by reducing feature map resolutions in the pooling layer. "The pooling layers of the CNN execute down sampling operations utilising the results collected from the convolutional layers." The pooling function is referred to as $pol()$.

The pooling function $M^{A}_{d,e,B}$ is then calculated for $act^{A}_{:,:,B}$. Here, $\Re_{d,e}$ denotes the local area in (d,e) . At the conclusion of the convolutional and pooling layers, there are multiple fully-connected layers.

$$M^{L}_{d,e,B} = pol\left(act^{L}_{d,e,B}\right), \forall (m.n) \in \Re_{d,e} \tag{14}$$

The loss() function is used to compute CNN's loss, which is represented by Eq.(15). This $Loss()$ function should be as small as possible, which is the goal of the current study. The goal function can be expressed formally as Eq. (16). A new SEEO model is used to fine-tune CNN's weight in order to achieve this objective.

$$Loss = \frac{1}{N} \sum_{n=1}^{N} l\left(\theta; M^{(n)}, O^{(n)}\right) \tag{15}$$

$$Obj = Min(Loss) \tag{16}$$

The total number of input-output relations $\left\{\left(J^{(n)}, M^{(n)}\right);\ n \in [1, \cdots, N]\right\}$ is indicated by N , and the CNN's overall parameter is denoted by θ . In addition, $J^{(n)}$ represents the n^{th} input data, $M^{(n)}$ represents the matching target labels, and $O^{(n)}$ represents the CNN output.

D. Seagull Optimization with Enhanced Exploration (SOEE)

The SOA model is a novel bio-inspired algorithm that is based on the migration and attacking behaviours of a seagull.

In general, the SOA model is good in solving the complex optimization model with lower computational complexity. In literature, it has been said that the convergence speed of the solutions increases, when the adaptive meta-heuristic operators are utilized. Therefore, the Seagull Optimization with Enhanced Exploration (SOEE) is introduced in this research work. The input to SOEE is the weight of the CNN, which is fine-tuned by the SOEE to enhance the eyeblink detection accuracy. The solution fed as input to SOEE is the weight of CNN, which is shown in Fig. 3.

The steps followed in the SOEE model is furnished in the upcoming section:

Step 1: Initialize the N search agent's population P.

Step 2: Initialize the parameters A, B, itr, Max^{itr} . Here, itr, Max^{itr} points to the current iteration and the maximal iteration, respectively.

Step 3: Set the value of $f_c \leftarrow 2; u \leftarrow 1; v \leftarrow 2$

Step 4: While $itr < Max^{itr}$ do

Step 5: Compute the fitness of the search agent using Eq. (16).

Step 6: Move to the migration behaviour of SOA model. Our contribution resides in this phase. In fact, three different processes take place in the Migration (exploration) phase: Collision avoidance, Movement along the direction of the best neighbor and stay closer with the best search agent.

- With the intention of avoiding the collision amongst the search agent, a new variable A referred as the movement behaviour of search agent is added.

$$C = A * P(X) \tag{17}$$

Here, C is the search agent's non- colliding position and P is the current position of the search agent. In addition, A can be computed as per Eq. (), wherein f_c is a variable that has been introduced to control the frequency of the employing A.

$$A = f_c - \left\{ X, \left[\frac{f_c}{Max^{itr}} \right] \right\} \tag{18}$$

- During the movement along the best neighbor's direction stage, our contribution has been included. Once the collision between the search agents is avoided, they are moved along the direction of the best neighbour. To make this movement more precise and to avoid the solutions from getting trapped into the local optima, a new mathematical model withy levy function $Levy(\beta)$ has been introduced in this research work. The newly developed mathematical expression is shown in Eq. (19).

$$M = B * \left[P_{best}(X) - P(X) \right] + Levy(\beta) \tag{19}$$

Here, $Levy(\beta)$ is the levy function undergone by the search agents while searching the global best solutions. In addition, $P_{best}(X)$, $P(X)$ and M points to the global best position of the solution, current position of the search agent, position of the search agent, respectively. With the intention of having a proposed balance between the exploration as well as exploitation phase, the variable B is computed as per Eq. (20). Our contribution resides in this computational phase also. In the existing SOA model, the random variable S has been generated within the limits $[0,1]$. But, here we've computed the value of S using the newly proposed expression given in Eq. (21).

$$B = 2 * A^2 * S \tag{20}$$

$$S = 1.07(7.9S_i - 23.3S_i^2 + 28.7S_i^3 - 13S_i^4) \tag{21}$$

- Remain close to the best search agent: finally the position of the search agent is updated with respect to the best search agent as per Eq. (22) of SOA model.

$$\vec{D} = \left| \vec{C} + \vec{M} \right| \tag{22}$$

In addition, $\vec{D}$ is the distance in between the best search agent and the current search agent.

Step 7: Move to the Attacking (exploitation) phase: in the attacking phase, the search agents undergo a spiral movement in the air. This spiral movement can be modelled in the three axes X, Y, Z as per Eq. (23)- Eq. (25), respectively.

$$x' = r * Cos(K) \tag{23}$$

$$y' = r * Sin(K) \tag{24}$$

$$z' = r * K \tag{25}$$

Step 8: The random variable K is modelled using the newly proposed expression given in Eq. (26). The proposed expression is based upon the singer map.

$$M = 1.07(7.9M_i - 23.3M_i^2 + 28.7M_i^3 - 13M_i^4) \tag{26}$$

Step 9: The updated position of the search agent is modelled as per Eq. (27). And eye region is located on that basis as shown in Fig. 4.

$$P(X) = (D * x' * y' * z') + P_{best}(x) \tag{27}$$

Step 10: Return $P(X)$

Step 11: Terminate

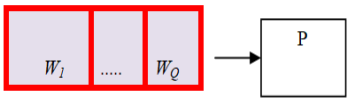
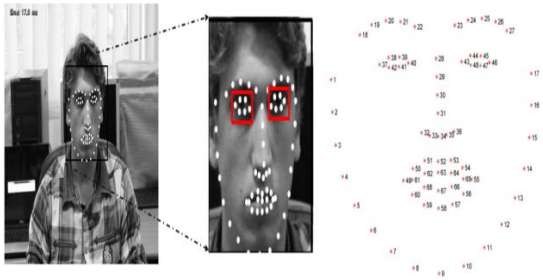


Fig. 3. Solution Encoding



(a) (b)

Fig. 4. An Illustration of Eye Landmark Detection: (a) Landmark Detection and (b) 68 Located Points.

VII. Eye Status Detection

Once, the eye region is localized, its status (open/ close) is identified by computing the eye aspect ratio. In this research work, the eye aspect ratio is computed between the eye points using Eq. (28). The eye computation for open and close eye is shown in Fig. 5.

$$EAR = \frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2.\|P_1 - P_4\|} \quad (28)$$

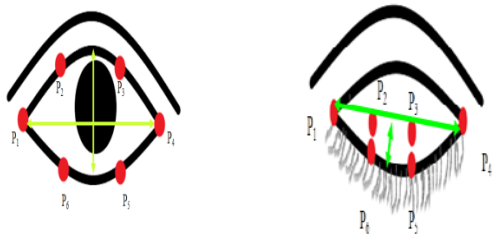
Here, P1,P2 ,P3 ,P4 ,P5 and P6 are the 2D landmark locations.

a) The eye is considered to be open if the Eye Aspect Ratio is greater than the threshold [21].

b) If the Eye Aspect Ratio is less than the threshold, the eye is considered closed.

c) The EAR of both eyes is averaged since eye blinking is produced simultaneously by both eyes.

A poor monetary worth the presence of an EAR value does not imply that a person is blinking. It can happen when a person closes his or her eyes for a prolonged period of time or makes a facial expression, such as yawning, or when the EAR records a brief random fluctuation of the landmarks.



(a) (b)

Fig. 5. EAR Computation of (a) Open and (b) Close Eye.

VIII. Eye Blink Detection and Classification

A. Eye Blink Detection

Once the eye's state has been determined, it is monitored for eye blink detection. For eye-blink detection, the normalized cross correlation technique is used in this study. The normalized cross correlation can be calculated using Eq. (29). In general, the correlation coefficient is used to assess the similarity of the present eye picture to the preserved template of the opened eye. As a result, the correlation coefficient is generally referred regarded as a measure of eye openness [22]. The template image matching to the user's eye is automatically obtained during the system's startup.

$$R(X',Y') = \frac{\sum_{X',Y'}[T'(X',Y').I(X+X',,Y+Y')]}{\sqrt{\sum_{X',Y'}\left[T'(X',Y')^2.\sum_{X',Y} I(X+X',,Y+Y')^2\right]}} \quad (29)$$

Here, R points to the correlation coefficient of the image and T is the template image. In addition, the original image is I ; and the co-ordinates are X, Y .

B. Eye Blink Classification

To identify deliberate eye-blinks with a length more than 250 milliseconds, the correlation coefficient is shifted over time. The on-set of the eye-blink is identified whenever the value of the coefficient is less than the predetermined threshold level of TL (2ms) for consecutive frames. If indeed the correlation coefficient value is greater than the threshold value TH, the off-set of the eye-blink is discovered (250ms). Experimentation was used to determine the TL and TH threshold values. Whenever a detected eye-blink lasts more than 250 milliseconds but less than 2 milliseconds, it is considered a "control" blink [23].

IX. Result and Discussion

A. Experimental Setup

The proposed work has been implemented in PYTHON. The dataset for evaluation has been collected from [32]. The collected sample images are shown in Fig. 6. And open eye sample image are shown in Fig. 7. The assessment has been carried out in terms of MAE, MAPE, MSE, MSLE, RMSE and Optimization error as well. the proposed work (CNN+SOEE) has been compared over the existing models like the CNN+Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN, respectively. The correctly detected eye-blinks are denoted as True Positives (TP), false detections are denoted as False Positives (FP), and missed eye-blinks are denoted as False Negatives (FN) [24].

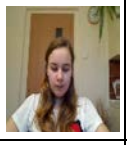
Close					
	sample image	Standard median filtering	proposed KMF	Viola-jones face detection model	I-ASM

Fig. 6. Sample Image for Close Eye based Eye-Blink Detection.

Open					
	sample image	Standard median filtering	proposed KMF	Viola-jones face detection model	I-ASM

Fig. 7. Sample Image for Open Eye based Eye-Blink Detection.

B. Convergence Analysis

In literature, it has been portrayed that the convergence of the solutions increases with adaptive parameter tunings in the standard optimization model. However, it is highly significant to have a mathematical evaluation, in order to show that the newly introduced SOEE model is highly convergent over the existing ones. Therefore, a convergence analysis has been undergone for the proposed work by varying the count of iterations. Since, the prime objective behind this research work is to minimize the loss function of CNN, which localizes the eyes accurately. Once, the eyes are localized with higher accuracy then the status of the eyes can be detected more precisely [25]. Since, the objective function is a minimization function, the approach that achieves the least cost function is said to be the best approach, as it has converged sooner. The resultant of the convergence analysis of the projected model is manifested in Fig. 8.

On observing the acquired outcomes, the proposed work is identified to be successful even at the highest iteration counts. This clearly says that the projected model is applicable for huge databases. Initially, at the 0th iteration count, the cost function of the proposed as well as existing model is found to be higher. And then as the count of iterations got expanded, there seems to minimization in the cost function [26]. At the 6th iteration count, the projected model has recorded the least cost function than the existing models. Most interestingly, the proposed work has recorded the most least cost function as 4.6 at the 25th iteration count, and at this count the cost function recorded by the existing models are identified to be greater than 4.9. As a whole, the projected model is said to be highly convergent, and hence solved the optimization problems effectively.

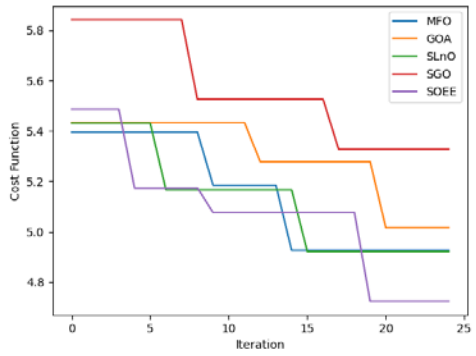


Fig. 8. Convergence Analysis of SEEO Model.

C. MAE Analysis

The MAE is the most commonly utilized for forecasting the accuracy of the model. In general, the Mean absolute error is the measure of the average absolute difference between the actual and predicted values in the eyeblink dataset. Mathematically, the MAE is given as per Eq. (30).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left| Y_i - \hat{Y}_i \right| \quad (30)$$

Here, Y_i is the actual outcome and $\hat{Y}_i$ is the predicted outcome.

In this research work, the MAE of the proposed work has been computed, and its results acquire are compared over the existing models like the CNN+Bi-LSTM [27], C-LSTM [3],Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN, respectively. This evaluation has been carried out by varying the learning percentage from 60, 70, 80 and 90, respectively. The results acquired are shown in Fig. 9. On observing the outcomes, the projected model has recorded the least MAE value for every variation in the learning percentage. Among the recorded results, the proposed work has recorded the least MAE value as 1.2 at 90th learning percentage. The Mae of the projected work recorded at the 60th learning percentage is 64.2%, 78.5%, 62.5%, 81.2%, 75%, 83.3%, 50% and 57.14% improved over the existing models like the CNN+Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN, respectively. As a whole, the projected model is said to be less prone to errors, and this is owing to the fine-tuning of the weight function of the CNN using the newly projected SOEE model. Since, the prime objective behind this research work is minimization of the loss function of the CNN, the minimized MAE acquired with the projected work has said that the defined objective has been successfully satisfied by the projected model [28].

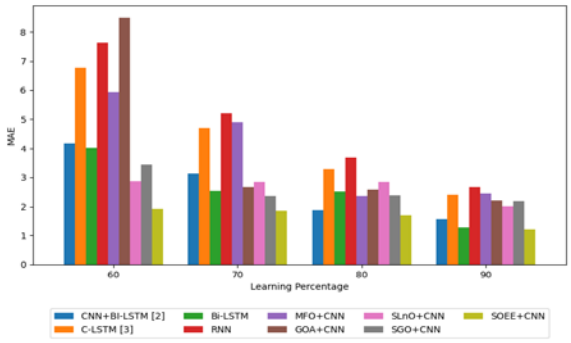


Fig. 9. MAE Analysis of SEEO Model.

D. MAPE Analysis

The MAPE also known as the mean absolute percentage deviation (MAPD) has been utilized for measuring the detection accuracy. Mathematically, the MAPE can be given as per Eq. (31).

$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{Y_i - \hat{Y_i}}{Y_i}\right| \quad (31)$$

The MAPE of the proposed work has been computed, and the results acquired are manifested in Fig. 10. This evaluation has been undergone by varying the learning percentage. The graphical results acquired have exhibited a reduced MAPE value for the projected work. The projected model had recorded the minimal MAPE values as 0.03, 0.035, 0.038 and 0.015at the 60th, 70th, 80th and 90th learning percentage. At the 60th learning percentage, the MAPE of the projected work is 37.5%, 64.2%, 34.2%, 68.75%, 69.5%, 80.7%, 41.8% and 51.9% improved over the existing models like the CNN+Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN, respectively.

E. MSE Analysis

MSE is the measure of the measure of the average of the squares of the errors (i.e. the averaged squared difference between the actual and the estimated value.

$$MSE = \frac{1}{N-M}\sum_{k=1}^{N-M}\left(Act - \Pr e\right)^2 \quad (32)$$

The MSE of the proposed work is the lowest one for every variation in the learning percentage. The MSE of the projected work is found be below 8% for every variation in the learning percentage. The results acquired are shown in Fig. 11. The MSE value recorded by the existing works is above 15%. The proposed work has recorded the least MSE value at the 90th learning percentage. Moreover, all the existing models has recorded the MSE value as 50, 140, 42, 170, 70, 162, 19, 22 and 9 for CNN+Bi-LSTM [2], C-LSTM [3],Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN and proposed work, respectively. From the acquired values, it is clear that the projected model has recorded the least MSE value. On observing the overall evaluation, it is clear that the presented work is less prone to errors [29].

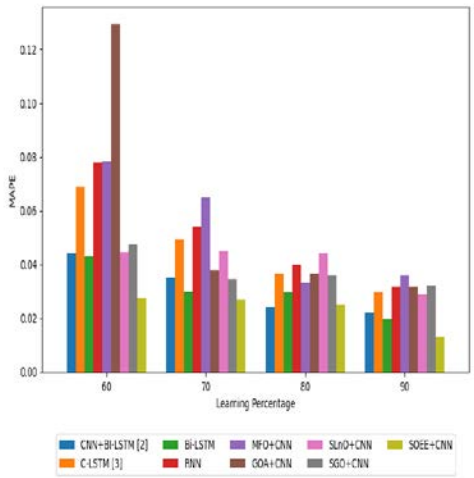


Fig. 10. MAPE Analysis of SEEO Model.

F. MSLE Analysis

The Mean squared logarithmic error (MSLE) of the projected model has been shown in Fig. 12. The MSLE is the ratio of the measure between the actual and the predicted outcomes.

$$MSLE = \frac{1}{N}\sum_{i=1}^{N}\left((\log(Y_i + 1) - (\log(\hat{Y_i} + 1)\right)^2 \quad (33)$$

The results acquired are shown in Fig. 12. On observing the outcomes, the projected model had exhibited lower MSLE values. The major reason behind this reduction owes towards the fine-tuning the weight function using the newly proposed optimization model [30].

G. RMSE Analysis

"The RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed". RMSE can be computed mathematically per Eq. (34).

$$RMSE = \sqrt{\frac{1}{N-M}\sum_{k=1}^{N-M}\left(Act - \Pr e\right)^2} \quad (34)$$

The RMSE performance is evaluated for both the proposed and the extant techniques by varying the learning percentage. The results acquired are shown in Fig.13. The proposed work has recorded the least value for every variation in the learning percentage. The RMSE of the projected model at 60th learning percentage is 71.4%, 83.3%, 70.2%, 85.7%, 77.7%, 81.8%, 50% and 53.4% improved over the existing models like the CNN+Bi-LSTM [2], C-LSTM [3],Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO+CNN and SOA+CNN, respectively [31].

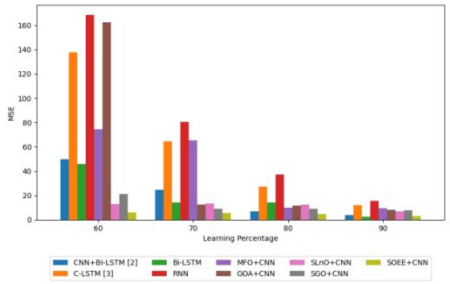


Fig. 11. MSE Analysis of SEEO Model.

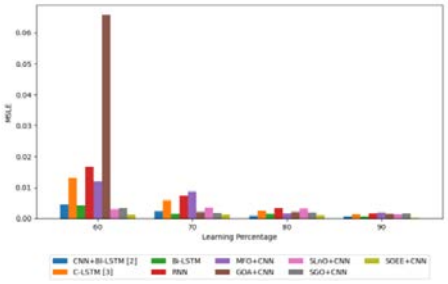


Fig. 12. MSLE Analysis of SEEO Model.

H. Performance Analysis of Proposed Work: With vs Without Optimization

The optimization algorithm deployed in this research work for eye landmark detection is said to play a major role in enhancing the detection accuracy of the eye blinks. It is essential to exhibit the supremacy of the optimization algorithms. Therefore, the proposed model has been validated with and without optimization logic. The results acquired are shown in Table II. The proposed work (KMF + I-ASM) with ESSO is compared over the existing models like Conventioanl Model (standard median filtering + Standard ASM) + ESSO and Conventioanl Model (standard median filtering +Standard ASM). The assessment has been carried out in terms of MAE, MAPE, MSE, MSLE, RMSE, respectively. The MAE of the projected model is 5.240172259, which is the least value when compared with the existing models. The MAE of the projected model is 16.3% and 15.5% better than the MAE of the conventional model + optimization and conventional model, respectively.

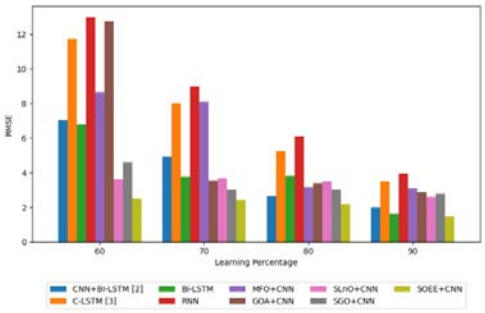


Fig. 13. RMSE Analysis of SEEO Model.

TABLE II. OVERALL PERFORMANCE ANALYSIS

	proposed work (KMF+I-ASM) with ESSO	Conventioanl Model (standard median filtering+Standard ASM)+ESSO	Conventioanl Model (standard median filtering+Standard ASM)
MSE	5.240172	6.261141	6.204798
MAE	1.754693	1.890539	1.909948
MAPE	0.025316	0.02734	0.027585
MSLE	0.001008	0.001201	0.00117
RMSE	2.289142	2.502227	2.490943

The MAPE of the projected model is 7.1% and 8.12% improved over conventional model + optimization and conventional model, respectively. In addition, the MSE of the projected model is 0.025316349, which is the least value when compared to conventional model + optimization = 0.027340005 and conventional model = 0.027584814. In addition, the projected model with optimization has recorded the least MSLE and RMSE also. Thus, from the overall evaluation, it's obvious that the major improvement behind the projected work in precisely detecting the eye blinks is only due to the fine-tuning of the weight of CNN via the optimization model [32].

I. Performance Analysis

The performance of the projected model has been validated in terms of MSE, MAE, MAPE, MSLE and RMSE, respectively. On observing the outcomes, the projected work has recorded the least error measures. The results acquired are shown in Table II. The MSE of the projected work is 5.71662, which is indeed the least6 value when compared to CNN + Bi-LSTM [2] = 24.57614, C-LSTM [3] = 64.39177, Bi-LSTM = 14.25622, RNN = 80.51956, MFO + CNN = 65.42977, GOA+CNN = 12.47136, SLnO+CNN = 13.40547 and SOA+CNN = 8.938489. In addition, an improvement of 54.1%, 71.6%, 52.6%, 74.8%, 67.6%, 77.3%, 33.1% and 44.19% has been recorded by the projected model over the existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO+CNN, GOA + CNN, SLnO + CNN and SOA+CNN, respectively in terms of MAPE. The MSE of the proposed work is 37.9%, 59.9%, 35.7%, 64.7%, 64.8%, 78.7%, 38.2% and 42.2% better than the MSE value recorded by existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO + CNN, GOA+CNN, SLnO + CNN and SOA+CNN, respectively. The MSLE of the proposed work is 74.09%, 91.1%, 71.8%, 93.04%, 90.2%, 98.2%, 62.6% and 65.5% better than the MSE value recorded by existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO+CNN, GOA+CNN, SLnO + CNN and SOA + CNN, respectively. Moreover, the RMSE of the projected model is also lower. The major reason behind this reduction in the error measures is owing to the utilization of the appropriate techniques for blink detection. Moreover, the prime objective of this research work is minimization of the error measures, which has been found to be achieved [33].

J. Overall Performance Analysis

The overall performance of the projected model is validated in terms of MSE, MAE, MAPE, MSLE and RMSE, respectively. The results acquired are shown in Table III. On observing the outcomes, the projected work has recorded the least error measures. The MSE value recorded by the projected model is 5.71662, which is 76.7%, 91.1%, 59.9%, 91.9%, 91.2%, 54.1%57.3% and 36.04% better than the MSE value recorded by existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO + CNN, GOA + CNN, SLnO + CNN and SOA + CNN, respectively. In addition, the projected model has recorded the least MAPE as 1.847275453, MSE as 0.026834139, MSLE as 0.001152411 and RMSE as 2.390945437. The major reason behind this improvement is owing to the utilization of the optimized deep learning model for eye landmark detection [34].

K. Analysis on Optimization Error

The optimization error measures recorded by the projected as well as existing models for 60, 70, 80 and 90 percent of learning are manifested in Table IV. At 60th learning percentage, the MSE value recorded by the projected model is 6.235624262, which is the least value than the existing models [35]. The MSE of the projected work at 60th learning percentage is 87.4%, 95.4%, 86.3%, 96.2%, 91.65%, 96.16%, 51.7% and 70.4% better than the MSE value recorded by existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO + CNN, GOA + CNN, SLnO + CNN and SOA+CNN, respectively.

TABLE III. OVERALL PERFORMANCE ANALYSIS OF THE PROPOSED WORK

	CNN+BI-LSTM [2]	C-LSTM [3]	Bi-LSTM	RNN	MFO+CNN	GOA+CNN	SLnO+CNN	CNN+SGO	CNN+ISGO
MSE	24.57614	64.39177	14.25622	80.51956	65.42977	12.47136	13.40547	8.938489	5.71662
MAE	3.136789	4.703131	2.530748	5.206161	4.894731	2.660722	2.838228	2.351218	1.847275
MAPE	0.035335	0.049186	0.03004	0.053776	0.064812	0.037911	0.044808	0.034593	0.026834
MSLE	0.002347	0.005867	0.001501	0.007401	0.008796	0.002181	0.003551	0.001795	0.001152
RMSE	4.957433	8.024448	3.775741	8.973269	8.088867	3.531481	3.661348	2.989731	2.390945

TABLE IV. ANALYSIS ON OPTIMIZATION ERROR

Learning Percentage		CNN+BI-LSTM [2]	C-LSTM [3]	Bi-LSTM	RNN	MFO+CNN	GOA+CNN	SLnO+CNN	CNN+SGO	CNN+ISGO
60	MSE	49.63121	137.5381	45.80966	168.2764	74.64012	162.3975	12.9309	21.13644	6.235624
60	MAE	4.18496	6.766904	4.015798	7.63049	5.931651	8.493981	2.871482	3.441972	1.920842
60	MAPE	0.04439	0.068813	0.042835	0.078037	0.078374	0.129324	0.044625	0.047733	0.027543
60	MSLE	0.004503	0.013231	0.004145	0.016769	0.011923	0.065666	0.003121	0.003383	0.001167
60	RMSE	7.044942	11.72767	6.768283	12.97214	8.639451	12.74353	3.595957	4.597439	2.497123
70	MSE	24.57614	64.39177	14.25622	80.51956	65.42977	12.47136	13.40547	8.938489	5.71662
70	MAE	3.136789	4.703131	2.530748	5.206161	4.894731	2.660722	2.838228	2.351218	1.847275
70	MAPE	0.035335	0.049186	0.03004	0.053776	0.064812	0.037911	0.044808	0.034593	0.026834
70	MSLE	0.002347	0.005867	0.001501	0.007401	0.008796	0.002181	0.003551	0.001795	0.001152
70	RMSE	4.957433	8.024448	3.775741	8.973269	8.088867	3.531481	3.661348	2.989731	2.390945
80	MSE	7.021394	27.4651	14.35815	37.27164	10.08974	11.63154	12.38245	8.942254	4.812689
80	MAE	1.883875	3.272843	2.51215	3.682338	2.363624	2.586079	2.847396	2.376085	1.688624
80	MAPE	0.024333	0.036495	0.029784	0.040014	0.033364	0.03649	0.044139	0.03589	0.024871
80	MSLE	0.00092	0.002566	0.001488	0.003387	0.001733	0.002148	0.003147	0.002065	0.001023
80	RMSE	2.649791	5.240716	3.789215	6.105051	3.176434	3.410505	3.518871	2.99036	2.193784
90	MSE	4.077275	12.19301	2.714369	15.68119	9.575808	8.282909	6.852754	7.944575	3.126543
90	MAE	1.551431	2.413057	1.276575	2.658186	2.45427	2.206566	2.005485	2.176067	1.192031
90	MAPE	0.022036	0.029499	0.019703	0.031625	0.035901	0.031573	0.028835	0.032025	0.012821
90	MSLE	0.000757	0.001373	0.00066	0.001648	0.001956	0.001578	0.001337	0.001672	0.000127

TABLE V. ANALYSIS ON OPTIMIZATION METRICS

Learning Percentage		CNN+BI-LSTM [2]	C-LSTM [3]	Bi-LSTM	RNN	MFO+CNN	GOA+CNN	SLnO+CNN	CNN+SGO	CNN+ISGO
60	Accuracy	0.88	0.865	0.865	0.855	0.88	0.895	0.9025	0.88	0.92
60	Precision	0.88	0.865	0.865	0.855	0.88	0.895	0.9025	0.88	0.92
60	F-Measure	0.88	0.865	0.865	0.855	0.88	0.895	0.9025	0.88	0.92
70	Accuracy	0.856667	0.836667	0.846667	0.853333	0.81	0.846667	0.85	0.706667	0.89

70	Precision	0.856667	0.836667	0.846667	0.853333	0.81	0.846667	0.85	0.706667	0.89
70	F-Measure	0.856667	0.836667	0.846667	0.853333	0.81	0.846667	0.85	0.706667	0.89
80	Accuracy	0.705	0.75	0.675	0.69	0.695	0.7	0.715	0.63	0.765
80	Precision	0.705	0.75	0.675	0.69	0.695	0.7	0.715	0.63	0.765
80	F-Measure	0.705	0.75	0.675	0.69	0.695	0.7	0.715	0.63	0.765
90	Accuracy	0.75	0.84	0.94	0.92	0.88	0.81	0.87	0.77	0.85
90	Precision	0.75	0.84	0.94	0.92	0.88	0.81	0.87	0.77	0.85
90	F-Measure	0.75	0.84	0.94	0.92	0.88	0.81	0.87	0.77	0.85

L. Analysis on Optimization Metrics

The performance of the projected work is evaluated in terms of Accuracy, Precision and F-Measure, respectively. This evaluation has been carried out for varying learning percentage. The results acquired are shown in Table IV. The proposed work has attained the highest accuracy as 92% in case of eye blink detection at 60th learning percentage. The accuracy of the projected work at 60th learning percentage is 4.3%, 5.97%, 5.97%, 7.06%, 4.3%, 2.7%, 1.9% and 4.3% improved over the existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO + CNN, GOA + CNN, SLnO + CNN and SOA + CNN, respectively. In addition, at the 70th learning percentage [36-39], the projected model has recorded 89% of accuracy in eye blink detection. Moreover, at 90th learning percentage, the proposed work has recorded accuracy as 85%, which is found to be 11.7%, 1.1%, 10.5%, 8.23%, 3.5%, 4.7%, 2.3% and 9.4% improved over existing models like CNN + Bi-LSTM [2], C-LSTM [3], Bi-LSTM, RNN, MFO + CNN, GOA + CNN, SLnO + CNN and SOA+CNN, respectively.

X. Conclusion

In this research work, a novel eye-blink detection model is developed. The proposed eye blink detection model is modeled by following seven major phases: (a) video –to-frame conversion, (b) Pre-processing, (c) face detection, (d) eye region localization, (e) eye landmark detection and eye status detection, (f)eye blink detection and (f) Eye blink Classification. Initially, from the collected raw video sequence (input), each individual frames are extracted in the video –to-frame conversion phase. Then, each of the frames are subjected to pre-processing phase, where the quality of the image in the frames are improved using proposed Kernel median filtering (KMF) approach. In the face detection phase, the Viola-Jones Model has been utilized. Then, from the detected faces, the eye region is localization within the proposed eye region localization phase. The proposed Eye region localization phase encapsulates two major phases: Feature extraction and landmark detection. The features like improved active shape models (I-ASMs), LBP are extracted from the detected facial images. Then, the eye region is localization by using a new optimized CNN framework. This optimized CNN framework is trained with the extracted features (I-ASM and LBP). Moreover, to enhance the classification accuracy of eye localization, the weight of CNN is fine-tuned using a new Seagull Optimization with Enhanced Exploration (SOEE), which is the improved version of standard Seagull Optimization Algorithm (SOA) [31]. The outcome from optimized CNN framework is providing the exact location of the eye region. Once the eye region is detected, it is essential to detect the status of the eye (whether open or close). The status of the eye is detected by computing the eye aspect ratio (EAR). Then, the identified eye blinks are classified based on the computed correlation coefficient as long and short blinks. Finally, a comparative evaluation has been accomplished to validate the projected model.

XI. Conflicts of Interest

There are no conflicts of interest declared by the authors.

References

[1] T.SreeSharmila,RSrinivasan, K K.Nagarajan, S.Athithya, "Eye Blink Detection Using Back Ground Subtraction and Gradient-Based Corner Detection for Preventing CVS", Procedia Computer Science, 2019.

[2] Mani V, Manickam P, Alotaibi Y, Alghamdi S, Khalaf OI. Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records. Electronics. 2021; 10(23):3003. https://doi.org/10.3390/electronics10233003.

[3] Rajamohana,S.P.Radhika,E.G.Priya,S.SangeethaS.,"Driver drowsiness detection system using hybrid approach of convolutional neural network and bidirectional long short term memory (CNN_BILSTM)", Materials Today: Proceedings, 2021.

[4] AzharQuddus, AliShahidiZandib, LauraPrest, Felix J.E.Comeau, "Using long short term memory and convolutional neural networks for driver drowsiness detection", Accident Analysis & Prevention, 2021.

[5] ZhongminLiu, YuxiPeng, WenjinHu, "Driver fatigue detection based on deeply-learned facial expression representation", Journal of Visual Communication and Image Representation, 2002.

[6] Rout, R., Parida, P., Alotaibi, Y., Alghamdi, S., & Khalaf, O. I. (2021). Skin Lesion Extraction Using Multiscale Morphological Local Variance Reconstruction Based Watershed Transform and Fast Fuzzy C-Means Clustering. Symmetry, 13(11), 2085.

[7] YanWang, RuiHuang, LeiGuo, "Eye gaze pattern analysis for fatigue detection based on GP-BCNN with ESM", Pattern Recognition Letters, 2019.

[8] D. Selvathi, "FPGA Based Human Fatigue and Drowsiness Detection System Using Deep Neural Network for Vehicle Drivers in Road Accident Avoidance System", Human Behaviour Analysis Using Intelligent Systems, 2019.

[9] Puneet Singh Lamba, Deepali Virmani& Oscar Castillo, "Multimodal human eye blink recognition method using feature level fusion for exigency detection", Soft Computing, 2020.

[10] Alotaibi, Y., Malik, M. N., Khan, H. H., Batool, A., ul Islam, Saif, Alsufyani, A., & Alghamdi, S. (2021). Suggestion Mining from Opinionated Text of Big Social Media Data. CMC-COMPUTERS MATERIALS & CONTINUA, 68(3), 3323-3338.

[11] RaimondasZemblys, Diederick C. Niehorster, Kenneth Holmqvist,"gazeNet: End-to-end eye-movement event detection with deep neural networks", Behavior Research Methods, 2019.

[12] M. Shahbakhti *et al*., "Simultaneous Eye Blink Characterization and Elimination from Low-Channel Prefrontal EEG Signals Enhances Driver Drowsiness Detection," in *IEEE Journal of Biomedical and Health Informatics*. doi: 10.1109/JBHI.2021.3096984.

[13] J. Wang, J. Cao, D. Hu, T. Jiang and F. Gao, "Eye Blink Artifact Detection With Novel Optimized Multi-Dimensional Electroencephalogram Features," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1494-1503, 2021. doi: 10.1109/TNSRE.2021.3099232.

[14] M. Shahbakhti *et al*., "VME-DWT: An Efficient Algorithm for Detection and Elimination of Eye Blink From Short Segments of Single EEG Channel," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 408-417, 2021. doi: 10.1109/TNSRE.2021.3054733.

[15] Suryanarayana, G., Chandran, K., Khalaf, O. I., Alotaibi, Y., Alsufyani, A., & Alghamdi, S. A. (2021). Accurate Magnetic Resonance Image Super-Resolution Using Deep Networks and Gaussian Filtering in the Stationary Wavelet Domain. IEEE Access, 9, 71406-71417.

[16] P. Ren *et al*., "Dynamics of Blink and Non-Blink Cyclicity for Affective Assessment: A Case Study for Stress Identification," in *IEEE Transactions on Affective Computing*. doi: 10.1109/TAFFC.2019.2946829.

[17] F. Wadehn, T. Weber, D. J. Mack, T. Heldt and H. -A. Loeliger, "Model-Based Separation, Detection, and Classification of Eye Movements," in *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 2, pp. 588-600, Feb. 2020. doi: 10.1109/TBME.2019.2918986.

[18] T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," in *IEEE Access*, vol. 8, pp. 83144-83154, 2020.doi: 10.1109/ACCESS.2020.2988660.

[19] Jha, N., Prashar, D., Khalaf, O. I., Alotaibi, Y., Alsufyani, A., & Alghamdi, S. (2021). Blockchain Based Crop Insurance: A Decentralized Insurance System for Modernization of Indian Farmers. Sustainability, 13(16), 8921.

[20] J. Cao *et al*., "Unsupervised Eye Blink Artifact Detection From EEG With Gaussian Mixture Model," in *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2895-2905, Aug. 2021. doi: 10.1109/JBHI.2021.3057891.

[21] Alotaibi, Y. (2020). A New Secured E-Government Efficiency Model for Sustainable Services Provision. Journal of Information Security and Cybercrimes Research, 3(1), 75-96.

[22] A. A. Jordan, A. Pegatoquet, A. Castagnetti, J. Raybaut and P. Le Coz, "Deep Learning for Eye Blink Detection Implemented at the Edge," in *IEEE Embedded Systems Letters*. doi: 10.1109/LES.2020.3029313.

[23] Z. Wang, J. Chai and S. Xia, "Realtime and Accurate 3D Eye Gaze Capture with CNN-Based Iris and Pupil Segmentation," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 190-203, 1 Jan. 2021. doi: 10.1109/TVCG.2019.2938165.

[24] X. Tian *et al*., "iBlink: A Wearable Device Facilitating Facial Paralysis Patients to Blink," in *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1789-1801, 1 Aug. 2019. doi: 10.1109/TMC.2018.2868660.

[25] Bharany, S., Sharma, S., Badotra, S., Khalaf, O. I., Alotaibi, Y., Alghamdi, S., & Alassery, F. (2021). Energy-Efficient Clustering Scheme for Flying Ad-Hoc Networks Using an Optimized LEACH Protocol. Energies, 14(19), 6016.

[26] H. K. Wong, J. Epps and S. Chen, "Automatic Pupillary Light Reflex Detection in Eyewear Computing," in *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 560-572, Dec. 2019. doi: 10.1109/TCDS.2018.2880664.

[27] C. -T. Lin *et al*., "EOG-Based Eye Movement Classification and Application on HCI Baseball Game," in *IEEE Access*, vol. 7, pp. 96166-96176, 2019. doi: 10.1109/ACCESS.2019.2927755.

[28] K. Li, Y. Gong and Z. Ren, "A Fatigue Driving Detection Algorithm Based on Facial Multi-Feature Fusion," in *IEEE Access*, vol. 8, pp. 101244-101259, 2020. doi: 10.1109/ACCESS.2020.2998363.

[29] Veeraiah, N., Khalaf, O. I., Prasad, C. V. P. R., Alotaibi, Y., Alsufyani, A., Alghamdi, S. A., & Alsufyani, N. (2021). Trust aware secure energy efficient hybrid protocol for manet. IEEE Access, 9, 120996-121005.

[30] C. Zhang, X. Wu, X. Zheng and S. Yu, "Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications," in *IEEE Access*, vol. 7, pp. 11829-11843, 2019. doi: 10.1109/ACCESS.2019.2891971.

[31] A. Sciarrone, I. Bisio, C. Garibotto, F. Lavagetto, G. H. Ssstaude and A. Knopp, "Leveraging IoT Wearable Technology Towards Early Diagnosis of Neurological Diseases," in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 582-592, Feb. 2021. doi: 10.1109/JSAC.2020.3021573.

[32] Z. M. Alhakeem, R. S. Ali and R. A. Abd-Alhameed, "Wheelchair Free Hands Navigation Using Robust DWT_AR Features Extraction Method With Muscle Brain Signals," in *IEEE Access*, vol. 8, pp. 64266-64277, 2020. doi: 10.1109/ACCESS.2020.2984538.

[33] V. Skaramagkas *et al*., "Review of eye tracking metrics involved in emotional and cognitive processes," in *IEEE Reviews in Biomedical Engineering*. doi: 10.1109/RBME.2021.3066072.

[34] Z. Wang, Q. Hong and X. Wang, "A Memristive Circuit Implementation of Eyes State Detection in Fatigue Driving Based on Biological Long Short-term Memory Rule," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. doi: 10.1109/TCBB.2020.2974944.

[35] Y. Zhou, S. He, Q. Huang and Y. Li, "A Hybrid Asynchronous Brain-Computer Interface Combining SSVEP and EOG Signals," in *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 10, pp. 2881-2892, Oct. 2020. doi: 10.1109/TBME.2020.2972747.

[36] W. Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features," in *IEEE Access*, vol. 7, pp. 118727-118738, 2019. doi: 10.1109/ACCESS.2019.2936663.

[37] M. Mahmood *et al*., "Soft Nanomembrane Sensors and Flexible Hybrid Bioelectronics for Wireless Quantification of Blepharospasm," in *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 11, pp. 3094-3100, Nov. 2020. doi: 10.1109/TBME.2020.2975773.

[38] Y. Xie *et al*., "Novel Wearable Sensors for Biomechanical Movement Monitoring Based on Electromagnetic Sensing Techniques," in *IEEE Sensors Journal*, vol. 20, no. 2, pp. 1019-1027, 15 Jan.15, 2020. doi: 10.1109/JSEN.2019.2943487.

[39] Gaurav Dhiman, VijayKumar,"Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems", Knowledge-Based Systems, Vol.165, February 2019.

[40] Dataset collected from :" https://www.blinkingmatters.com/research", Access Date: 2021-10-23.

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

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

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

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

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

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

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