ITC 3/52
Information Technology and Control
Vol. 52 / No. 3 / 2023
pp. 712-729
DOI 10.5755/j01.itc.52.3.33719

Face Positioned Driver Drowsiness Detection Using Multistage Adaptive 3D Convolutional Neural Network

Received 2023/03/27 | Accepted after revision 2023/06/05

HOW TO CITE: Adhithyaa, N., Tamilarasi, A., Sivabalaselvamani, D., Rahunathan, L. (2023). Face Positioned Driver Drowsiness Detection Using Multistage Adaptive 3D Convolutional Neural Network. *Information Technology and Control*, 52(3), 712-729. https://doi.org/10.5755/j01.itc.52.3.33719

Face Positioned Driver Drowsiness Detection Using Multistage Adaptive 3D Convolutional Neural Network

N. Adhithyaa

Department of Artificial Intelligence, Kongu Engineering College, Perundurai, Tamil Nadu, India - 638060 phone: +91 7845243520; e-mail: adhithyaa.research@gmail.com

A. Tamilarasi, D. Sivabalaselvamani, L. Rahunathan

Department of Computer Applications, Kongu Engineering College, Perundurai, Tamil Nadu, India - 638060

Corresponding author: nadhithyaaresea@outlook.com

Accidents due to driver drowsiness are observed to be increasing at an alarming rate across all countries and it becomes necessary to identify driver drowsiness to reduce accident rates. Researchers handled many machine learning and deep learning techniques especially many CNN variants created for drowsiness detection, but it is dangerous to use in real time, as the design fails due to high computational complexity, low evaluation accuracies and low reliability. In this article, we introduce a multistage adaptive 3D-CNN model with multi-expressive features for Driver Drowsiness Detection (DDD) with special attention to system complexity and performance. The proposed architecture is divided into five cascaded stages: (1) A three level Convolutional Neural Network (CNN) for driver face positioning (2) 3D-CNN based Spatio-Temporal (ST) Learning to extract 3D features from face positioned stacked samples. (3) State Understanding (SU) to train 3D-CNN based drowsiness models (4) Feature fusion using ST and SU stages (5) Drowsiness Detection stage. The Proposed system extract ST values from the face positioned images and then merges it with SU results from each state understanding sub models to create conditional driver facial features for final Drowsiness Detection (DD) model. Final DD Model is trained offline and implemented in online, results show the developed model performs well when compared to others and additionally capable of handling Indian conditions. This method is applied (Trained and Evaluated) using two different datasets, Kongu Engineering College Driver Drowsiness Detection (KEC-DDD) own dataset and National Tsing Hua University Driver Drowsiness Detection (NTHU-DDD) Benchmark Dataset. The proposed system trained with KEC-DDD dataset produces accuracy of 77.45% and 75.91% using evaluation set of KEC-DDD and NTHU-DDD dataset and capable to detect driver drowsiness from 256×256 resolution images at 39.6 fps at an average of 400 execution seconds.

KEYWORDS: Driver Drowsiness Detection, Convolution Neural Network, Face Positioning, Spatio Temporal Learning, 3D-CNN, KEC-DDD Dataset.

1. Introduction

Road accidents are dangerous to human community. According to National Highway Traffic Safety Administration report, USA, 24.5% accidents and Ministry of Road Transport and Highways report, India, 27% accidents are caused by driver fatigues. Risk of accident increases by four to five times in almost all countries [11]. Regular accidents especially in densely populated country like India heavily affects people safety. As a necessary consequence, research towards detecting driver drowsiness is significant. To prevent drowsiness, behavioural strategies including drinking tea or coffee, stopping for a little nap, and riding with a passenger are typically advised. These precautions, however, might not work if a motorist is not aware that he or she is tired [12]. Drowsiness is usually indicated by excessive yawning, bowing, or head sliding, as well as persistent blinking and there are variety of methods to measure the driver drowsiness level but when it comes to Indian conditions, most of the research work does not produce good results. For example, during yawning, Indians have the habit of placing their hands to cover the mouth as not to disturb others and this habit is reflected in most of the Indian drivers during driving, the works in conventional models does not consider this specific feature related to Indian conditions. In this work most of the drowsiness features are considered and a global driver drowsiness detection model using 3D Deep Convolution Neural network.

Driver Drowsiness is linked to psychological and physiological changes of driver such as blink rate, pulse rate, anxiety, and so on. Generally, these methods fall under four different categories. Image-based measures; vehicle-based measures; biological-based measures; hybrid-based measures [2]. In image-based measures, drowsiness symptoms can be seen and recorded using cameras or visual sensors. These measures are further divided for motion of lips, head movements and frequency of eyes closures [25]. Although Computer Vision based sleepiness detection techniques are most useful, their effectiveness is influenced by changes in lighting, facial expressions, and stance. However, with the advancement of deep learning, sleepiness detection methods based on convolutional neural networks (CNNs) are now a state-of-the-art method [23].

Deep belief network [26] is introduced to identify the facial landmarks with own dataset collected across different ages, genders and under various illumination conditions and 68 facial landmarks are identified. Using different deep learning architectures like ResNet50, VGG16, Inception V3 and VGG19, various varieties of DDD systems [8, 21-22] are introduced but when we combine methods to work sequentially to improve the performance, feature fusion issue arises, resulting in losing important facial elements [4]. To avoid feature fusion losses, a deep cascaded convolution neural network that identifies exact features of the facial regions is trained offline and blended for online monitoring [4], Spatial and Temporal space is explored for the facial regions after which bilinear feature fusion [3] to take frame level annotations to LSTM for drowsiness detection [7]. These methods operate under low time responses, especially when comes to Indian conditions. A residual 3D CNN architecture is introduced and compared with similar 2D networks to present the advantages of Spatio Temporal learning [27].

A conditional spatio-temporal data representation using 3D-DCNN framework is introduced for learning through direct inputs for driver drowsiness detection without considering the real time online monitoring of driver states[24]. Through online monitoring, a new 3D Conditional Generative Adversarial Network and Two-Level Attention Bidirectional Long Short-Term Memory (3DcGAN-TLABiLSTM) [10] is introduced and achieves a reasonable frame processing time compared to 36.9 fps in 3D-DCNN. Even though there are enough varieties in CNN, eye and mouth conditions in maximum all the DDD models have low performance as they occupy smaller part of the frames, special features for eye and mouth are used in ensemble Multi-CNN Deep Learning model [5, 1]. R-CNN is introduced as an alternative model through which 93 % of accuracy is received [6]. All the above discussed works have high intrusion, low robustness, and low reliability which require huge processing power. This provides a huge scope and demand for Driver drowsiness detection research.

We propose a multi-stage adaptive 3D-CNN for face positioned DDD System and the proposed innovations are as follows:

1 A three-stage model with a non-intersection over union suppression technique to identify five-points (Left eye, Right eye, Nose, left end of mouth and right end of mouth) along with the bounding

boxes by a feather-like CNN architecture carefully designed to easily operate with frames (Stage 1).

2 A separate adaptive learning (Learning from samples irrespective of its class) for understanding the state of driver is designed using a multistage adaptive 3D-CNN to increase drowsiness classification performance (Stage 2 to 5).

1.1. Preliminaries

Convolutional Neural Network (CNN) is initially introduced [15] as a weighted filter model with multiple connected layers. CNN is popularly used in vision-based tasks such as Image Classification, Recognition and Object Detection. Its design structure has characteristics of high scaling, high degree shifting and misinterpretation of invariances such as defined segmented area in convolution process (temporary space where convolution takes place), Weight sharing and Spatio-Temporal (ST) Sampling. As we use local connection and weight sharing in CNNs, locally minimal meaningful features are extracted and this property of CNN makes it a preliminary feature detector of a small part of image in a set of images. The major part of convolution relays on identifying the feature map and its unit position (m,n) in 2D convolution is given by,

$$a_{ij}^{mn}=\alpha\left[\sum_a^W \sum_b^H \left(x^{ab} w_{ij}^{ab}\right)+ B_{ij}\right], \quad (1)$$

where α is activation function, x is latent information (unit pixel value) of position (m,n) in i^{th} feature map corresponds to j^{th} layer, w is the kernel associated with the local feature map. The feature height and width is H and W and a,b are its initial values. For each feature map generated at each layer, a different bias B_{ij} is associated. The dimensions of feature map are reduced by pooling with spatial adjacent values generated in previous feature maps. The final feature does not only contain local information, it can be combined with another local spatial neighbours to describe whole image. Even though the features from 2D-CNN are robust and has good impact in sequential data applications, it considers only the spatial data that would not be capable to produce good results for time dimension oriented dynamic applications. To process the additional temporal information in sequential data, 3D-CNN is introduced [13]. A 3D Feature map is used to convolve with 3D volume of combined set of image inputs to create a latent 3D feature map for the next layer. Through this method an additional temporal information is captured and its unit position (m,n,t) in 3D convolution is given by

$$a_{ij}^{mnt}=\alpha\left[\sum_a^W \sum_b^H \sum_c^D \left(x^{abc} w_{ij}^{abc}\right)+ B_{ij}\right], \quad (2)$$

where α is activation function, x is latent information (unit pixel value) of position (m,n,t) in i^{th} feature map corresponds to j^{th} layer, w is the 3D kernel associated with the local feature map. The feature height, width and depth is H,W and D and a,b,c are its initial values. B_{ij} is bias associated with the feature map. 2D explores spatial data through image by image while 3D explores spatial and temporal data simultaneously for many images as 3D-kernals can explore additional temporal dimension. We use 2D convolution for face positioning and 3D convolution for classification and sub-classification to identify the driver drowsiness.

Algorithm 1: Driver Drowsiness Detection

Input: Sequence of Driving Images

Output: Drowsy or non-Drowsy Class

1. **for** $i \leftarrow 1$ to frames in image pyramid **do**
2. I-Net, P-Net and O-Net operations
   compute face positioned image, $N \leftarrow image_i$
3. **for** $j \leftarrow 1$ to N positioned images **do**
4. $st \leftarrow learningmodel_{st}(j,\theta)$,extract ST values,
5. **for** $k \leftarrow 1$ to M sub models **do**
6. $op_i \leftarrow learningmodel_U(a; \theta_u)$,
   compute output fn. of models
7. **return** op
8. **end for**
9. **for** op $\leftarrow 1$ to N ($\forall$ op $\in$ $image_i$) **do**
10. $\gamma = learningmodel_{fu}(\mathbf{st}, op; \theta_{fu})$,
11. **for** $\gamma \leftarrow 1$ to N positioned images **do**
12. $R_{Det} = learningmodel_{det}(\upsilon; \theta_{det})$,
    $\upsilon \in Norm.\gamma$,
    compute final detection result
13. **end for**
14. **end for**
15. **end for**
16. **end for**

2. Architecture

The proposed architecture shown in Figure 1(b) contains five major stages, 1. Three stage architecture for face positioning, 2. 3D State Learning (Spatial and Temporal values Learning), 3. 3D State understanding (Training sub models), 4. Feature fusion and 5. Final Detection Model. Initially from the sequence of images, we resize to form an image pyramid then a three-stage architecture is introduced to operate with these images in cascaded manner, in first stage, Input-Net (I-Net) is a fully connected CNN used to obtain the suitable regression vectors to position the face. On obtaining the vectors, we may get overlapped candidates, these candidates are further calibrated and we suppress the overlaps using Non-Intersection Over Union Suppression (NIOUS) [20] over general

Figure 1(a)

Driver state positions

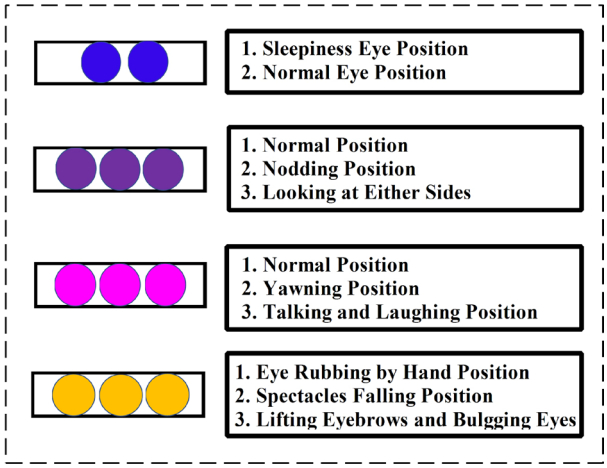
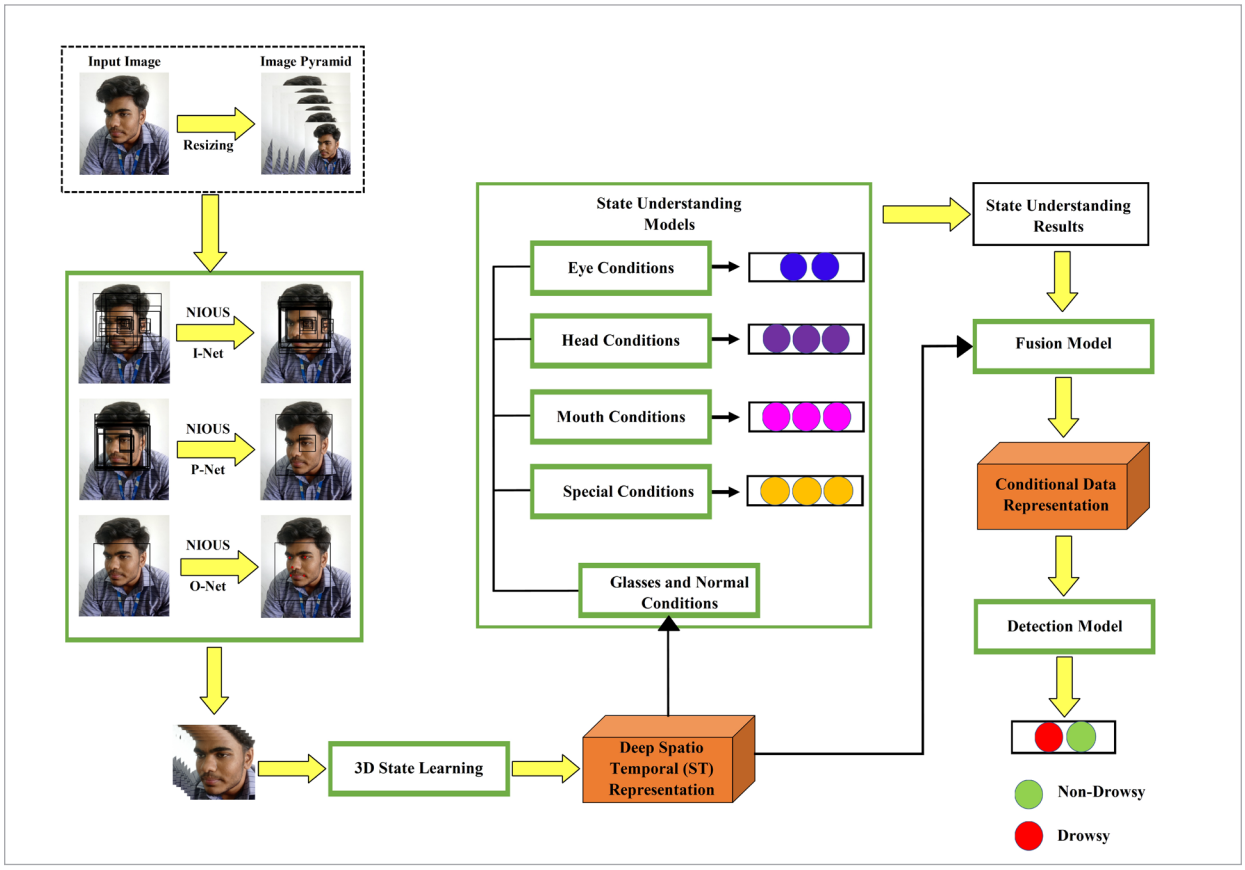


Figure 1(b)

Cascaded form of a three stage CNN producing bounding boxes at stage 1 (I-net) and providing high suppression at stage 2 (Process-Net) and finalizing facial positions in stage 3 (O-net), adaptive state learning structure designed using 3D-CNN (green box) and features/outputs of the designed model (orange box)



Non-Maximum Suppression (NMS) to merge highly correlated vectors. Vectors obtained are given to second stage of CNN, Process-Net (P-Net), this network further refines the false positives of regression vectors obtained from I-Net using NIOUS. In third stage, Output-Net (O-Net) gives us the exact facial positions and ensures the correct regression vectors as shown in Figure 1(b). These outputs are stacked and given as input to 3D-state learning for extracting ST values, we define sub models to understand different driver states. We then use fusion model to understand one or more driver states in a single image using results of sub models. Fusion model extracts conditional features in frames to execute final detection model.

One-hot encoding is a general method of representation of a legal linear combination with high (1) and others a low (0) value. During feature fusion a condition model is developed from these one-hot vectors and ST data. Finally, detection model identifies the drowsiness.

2.1. Face Positioning

Many CNN based algorithms are available for face positioning. We notice several performance limitations due to following reasons: 1. Few filters while performing convolution may fail to differentiate input parameters as lack of diversity. 2. Huge filter size is used and it is not needed, as this problem falls in only two classes (1. Face and 2. Not Face), filter size can be reduced, we fixed it as 3 x 3, this will reduce total computational complexity as we use this stage for identifying driver is available in the frame in order to process the frame for drowsiness detection. The 2D-CNN architecture used for positioning face is given in Figure 3.

2.2. Spatio-Temporal (ST) State Learning Phase

In this section, we describe about state learning model designed using 3D-CNN. 3D-CNN is used to extract the ST data from the sequence of frames. Driver facing cameras in vehicle may record the events at different conditions such as change in background with different lighting situations. As diversifications are high, we must deal with additional temporal dimension. We designed state learning by spatial and temporal values to develop a discriminant feature from the inputs. Spatial values are the exact position of the pixel values in each image and temporal values are the change of pixel values from one frame to another frame with respect to time. Exploring a third dimension in a single image is not possible, we need to have a set of images to identify the change associated with time sequence. We need to process a sequence set of frames simultaneously to identify the ST data. Proposed 3D-CNN is used to extract ST data from the given input sequence. Let $x \in S^{WxHxT}$, x is the input training video and W, H and T are Width, Height, and Temporal length, for input x, the state learning by 3D-CNN is given as

$$st=lm_d(x|P_d),\ st \in S^{W_{st} * H_{st} * D_{st}} \quad (3)$$

P_d is the parameter vector of state learning and st is learnt ST data from the input x. W_{st}, H_{st} and D_{st} is Width, Height, and Depth of ST data. This ST data can also be defined as the activation values for the hidden layer from the last computed convolution layer in proposed 3D-CNN adaptive state learning model. We designed 3D-CNN with 4 convolution and 2 pooling layers and the detailed architecture of 3D-CNN is given in Figure 2. To identify ST data simultaneously, we use 3D-local receptive field and its operation is given by

$$a= \rho\left[\sum_x^{W_{lr}} \sum_y^{H_{lr}} \sum_z^{D_{lr}} \left(v_{x,y,z} w_{x,y,z}+b\right)\right], \quad (4)$$

where W_{lr}, H_{lr} and D_{lr} are width, height and depth of the local receptive field, and v, w and b are input, weight, and its associated bias. The activation value a, triggers the hidden unit functions and ρ is the local activation function used in convolution, we use Rectified Linear Units (ReLU) [14] for all local activations in proposed 3D-CNN. As discussed in earlier sections our designed 3D-CNN extracts spatial and temporal values simultaneously then conveys to state understanding and fusion model to make Conditional feature for drowsiness classification model.

2.3. State Understanding Phase

The goal of this section is to make models to understand driver physiological states and environmental conditions like night time, day time, wearing glasses and other important facial elements of the driver. This will help us to develop an integrated adaptive state learning network according to state conditions. We hypocrite that the data collected (video) is associated with the state conditions and driver drowsiness. These are explained clearly in training and inference

Figure 2

Representation of 3D-CNN Architecture. Purple denotes images stacked after face positioning (i.e., input images for DDD) and green denotes extracted Deep ST Values. Yellow denotes pooling layers and grey are convolution layers respectively. Numbers in top are depths and bottom are 3D kernel volumes of its associated layers

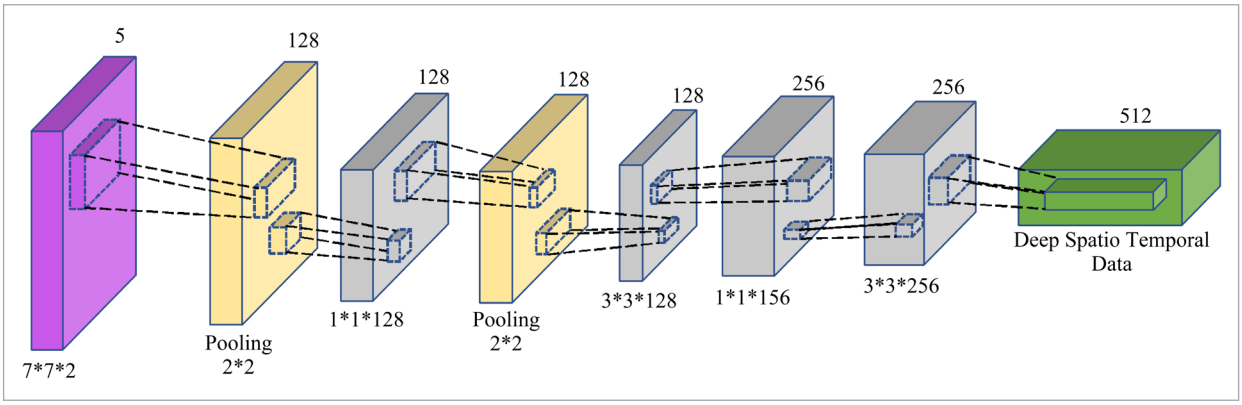
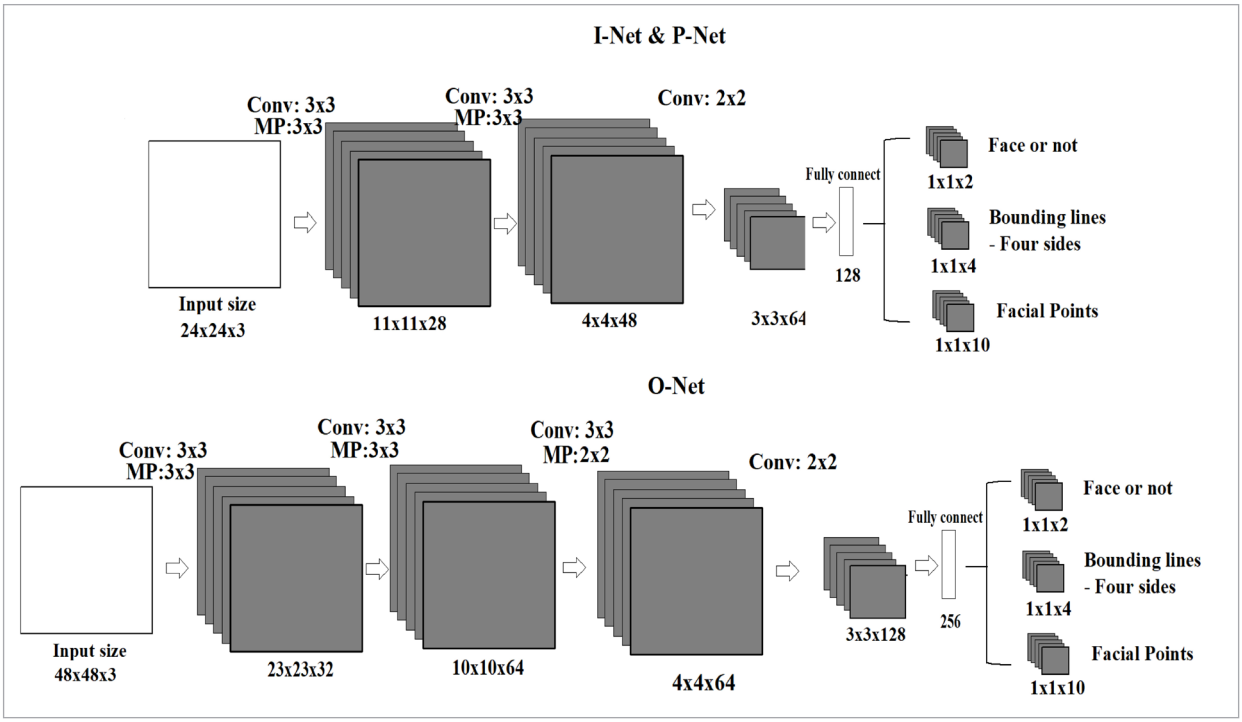


Figure 3

Architecture used for face positioning, I-Net, P-Net and O-Net Architectures with stride and pooling as 1 and 2



section. This proposed work has a main category of glasses and normal state conditions and four sub categories of driver state conditional elements, 1. Glasses and Normal State O_{gn} 2. Head condition model O_h 3. Mouth condition model O_m 4. Eye condition model O_e and 5. other special condition model O_{sc}. We use one-hot vector to define the states and its facial conditions, assigned one-hot vectors are given in Table 1. We assume that linear kernels will face difficulty in handling ST data due to highly overlapped distributions and so we use fully connected Neural Network (NN) to deal with ST data carefully. The predictions

Table 1
Annotations of Models in State Understanding stage

State Conditions	Assigned One-Hot Vector	Driver State Condition
Glasses and Normal State	10000 (A)	Day Normal State
	01000 (B)	Day Glass State
	00100 (C)	Night Normal State
	00010 (D)	Night Glass State
	00001 (E)	Day Sun Glasses
Eye Conditions	{(A), (B), (C), (D), (E)}10 - 10	Sleepiness Eye Position
	{(A), (B), (C), (D), (E)} - 01	Normal Eye Position
Head Conditions	{(A), (B), (C), (D), (E)} - 100	Normal Position
	{(A), (B), (C), (D), (E)} - 010	Nodding Position
	{(A), (B), (C), (D), (E)} - 001	Looking at Both Sides
Mouth Conditions	{(A), (B), (C), (D), (E)} - 100	Normal Position
	{(A), (B), (C), (D), (E)} - 010	Yawning Position
	{(A), (B), (C), (D), (E)} - 001	Talking and Laughing Position
Other Special Conditions	{(A), (B), (C), (D), (E)} - 100	Eye Rubbing by Hand Position
	{(A), (B), (C), (D), (E)} - 010	Spectacles Falling Position
	{(A), (B), (C), (D), (E)} - 001	Lifting Eyebrows and Bulging Eyes

of the models are represented as

$$\begin{aligned}&\{\ \hat{O}_{gn}=lm_{gn}\left(a;\theta_{gn}\right),\quad O_{gn}\in S^{O_{gn}*1}\ ,\\ &\hat{O}_{h}=lm_{h}\left(a;\theta_{h}\right),\quad O_{h}\in S^{O_{h}*1}\ ,\\ &\hat{O}_{m}=lm_{m}\left(a;\theta_{m}\right),\quad O_{m}\in S^{O_{m}*1}\ ,\\ &\hat{O}_{e}=lm_{e}\left(a;\theta_{e}\right),\quad O_{e}\in S^{O_{e}*1}\ ,\\ &\hat{O}_{sc}=lm_{sc}\left(a;\theta_{sc}\right),\ O_{sc}\in S^{O_{sc}*1}\ \}\end{aligned} \tag{5}$$

where $\hat{O} \in \{\hat{O}_{gn}, \hat{O}_{h}, \hat{O}_{m}, \hat{O}_{e}, \hat{O}_{sc}\}$ are the predictions from input data x, $O \in \{O_{gn},O_{h},O_{m},O_{e},O_{sc}\}$ are the input dimension representation of state with conditions and $\theta \in \{\theta_{gn}, \theta_{h}, \theta_{m}, \theta_{e}, \theta_{sc}\}$ are the parameters of its associated model that is given in fully connected network architecture of understanding model. We design all the models with three hidden layers and one output layer. The operative function of these models is given by

$$\begin{aligned}op=f_{op}\{f_{hl3}[f_{hl2}(f_{hl1}(stW_{hl1}+b_{hl1})W_{hl2}+\\ b_{hl2})W_{hl3}+b_{hl3}]W_{o}+b_{o}\},\end{aligned} \tag{6}$$

where st is Spatial Temporal values derived from state learning using 3D-CNN. W_{hl3}, W_{hl2}, and W_{hl1} are weights of hidden layers and W_{o} is weight of output layer, b_{hl3}, b_{hl2}, and b_{hl1} are the bias associated with the hidden layers and b_{O} is the bias in output layer. f_{hl1}, f_{hl2} and f_{hl3} are the activation functions of the hidden layers and f_{op} is the final activation function in output layer. Sub models learns through back propagation, intends to identify a condition for given ST data st, then calculates the difference between predicted and fixed annotations to train network parameters. The output dimensions of the state understanding models will always depends on the target classes to predict. For an instance, the output op of the glasses and normal state understanding model has five target classes for a given ST data as input, similarly sub models are trained to optimize Objective Function (OF) is given as follows,

$$\begin{aligned}&OF_{su}\left(\hat{O}, O;\theta\right)=\{\min\left(\theta_{d}, \theta_{gn}, \theta_{h}, \theta_{m}, \theta_{e},\theta_{sc}\right)\gamma\\ &\sum_{i}\left[OF_{gn}\left(O_{gn},\hat{O}_{gn}\right)+OF_{h}\left(O_{h},\hat{O}_{h}\right)+\right.\\ &\left.OF_{m}\left(O_{m},\hat{O}_{m}\right)+OF_{e}\left(O_{e},\hat{O}_{e}\right)+OF_{sc}\left(O_{sc},\hat{O}_{sc}\right)\right]\},\end{aligned} \tag{7}$$

where $O \in \{O_{gn},O_{h},O_{m},O_{e},O_{sc}\}$ are annotations of input, and OF_{gn}, OF_{h}, OF_{m}, OF_{e}, and OF_{sc} are the SoftMax cross entropy loss functions that calculate the difference between the actual annotation and the predicted results. Then γ is the hyperparameter for regularizing the sum of error values received from loss functions of all sub models. Further details about training are discussed in training and inference section. Using ST data and the results of state understanding models, a new fusion model is created to form a Conditional feature representation for final detection model.

2.4. Feature Fusion Phase

Feature fusion model is designed to learn collection of adaptive-Conditional feature representation from

the ST representation st and state conditional annotations $\hat{O} \in \{\hat{O}_{gn}, \hat{O}_h, \hat{O}_m, \hat{O}_e, \hat{O}_{sc}\}$. Using ST data extracted from 3D-CNN, st ∈ and predicted state conditions of sub models $\hat{O}$, fusion model identifies the collection of adaptive-Conditional feature representation γ. This γ vector is calculated by multiplicative interaction approach [9, 17-19]. The highly dependent and relevant features are identified by multiplicative interaction among the feature maps (element-wise). As the proposed fusion model requires to learn from two form of sources, to handle this, a training procedure defined by Hong et al [9] is adopted. γ corresponding to Fusion model lm_{fu} is given as,

$$\gamma = lm_{fu}(\mathbf{st}, O; \theta_{fu}) \tag{8}$$

$$\gamma = W_{fu}\left(W_{fea}\mathbf{st} \odot W_{gn}O_{gn} \odot W_hO_h \odot W_mO_m \odot W_eO_e \odot W_{sc}O_{sc}\right) + b_{fu}, \tag{9}$$

where $b_{fu} \in S^{d*1}$ is the bias of fusion model and ⊙ represent element wise multiplication. $W_{fu} \in S^{H*d}$, $W_{fea} \in$, are the weights. Here H and d are hidden units and its total count in this fusion layer. γ is unnormalized adaptive Conditional feature representation.

This six-way inner tensor product helps in identifying the correlation among defined sub-classes. Figure 4 shows input images, its ST values and Conditional feature representations. From this procedure, we receive the resultant nearer to zero, which may produce bad results and may sometimes violates the computational procedure by exceeding its range. A SoftMax normalization procedure is used to solve these issues and preserve the dependencies among ST data and outputs of sub models. Normalized γ represented as υ_i is given by

$$\upsilon_i = \frac{\exp(\gamma_i)}{\sum_j \exp(\gamma_j)}, \tag{10}$$

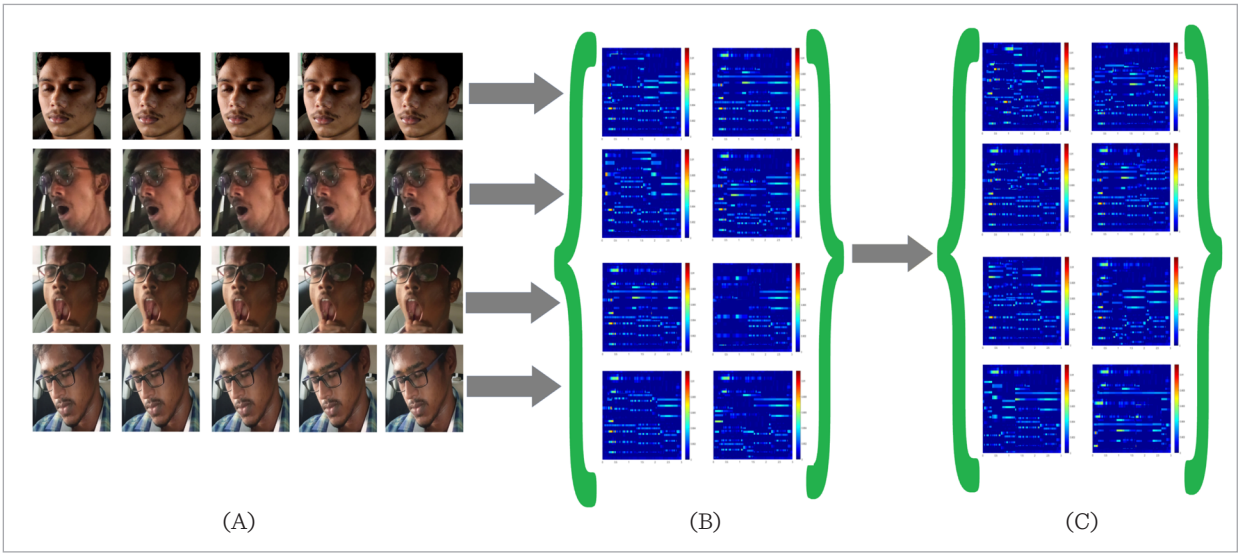
where υ_i is normalized i^{th} element feature and γ_i is unnormalized combined feature at i^{th} element. Now υ represents the Conditional feature over ST data and outputs of sub models, which is an input for final detection model.

2.5. Feature Fusion Phase

Fusion phase provides with cluster of Conditional feature υ, contains feature information about driver expressions of all classes and the final driver drowsiness identification model of the proposed system is constructed using Conditional feature υ. A similar fully connected NN used in state understanding phase is used again upon fusion model to derive the final two class (Drowsy and Non-Drowsy) output is given as

Figure 4 (a-c)

A. Represents input images; B. generated ST value representations; C. conditional feature representations



$$R_{Detect}=lm_{detect}(\upsilon;\theta_{detect}), \quad (11)$$

where R_{Detect} is the final output for parameter set θ_{detect} for final detection model. R_{Detect} falls under two class units (output layer units in this fully connected layer are 1. drowsy and 2. non-drowsy) and we implement SoftMax activation function to measure likeliness of output units for every input samples. If SoftMax function returns a high value in the drowsy unit, then the driver is sleepy and if it returns a high value in non-drowsy unit then the driver is in normal condition. For better results, a common optimization is carried out for both fusion and detection model to minimize the loss by comparing annotated form of final detection with features and it is given by

$$\min(\theta_{fu}, \theta_{detect}) \sum_{i} Er_{det}\left(R_{Detect},R'_{Detect}\right), \quad (12)$$

where is expected outcome of input sample and is SoftMax cross entropy loss function to recalculate the deviations with fixed number of iterations i and embedded to all the previous laid models in the proposed system

3. Training and Augmentation

CNN training in face positioning phase has three objectives, 1. facial classification, 2. regression bounding lines, 3. identifying facial landmarks, to compute results for facial classification, we have two class units (1. face and 2. non-face), for regression bounding lines, we have four class units (1. Left, 2. Right, 3. Length, 4. Width) representing the bounding boxes (candidate window) and for facial landmarks, we have 10 class units (1. Left eye, 2. Not-left eye, 3. Right eye, 4. Not-right eye, 5. Nose, 6. Not-nose, 7. Left end of lips, 8. Not left end of lips, 9. Right end of lips and 10. Not right end of lips) in the final output layer of fully connected layer. If the corresponding values from the units are high, then the proposed system is likely to adhere that class unit. The error minimization is carried out for all 3 stages (I-net, P-net, O-net) to repeatedly train the network and so we use loss functions for error measurements. For facial binary classification, we use cross entropy loss function given in equation (13). For regression bounding lines and facial landmarks, we adopt Euclidean loss function given in equation (14) and (15).

$$Loss_i^{det}= -\begin{pmatrix} O_i^{det}\log(P_i)+\left(1-O_i^{det}\right) \\ (1-\log(P_i)) \end{pmatrix} \quad (13)$$

$$Loss_i^{box}=\left|O_i^{'box}- O_i^{box}\right|_2^2 \quad (14)$$

$$Loss_i^{loc}=|O_i^{'loc}-O_i^{loc}|_2^2, \quad (15)$$

where P_i is probability of being face for input x_i and $O_i^{det}\in\{0,1\}$ is the expected outcome. O_i^{box}, $O_i^{'box}$ are expected and actual regression bounding line targets such that $O_i^{box}\in R^4$. Similarly, O_i^{loc}, $O_i^{'loc}$ are expected and actual facial positions such that $O_i^{loc}\in R^{10}$. There exists a scenario for which $Loss_i^{box}$ and $Loss_i^{loc}$ can be fixed as zero, for example, when the input is only background image (case: without a driver) then there is no need to find bounding boxes and facial positions and these two losses can be set to 0. We handle this situation by introducing a new indicator and the learning rate of face positioning is given by,

$$\min \sum_i^N \sum_{j\in\{det,box,loc\}} \gamma_j \beta_i^j Loss_i^j \quad (16)$$

where γ_j is the rate of importance and $\beta_i^j\in\{0,1\}$ acts as an indicator. The proposed face positioning system of first two stages (I-net and P-net) uses $\gamma_{det}=1$, $\gamma_{box}=0.5$, $\gamma_{loc}=0.5$ and O-net uses $\gamma_{det}=1$, $\gamma_{box}=0.5$, $\gamma_{loc}=1$. Stochastic Gradient Descent (SGD) is deployed to train these three stages and then final facial positions are stacked separately. These stacked images form another image pyramid that are given as input to 3D adaptive state learning network. The proposed 3D adaptive state learning network has two main Objective Functions (OF's) defined in Equation (7) and (12) of state understanding model and detection model. It is very important to consider these performances for optimization to produce better results. From these equations, the objective function of whole proposed system is given by,

$$\min \sum_i^N \sum_{j\in\{det,box,loc\}} \gamma_j \beta_i^j Loss_i^j+\min(P_d,\theta_{su},\theta_f,\theta_{det}) \left[(1-\lambda)OF_{su}\left(O_c,\hat{O}_c\right)+\lambda Er_{det}\left(R_{Detect},R'_{Detect}\right)\right], \quad (17)$$

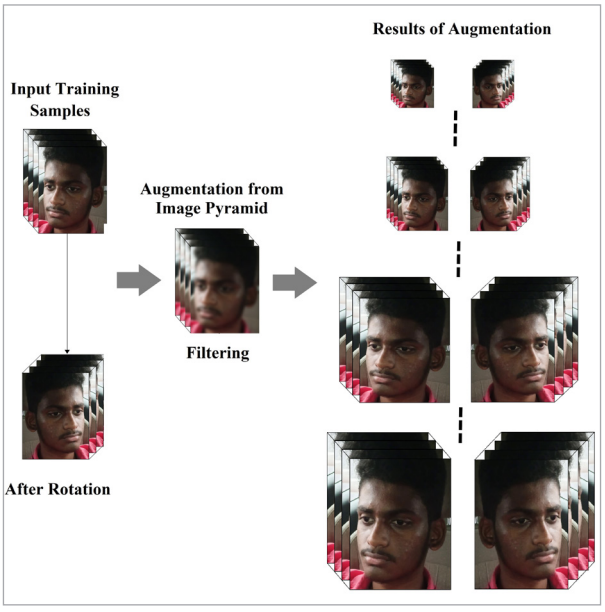
where λ is the balancing parameter for understanding and detection phases. This objective function optimizes all five modules of the proposed system simultaneously. Even though we have five modules to train, we are not training all modules simultaneously and we

train according to the groups that impact output of proposed system architecture, the overall output depends on 1. Face positioning 2. ST learning, 3. State understanding, 4. Fusion and 5. Detection. So first we train face positioning model then ST Learning and State understanding followed by fusion and detection models. For the objective of driver drowsiness identification from video, the proposed system has ST learning that creates ST data used for state understanding to create models and sub- models, then by using the ST data and State understanding information's, adaptive Conditional feature is created by fusion model and using adaptive Conditional feature, drowsiness is detected.

Overfitting is the general issue in all learning models especially in most of the unsupervised learning designs. Generally overfitting can be reduced by transforming the dataset in different ways without losing output labels and inducing the transformed knowledge to the learning system. In our work, the stacked images from face positioning phase are rotated (horizontal transformation). Since the computation for this rotation is significantly low, we created a new data set with low computational cost. The original images and rotated images are transformed using gaussian filter and the transformations are used to additionally train our proposed system to fix the patches in training phase. Further, we generate four different variations of input sample through image pyramid technique to include in the experiments is shown in Figure 5, thus more additional input samples are created with horizontal transformation and image pyramid technique. Without implementing these two augmentation techniques, our proposed system suffers from overfitting and low convergence rate and this is further discussed in ablation experiments.

Figure 5

Augmentation procedure (Creation of 4 different variants to improve test and train size)



4. Experimentation and Analysis

4.1. Datasets

In this section, we evaluate our proposed system with two datasets 1. NTHU-DDD Dataset (Benchmark dataset) and 2. KEC-DDD Dataset (own collection) and compared using major performance metrics. During emerging days of driver drowsiness detection studies, most of the researchers used private datasets, Now-a-days many driver drowsiness detection datasets are available for public access. We requested access for NTHU-DDD dataset through end user license agreement given by NTHU and downloaded the dataset from their FTP server to implement our proposed driver drowsiness detection system.

NTHU-DDD Dataset consists of samples collected from 36 eligible drivers under various scenarios. Training dataset consists of 50% data of drivers (18 members) with five main classes (glasses, no glasses, night glasses, night no glasses, sunglasses) and four sub classes (non-sleepy combination, sleepy combination, slow blink with nodding and yawning) corresponding to each main classes. Totally training has 360 video clips of around 4 to 7 minutes and all put together around nine and half hours, all videos are in 640x480 AVI format and transformed to 256 x 256 pixels each uniformly for proposed system execution. Similarly, evaluation set has videos of 4 drivers with same main and sub classes, totally evaluation has 20 video clips. We consider only 722,225 frames from training and 173,268 frames from evaluation set for our proposed model training as frame level annotations are readily available in the dataset.

KEC-DDD Dataset is created by Department of Artificial Intelligence, Kongu Engineering College, India (Own Dataset) in a simulated environment of vehicle maintenance lab of KEC. Overall dataset consists of

84 driver members under 5 main classes (same as NTHU-DDD) and 11 sub classes (defined according to Indian conditions) of different drowsiness combinations associated with main class. Initially 420 video clips are recorded, then it is sliced for sub classes which consists of totally 4620 video clips of around 90 to 140 seconds per clips. Initially the video is recorded in 30fps at 3840 x 2160 at high quality 4k resolution, then after considering the computational load, images are down sampled using bilinear interpolation method (retains important features) available in OpenCV and are transformed into 256 x 256 pixels each uniformly. The frame level and clip level annotations are defined using non-intersection over union suppression technique and transformed images are stacked in the respective sub classes of around 600 to 800 images each to form a video to improve the experimentation results. For experiments, 60:30:10 rule is followed for training, testing and evaluation procedures. Figure 6 shows the snapshots of KEC-DDD Dataset, as samples of NTHU-DDD is well known to all, we did not display. It is easily observed that more variants (sub classes) are created corresponding to Indian drowsiness conditions in our dataset. Training and evaluation are carried out separately using both the datasets.

4.2. Experimental Results

During initial stage of training, annotations for face positioning is created using non-intersection over union suppression technique are listed below,

- Negatives – where regions of NIOU ratio < 0.275 to ground truth
- Positives – where regions of NIOU ratio > 0.625 to ground truth
- Part facial regions – where regions of NIOU ratio is $0.375 \leq \text{area} \leq 0.675$ to ground truth
- Facial points – labelled five facial locations.

Region ($0.275 < \text{area} < 0.375$) is left during NIOU, as there exists an undecidable variational gap between negative and part faces. During driver face positioned training, these positives and negatives are used for classification, positives and part faces are used for bounding box and facial points are used to localize and provide additional conformation about the positioned faces.

Figure 6

Few Samples of KEC-DDD dataset; (a) eye closed position; (b) eye rubbing by hand position; (c) eye rubbing by hand without glasses position; (d) head nodding position; (e) lifting eyebrow position; (f) looking either side distracted position; (g) normal head, eye and mouth position; (h) spectacles falling; (i) talking and laughing; (j) yawning position

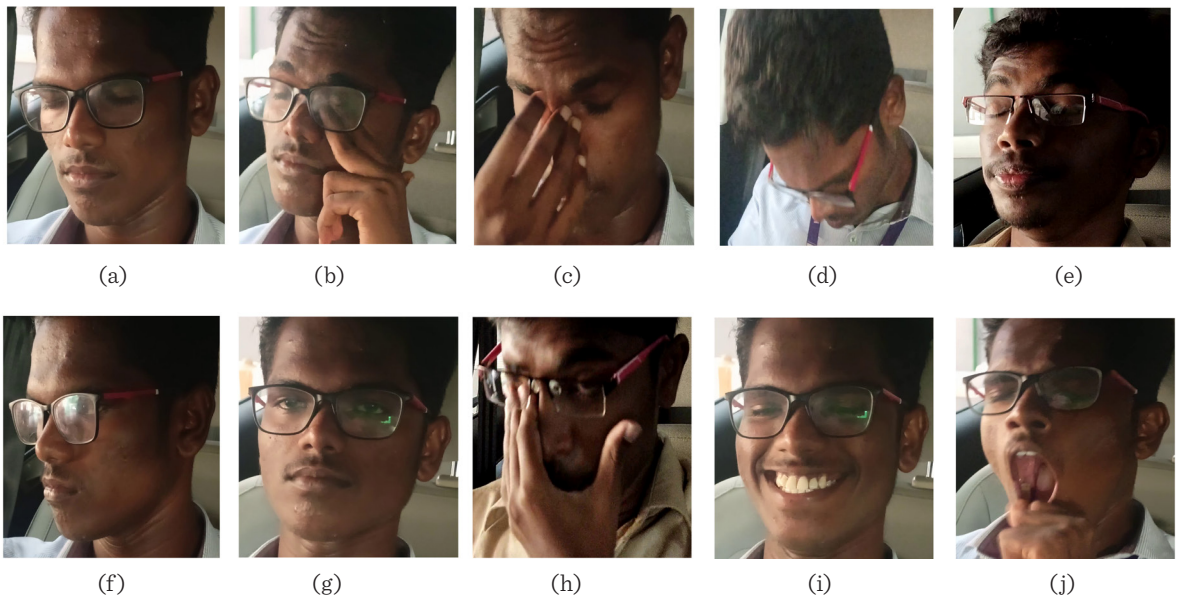


Table 2
Average Validation accuracy of face positioning phase

Methods	Accuracy
Cascaded CNN[4]	95.1
Ours (KEC-DDD)	96.3
Ours (NTHU-DDD)	94.9

We used 2Dconv layer function for face positioning, 3Dconv layer function for ST learning, State understanding and Detection phases, so it is obvious to compare the results with other models at face positioning stage and drowsiness detection stage. We demonstrate our results with the decided evaluation set of KEC-DDD and NTHU-DDD dataset. Performance of the proposed system is measured at face positioning stage and final drowsiness detection stage and compared its results with other models. The positioning phase validation results is presented in Table 2. The training accuracies during state understanding phase are shown in Figure 7 separately for all sub models. The validation accuracies of state understanding models are given by y/x , where y is correct classifications and x is input samples of the respective sub models. Validation results of state understanding phase which composes of five models, 1. Glasses and normal conditions lm_{gn}, 2. Head conditions lm_{h}, 3. Eye Conditions lm_{e}, 4. Mouth Conditions lm_{m} and 5. Special conditions lm_{sc} are shown in Table 3. The average accuracies are calculated by taking mean of respective heads, so that the total classification numbers can be neutralized. Final average accuracy of state understanding phase is 0.888 for KEC-DDD and 0.866 for NTHU-DDD dataset. It is also observed that high accuracies are found in lm_{gn} and comparatively low accuracies are found in lm_{e}, this gaps between sub models are matched by bias of ST learning to support final detection. Output of the state understanding phase is always based on size of the target element, as mouth and eye are small compared to glasses and head region in entire frame in both datasets, models lm_{gn} and lm_{h} may be overfitted and produces good accuracies as shown in Table 3. The overall performance of the proposed system is calculated by F-Measure. It is calculated by harmonic mean of precision and recall, F-measure is given by,

Figure 7
Training accuracies of sub models and overall system on both the datasets

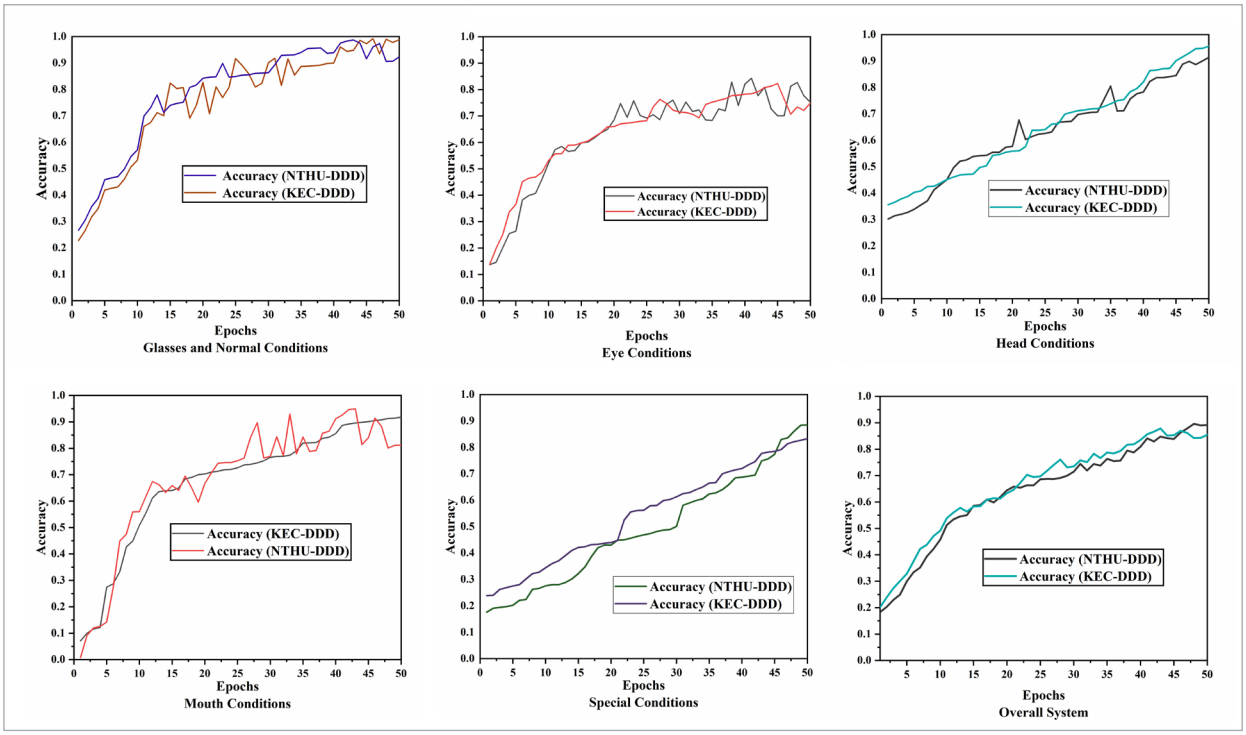


Table 3
Average validation accuracy of models in state understanding phase using KEC-DDD and NTHU-DDD

State/Conditions		Glasses and normal Conditions	Head Conditions	Mouth Conditions	Eye Conditions	Other Special Conditions
Day Normal State	KEC-DDD	0.98	0.98	0.97	0.88	0.84
Day Glasses State		0.96	0.92	0.94	0.80	0.76
Night Normal State		0.98	0.94	0.96	0.81	0.77
Night Glasses State		0.96	0.95	0.87	0.91	0.86
Day Sun Glasses State		0.97	0.96	0.77	0.77	0.71
Average		**0.97**	**0.95**	**0.902**	**0.834**	**0.788**
Total Average		**0.888**				
Day Normal State	NTHU-DDD	0.97	0.97	0.96	0.87	0.83
Day Glasses State		0.95	0.91	0.9	0.79	0.75
Night Normal State		0.97	0.93	0.89	0.77	0.76
Night Glasses State		0.95	0.94	0.85	0.76	0.77
Day Sun Glasses State		0.96	0.95	0.77	0.77	0.71
Average		**0.96**	**0.94**	**0.874**	**0.792**	**0.764**
Total Average		**0.866**				

Table 4
Comparison of average validation accuracy of driver states across models using evaluation sets of KEC and NTHU-DDD

Comparisons/ State	Day Normal State	Day Glasses State	Night Normal State	Night Glasses State	Day Sun Glasses State	Avg.	Ref.
MobileNetV2-DCNN	0.678	0.607	0.563	0.546	0.704	0.617	[12]
BiLSTM-DCNN	0.715	0.627	0.657	0.638	0.713	0.67	[3]
MultiCNN-Deep Model	0.649	0.716	0.748	0.752	0.581	0.689	[1]
3D-DCNN	0.666	0.733	0.765	0.769	0.598	0.706	[24]
R-CNN	0.701	0.768	0.768	**0.804**	0.633	0.735	[16]
Ours (NTHU-DDD)	0.792	0.777	0.761	0.73	0.734	0.759	-
Ours (KEC-DDD)	**0.807**	**0.792**	**0.776**	0.745	**0.749**	**0.774**	-

The best average scores are in bold

Where precision is measuring the grade of positive prediction of the system and recall is ability to predict maximum positive samples, this quantitative measure provides with the average on all videos in all categories. Average accuracy of 0.774 for KEC-DDD and 0.759 for NTHU-DDD dataset is achieved using respective evaluation sets.

We compare our final results with recent methods like MobileNetV2-DCNN [12], BiLSTM-DCNN [3], MultiCNN-Deep Model [1], 3D-DCNN [24] and R-CNN [16] used in driver drowsiness prediction, shown in Table 4. The results shows that the proposed system outperforms almost all other methods in all state conditions except in night glasses state due to low visi-

bility at night time eye positioning when driver is in glasses (both the datasets behave similarly). Individually, our proposed state understanding models perform well even though all others are competitive deep networks. The average accuracy of both drowsiness and non-drowsiness of the proposed driver drowsiness detection system is given in Table 5. The ROC-AUC curve comparing with other methods at face positioning and drowsiness detection stage is shown in Figure 8(a) and 8b, it is noted that the proposed system is very much tolerant to the false positive rate, as approximately its rate is lesser than 0.04 and after that the curve shows greater benefit for the proposed system. Figures 9(a)-(b) shows the sample screenshots of drowsiness detection using KEC-DDD and NTHU-DDD dataset.

Table 5

F-Measures and accuracies of DDD using Evaluation set of KEC-DDD Dataset

State	Drowsy (F-Measure)	Non-Drowsy (F-Measure)	Accuracy
Day Normal State	0.819	0.795	0.807
Day Glasses State	0.799	0.785	0.792
Night Normal State	0.783	0.769	0.776
Night Glasses State	0.752	0.738	0.745
Day Sun Glasses State	0.756	0.742	0.749
Average	0.781	0.767	**0.774**

Figure 8a

ROC-AUC of positioning stage for various datasets

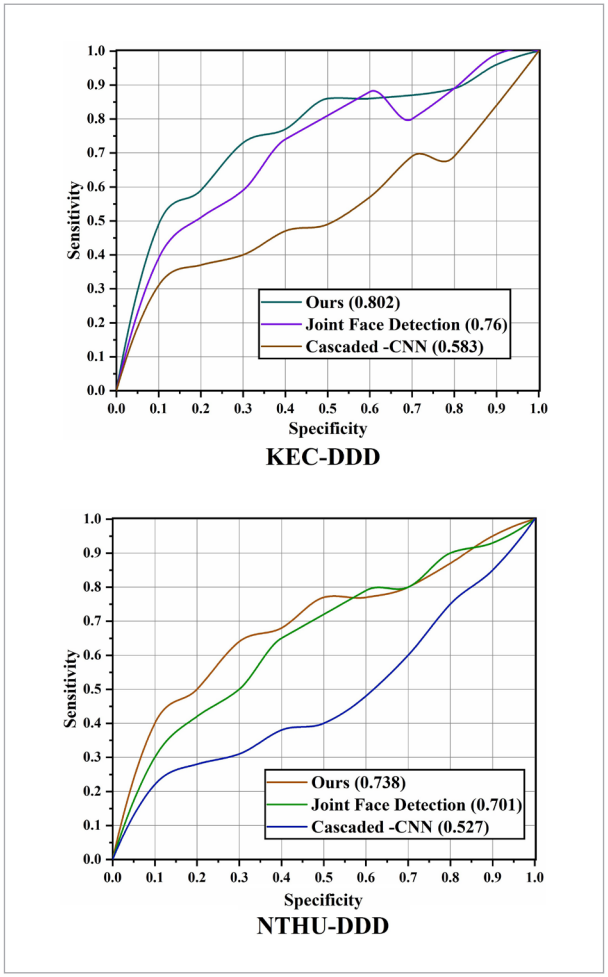


Figure 8b

Final ROC-AUC of whole system for various datasets

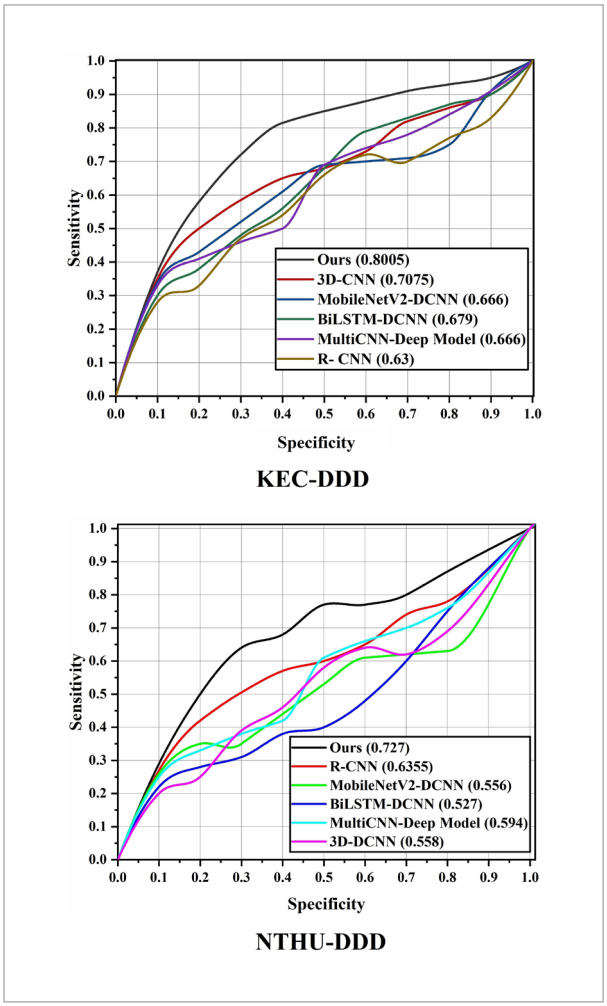


Figure 9a
Output using KEC-DDD Dataset

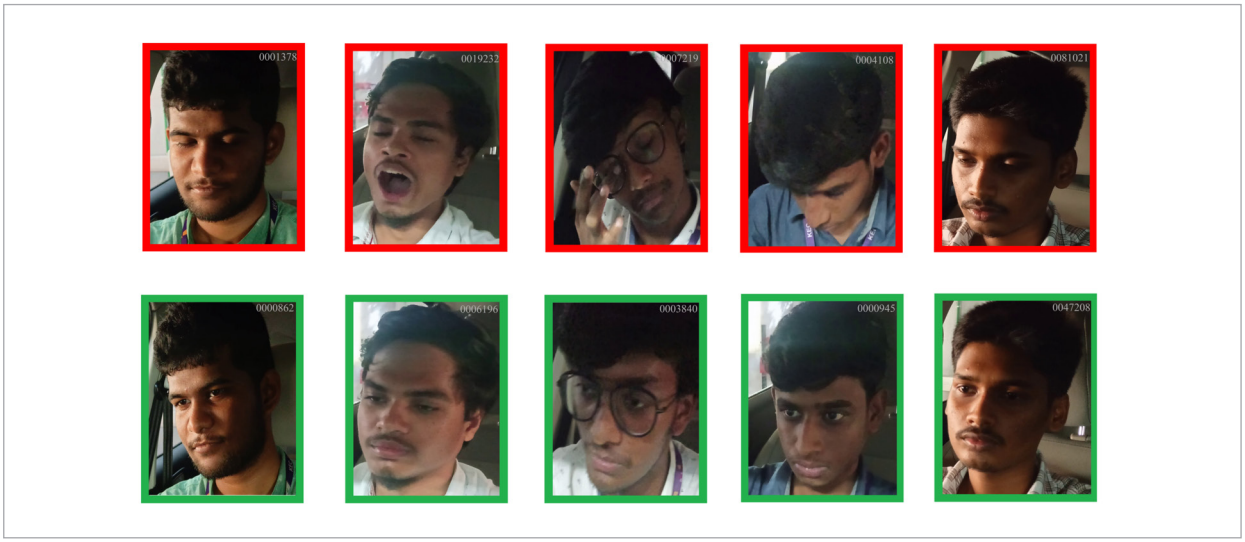
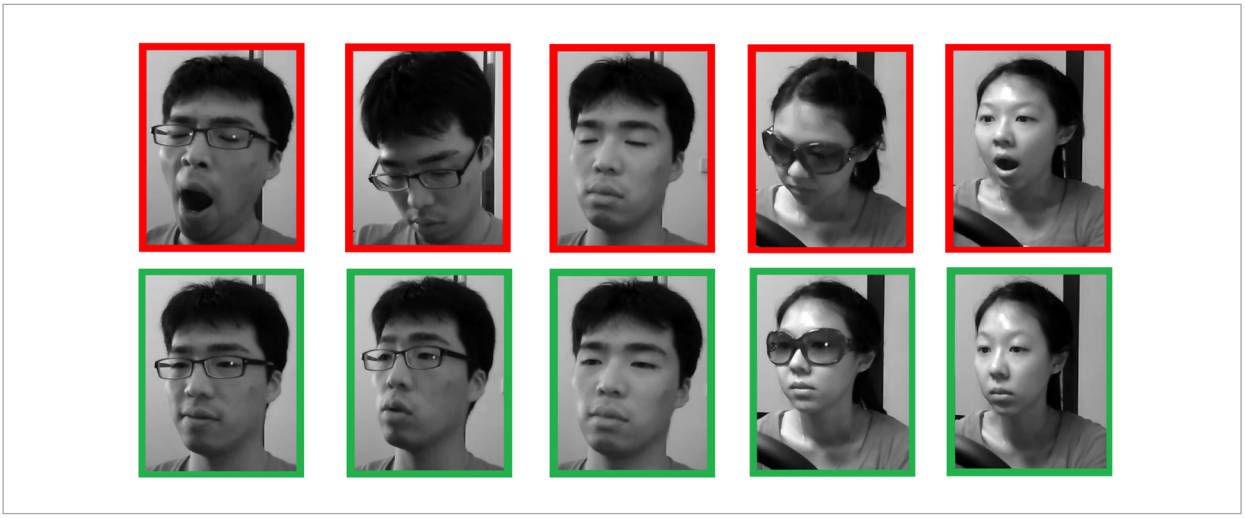


Figure 9b
Output using NTHU-DDD Dataset

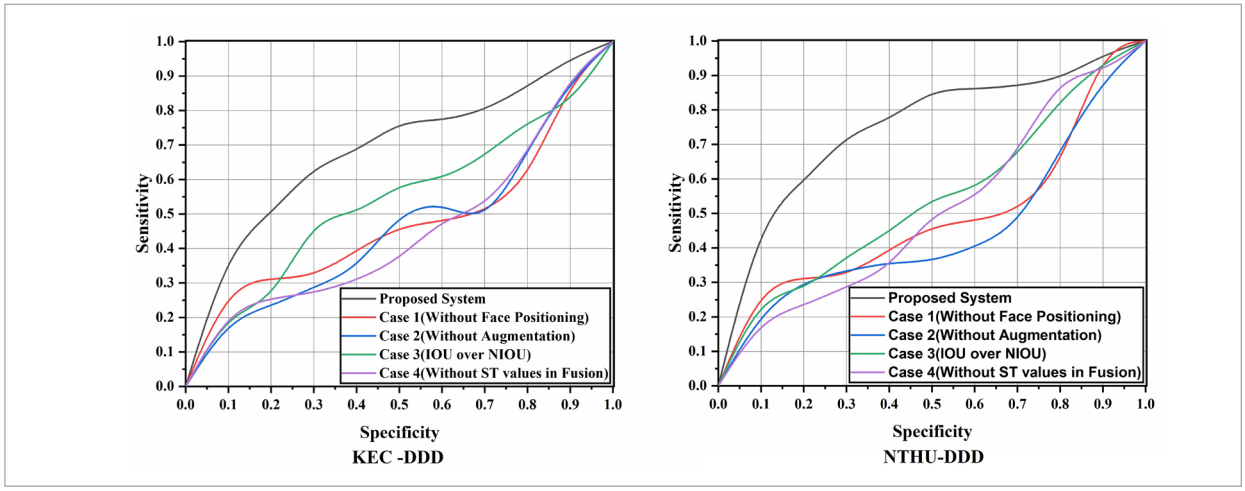


4.3. Ablation Study

Since we introduced a complex architecture with a new dataset, performing ablation study is a seal of approval. An ablation study is carried out with KEC-DDD dataset for investigating the proposed architecture based on four cases, 1. Without performing face positioning, directly giving raw input to the ST learning phase, 2. Excluding the augmented datasets for performing training, 3. Replacing the suppression technique from NIOU to Intersection over Union. 4. Feature fusion exempting the support from ST learning phase.

After blending our system to train for listed cases, models representing each case are evaluated separately. Then the results received for the evaluation set of KEC-DDD dataset for each case are shown with ROC-AUC curve in Figure 10. The outcomes are as expected in almost all cases except for case 3, both

Figure 10
ROC-AUC of our model blended for various cases in ablation study



NIOU and IOU seems to produce a slight variational performance. From Figure 10, our proposed system passes tests made under cases 1,2 and 4, which intuitively provides our architecture performs well.

4.4. Complexity Analysis

Practically for any CNN based architectures, complexity relies on major parameters like input image size, kernel size and pooling size. Training needs higher computation time than testing, since it has to backpropagate and adjust the weights for required iterations. On the other hand, testing has low time complexity as it only depends on result computation. We calculate complexities that applies for both training and testing of proposed system. Furthermore, theoretically, the complexity of our proposed system is given by maximum of 2D and 3D CNN computations taking in our proposed system represented by

$$\text{F-measure}=\frac{2(\text{Precision}*\text{Recall})}{\text{Precision}+\text{Recall}}. \quad (18)$$

where i represents the computations for 1 to n^{th} convolution layer, W_i,H_i,D_i and x_i,y_i,z_i denotes width, height, depth of the inputs and width, height, depth of the kernels in i^{th} layer corresponding to the respective 2D and 3D convolutions. Computational complexity of two layered understanding and detection models is $O(C*N^2)$, where N is hidden layer dimensions and C is targeting domain. We calculate the execution time by leaving output time display, and achieved around 39.6 FPS at 30.2 ms at an average of 400 execution seconds, which is almost processing a live dynamic real scenario.

The proposed system is developed using TensorFlow library and implemented using Core i7 3.4 GHz 16GB RAM with 12 GB GeForce GTX TITAN X. The execution time of different models are compared in Table 6.

Table 6
Comparison for speed across models trained with KEC-DDD Dataset

Method	MobileNetV2-DCNN	BiLSTM-DCNN	MultiCNN-Deep Model	3D-DCNN	R-CNN	Ours (KEC-DDD)
Speed by using 12 GB GeForce GTX TITAN X (Avg. execution seconds- 400)	42.4 FPS	38.1 FPS	34 FPS	36 FPS	35.7 FPS	39.1 FPS
Reference	[12]	[3]	[1]	[24]	[16]	-

5. Conclusion

In this work, architecture for Face positioning, ST learning, State understanding, feature fusion and Detection phases of proposed system is designed and implemented for driver drowsiness detection according to Indian conditions (Indian driver face positions). we started by giving input to face positioning phase that is designed with cascaded three stage 2D convolution layers and face classified outputs are stacked to provide input to Spatio Temporal Learning Stage where the Spatio Temporal values are created and are passed to State understanding phase. Models and sub models defined in state understanding phase are trained to hold the knowledge of respective driver state conditions. Along with the knowledge of state conditions, ST values are passed to feature fusion stage by which a Conditional feature representation is created. This Conditional feature is given as input to final fully connected layer (Detection phase) by which drowsiness and non-drowsiness of the driver is classified. This proposed procedure is carried out using two datasets KEC-DDD (own dataset) and NTHU-DDD training dataset. Additionally, an ablation study to conform effectiveness of our architecture is conducted for four different cases and results are discussed separately. Results of the proposed system are measured for both the datasets at two stages (Face Positioning and Final Detection) and compared with literatures discussed earlier. From the results, it can be concluded that the proposed system outperforms all other methods like 3D-CNN, R-CNN and MultiCNN-Deep Model in Indian conditions (Indian driver face positions) and capable to detect driver drowsiness from 256×256 resolution images at 39.6 fps at an average of 400 execution seconds. Even though we produced acceptable results for driver drowsiness detection, it is still hard to implement in the real time vehicle as we face the limitations like 1. GPU unit in terms of cost, 2. system accepts only labelled samples with huge count at various situations of driver state and 3. System is trained offline and these limitations can be fixed in near future.

Acknowledgement

This work was sponsored by Indian Council of Medical Research – (Ref: Adhoc/113/2017/HSR).

References

1. Ahmed, M., Masood, S., Ahmad, M., El-Latif, A. A. A. Intelligent Driver Drowsiness Detection for Traffic Safety Based on Multi CNN Deep Model and Facial Subsampling. IEEE Transactions on Intelligent Transportation Systems, 2021, 1-10. https://doi.org/10.1109/TITS.2021.3134222
2. Albadawi, Y., Takruri, M., Awad, M. A Review of Recent Developments in Driver Drowsiness Detection Systems. Sensors, 2022, 22. https://doi.org/10.3390/s22052069
3. Ansari, S., Naghdy, F., Du, H., Pahnwar, Y. N. Driver Mental Fatigue Detection Based on Head Posture Using New Modified reLU-BiLSTM Deep Neural Network. IEEE Transactions on Intelligent Transportation Systems, 2021, 1-13. https//doi.org/10.1109/tits.2021.3098309
4. Chen, S., Wang, Z., Chen, W. Driver Drowsiness Estimation Based on Factorized Bilinear Feature Fusion and a Long-Short-Term Recurrent Convolutional Network. Information, 2020, 12, 3. https://doi.org/10.3390/info12010003
5. Dua, M., Shakshi, Singla, R., Raj, S., Jangra, A. Deep CNN Models-based Ensemble Approach to Driver Drowsiness Detection. Neural Computing and Applications, 2020, 33, 3155-3168. https://doi.org/10.1007/s00521-020-05209-7
6. Ed-Doughmi, Y., Idrissi, N., Hbali, Y. Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network. Journal of Imaging, 2020, 6. https://doi.org/10.3390/jimaging6030008
7. Guo, J.-M., Markoni, H. Driver Drowsiness Detection using Hybrid Convolutional Neural Network and Long Short-Term Memory. Multimedia Tools and Applications, 2018, 78, 29059-29087. https://doi.org/10.1007/s11042-018-6378-6
8. Hashemi, M., Mirrashid, A., Beheshti Shirazi, A. Driver Safety Development Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network. SN Computer Science, 2020, 1. https://doi.org/10.1007/s42979-020-00306-9
9. Hong, S., Oh, J., Lee, H., Han, B. Learning Transferrable Knowledge for Semantic Segmentation with Deep Convolutional Neural Network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 3204-3212. https://doi.org/10.1109/CVPR.2016.349

10. Hu, Y., Lu, M., Xie, C., Lu, X. Driver Drowsiness Recognition via 3D Conditional GAN and Two-Level Attention Bi-LSTM. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30, 4755-4768. https://doi.org/10.1109/TCSVT.2019.2958188

11. Hugar, J. G., Naseer, M. M., Waris, A., Khan, M. A. Road Traffic Accident Research in India A Scientometric Study from 1977 to 2020. Digital Communications @ University of Nebraska-Lincoln, 2020. https//papers.ssrn.com/sol3/papers.cfm?abstract_id=3893062

12. Husain, S. S., Mir, J., Anwar, S. M., Rafique, W., Ullah, M. O. Development and validation of a deep learning-based algorithm for drowsiness detection in facial photographs. Multimedia Tools and Applications, 2022, 81, 20425-20441. https://doi.org/10.1007/s11042-022-12433-x

13. Ji, S., Xu, W., Yang, M., Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35, 221-231. https://doi.org/10.1109/TPAMI.2012.59

14. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 2012, 25.

15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. Gradient-based Learning Applied to Document Recognition. Proceedings of the IEEE, 1998, 86, 2278-2324. https://doi.org/10.1109/5.726791

16. Magán, E., Sesmero, M. P., Alonso-Weber, J. M., Sanchis, A. Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images. Applied Sciences, 2022, 12, 1145. https://doi.org/10.3390/app12031145

17. Memisevic, R. Learning to relate images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35, 1829-1846. https://doi.org/10.1109/TPAMI.2013.53

18. Sathesh, S., Pradheep, V. A., Maheswaran, S., Premkumar, P., Gokul Nathan, S., Sriram, P. Computer Vision Based Real Time Tracking System to Identify Overtaking Vehicles for Safety Precaution Using Single Board Computer. Journal of Advanced Research in Dynamical and Control Systems, 2020, 12(7), 1551-1561. https://doi.org/10.5373/JARDCS/V12SP7/20202258

19. Sathesh, S., Maheswaran, S., Mohanavenkatesan, P., Mohammed Azarudeen, M., Sowmitha, K., & Subash, S. Allowance of Driving Based on Drowsiness Detection Using Audio and Video Processing. In Computational Intelligence in Data Science 5th IFIP TC 12 International Conference, ICCIDS 2022, Virtual Event, March 24-26, 2022, Revised Selected Papers. Cham Springer International Publishing, 2022, 235-250. https://doi.org/10.1007/978-3-031-16364-7_18

20. Shepley, A., Falzon, G., Kwan, P. Confluence A robust Non-IoU alternative to Non-maxima Suppression in Object Detection. Computer Vision and Pattern Recognition, 2020. https//doi.org/10.48550/arXiv.2012.00257

21. Vijayan, V., Sherly, E., Thampi, S. M., El-Alfy, E.-S. M. Real-time Detection System of Driver Drowsiness Based on Representation Learning Using Deep Neural Networks. Journal of Intelligent & Fuzzy Systems, 2019, 36, 1977-1985. https://doi.org/10.3233/JIFS-169909

22. Wijnands, J. S., Thompson, J., Nice, K. A., Aschwanden, G. D. P. A., Stevenson, M. Real-time Monitoring of Driver Drowsiness on Mobile Platforms Using 3D Neural Networks. Neural Computing and Applications, 2019, 32, 9731-9743. https://doi.org/10.1007/s00521-019-04506-0

23. You, F., Li, X., Gong, Y., Wang, H., Li, H. A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration. IEEE Access, 2019, 7, 179396-179408. https://doi.org/10.1109/ACCESS.2019.2958667

24. Yu, J., Park, S., Lee, S., Jeon, M. Driver Drowsiness Detection Using Condition-Adaptive Representation Learning Framework. IEEE Transactions on Intelligent Transportation Systems, 2019, 20, 4206-4218. https://doi.org/10.1109/TITS.2018.2883823

25. Zhang, K., Zhang, Z., Li, Z., Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 2016, 23, 1499-1503. https://doi.org/10.1109/LSP.2016.2603342

26. Zhao, L., Wang, Z., Wang, X., Liu, Q. Driver Drowsiness Detection Using Facial Dynamic Fusion Information and a DBN. IET Intelligent Transport Systems, 2017, 12, 127-133. https://doi.org/10.1049/iet-its.2017.0183

27. Zhao, L., Wang, Z., Zhang, G., Gao, H. Driver Drowsiness Recognition Via Transferred Deep 3D Convolutional Network and State Probability Vector. Multimedia Tools and Applications, 2020, 79, 26683-26701. https://doi.org/10.1007/s11042-020-09259-w