Hindawi
Scientific Programming
Volume 2022, Article ID 8775607, 12 pages
https://doi.org/10.1155/2022/8775607

Research Article

An Optimized Approach Using Transfer Learning to Detect Drunk Driving

Ankit Kumar,[1] **Ajay Kumar**,[1] **Mayank Singh**,[1] **Pradeep Kumar**,[1] **and Anchit Bijalwan**[2]

[1]*Department of Computer Science and Engineering, JSS Academy of Technical Education, Noida, India*
[2]*Faculty of Electrical and Computer Engineering, Arba Minch University, Arba Minch, Ethiopia*

Correspondence should be addressed to Anchit Bijalwan; anchit.bijalwan@amu.edu.et

Received 21 July 2022; Accepted 5 September 2022; Published 20 September 2022

Academic Editor: Punit Gupta

Although the statistics show a slow decline in traffic accidents in many countries over the last few years, drunk or drug-influenced driving still contributes to enough shares in those records to act. Nowadays, breath analysers are used to estimate breath alcohol content (BAC) by law enforcement as a preliminary alcohol screening in many countries. Therefore, since breath analysers or field sobriety testers do not accurately measure BAC, the analysis of blood samples of individuals is required for further action. Many researchers have presented various approaches to detect drunk driving, for example, using sensors, face recognition, and a driver's behaviour to confound the shortcomings of the time-honoured approach using breath analysers. But each one has some limitations. This study proposed a plan to distinguish between drivers' states, that is, sober or drunk, by the use of transfer learning from the convolutional neural network (CNN) features to the random forest (RF) features with an accuracy of up to 93%, which is higher than that of existing models. With the same dataset, to validate our research, a comparative analysis was performed with other existing model classifiers such as the simple vector machine (SVM) with an accuracy of 65% and the K-nearest neighbour (KNN) with an accuracy of 62%, and it was found that our approach is an optimized approach in terms of accuracy, precision, recall, *F*1-score, AUC-ROC curve, and Matthew's correlation coefficient (MCC) with confusion matrix.

1. Introduction

As per the Ministry of Road Transport and Highways, India, traffic collisions result in the casualties of approximately four lakh individuals and leave nearly fifty thousand individuals with nonfatal injuries around the country each year [1]. In simple words, if the crowd is going to watch a match in a stadium with a capacity of one lakh and among them, a road crash happens while approaching the stadium, there is a chance that at least one person will die and up to thirty people will hold nonfatal injuries. The victims of such accidents and lethal and nondeadly wounds include weak street clients, such as pedestrians, cyclists, motorcyclists, and other travellers. Other than the citizens' existence lost in misfortunes, they also cause a heavy monetary weight on their families, such as treatment and last rites costs. Likewise, automobile accidents sway public economies, costing countries practically two percent of their yearly growth in domestic production. A driver affected by intoxicants has a critical danger factor for a conveyance accident. Drunk driving dramatically increases road traffic injury risk as the driver's blood alcohol concentration increases. Drunk driving increases road traffic injury risk to varying degrees depending on psychoactive drug abuse. Many solutions are available in the industry to prevent traffic collisions. However, either they are expensive, like autopilot cars, or nonscalable and difficult to implement, such as law enforcement personnel using a breath analyser to check alcohol in the air drivers breathe out [2].

Nevertheless, the world is a bystander to severe traffic collisions and casualties. Let us drive into a considerable effort made by researchers to curb traffic accidents due to drunk driving that happened in the last two decades, inspired by modern technology of their age in a category manner.

The problem statement is to find an efficient system to detect drunk driving that gives accurate results in order to prevent road accidents or injuries under all constraints. Because car accidents significantly affect public economies, costing nations close to 2% of their annual GDP. Driving under the influence of alcohol increases a serious risk factor for a car accident.

Drunk driving detection using Gabor filters and iris recognition: the alcohol detection system focuses on three key objectives using iris recognition and the Gabor filter. The first step is to obtain an image of the iris. Following that, the image must be encoded into a responsive format for calculation and computation. Finally, a signal from the open-source recognition system will control the car/vehicle via a microcontroller and relay circuit attached to the car/ignition vehicle's system.

Neural network for drunk driving detection: researchers use face photos, such as the cheek, chin, neck, ear, and hand, collected by a camera and analyse them by using a 3-layered neural network to determine whether a driver is inebriated or not. However, there were only a few people whose face colour did not affect them after drinking.

Speech-based drunk driving detection: the extraction of low-level auditory characteristics and an n-way direct classification or regression using maximum margin classifiers are two states in a classic approach to speaker state detection. Prosodic contours may be used differently by an intoxicated speaker than by a sober speaker.

Drunk driving detection by using a noninvasive biological sensor system: the air-pack sensor that monitors the AP-PW is housed in a seat. AP-PW is used to measure the digital pulse volume and the breath-alcohol concentration at the same time, utilizing a finger clip photoplethysmography.

Drunk driving detection by pattern: drunk driving detection based on driving patterns makes use of mobile phones as a platform because they combine detection and communication features.

Engine locking system-based drunk driving detection: the alcohol sensor (MQ-3) detects a vehicle speed of zero when the driver starts the vehicle. If the driver is found to be inebriated, the ignition system will be turned off instantly, along with an alarm and communication to the police station.

Detection of drunk driving based on a sensor for alcohol: water bunches with a dripping vapor tension of 47 mmHg and a temperature of about 98.6 degrees Fahrenheit can be separated into positively and negatively levied water bunches by blowing between parallel plate electrodes consisting of a counter electrode to which a voltage is applied, and a detection electrode connected to a picoampmeter in exhaled gas by humans. This helps to determine that the exhaled gas was truly from a person's breath.

Various strategies and algorithms for classifying intoxicated and sober people have been documented to date. Even though automobile collisions due to drunk driving are responsible for a significant number of fatal and nonfatal injuries, the primary reasons for the same are the poor performance of the algorithm being used, unwieldy integration procedures, inadequate training datasets, and less responsive systems. This work advocates a simple, easy-to-implement, scalable, and economical solution equipped with modern technology and futuristic models.

The study is divided into various sections: the first is the introduction portion of Section 1, which addresses the issues of drunk driving as it now exists, and the most recent cutting-edge research studies in the field of the drunk driving detection system are outlined in Section 2. In Section 3, the proposed materials and methodology is described, and Section 4 describes the results and discussion of the performance analysis of the proposed technique. Lastly, the conclusion and future scope are given in Sections 5 and 6, respectively.

2. Related Work

Harkous et al. [3] address the given problem using a 2-phase machine learning system. In phase 1, the vehicle simulator provides time-series sensor data. The sensor data is then selected based on priorities, and after the hidden Markov model (HMM) is applied to sensor data, these sensors gather data from the steering wheel, throttle, vehicle's centre of gravity, lateral acceleration/speed, longitudinal acceleration/speed, vertical acceleration/speed, pitch rate, yaw rate, and roll rate. The hidden Markov model prediction accuracy is 79%, which is the highest for longitudinal acceleration. In phase 2, researchers prioritised lists of sensors higher than a given threshold, and the recurrent neural network (RNN) machine learning algorithm was applied to subgroups of sensor data. The RNN shows a prediction accuracy of 95% with the given subgroup sensor data, classifying the data as either drunk or sober behaviour. Researchers also share experiment results of RNN-HMM and RNN alone, stating that applying both the machine learning algorithms leads to better results.

Chang et al. [4] presented a mode to supersede the definitive breath-type proportions to address expensive devices and hygiene concerns. A two-stage machine learning system differentiates between drunk and sober driving. Stage 1 is to identify the age range of the driver using the simplified VGG network. Data are segregated into age groups of 18–30, 31–50, and ≥51 years to train the model, and the obtained prediction accuracy is 89.62%. Stage 2 is to identify the facial features of an influenced driver using the simplified DenseNet. For this stage, collected data were classified into four categories: nondrinking, drinking within the bar, exceeding the bar, and heavy drinking, and the prediction accuracy obtained here is 87.44%. This experiment also implies that the age factor significantly affects prediction.

Li et al. [5] suggested a technique for drunk driving detection based on the random forest algorithm using feature selection. The driving simulator helps collect the driver's behaviour data, and then features are chosen as per the feature significance in the mentioned algorithm. Next, a dummy variable simulates the real-world environment, such as the geometric markers of nonnative roads. Finally, different classifiers apply selected features to get a holistic comparison of the performance of each. Linear discriminant analysis, support vector machine, AdaBoost, and random forest are the classifiers used. The performance parameters are prediction accuracy, $F1$-score, receiver operating characteristic curve, and area under the curve value. Experiment results show that accelerator depth, speed, distance to the centre of the lane, acceleration, engine revolution, brake depth, and steering angle are used to classify the drivers' states. AdaBoost and RF classifiers have an accuracy of 81.48%.

Mehta et al. [6] created a new dataset named "Dataset of Perceived Intoxicated Faces," which contains audio-visual data collected via semisupervised efforts. The duration of the data collection is up to ten seconds. Deep learning algorithms such as convolutional neural networks (CNN), recurrent neural networks (RNN), 3D CNN, audio-based LSTM, and deep neural networks (DNN) are applied to them as part of the experiment. Behaviour-based detection such as lingo, gait, or facial countenances and biosignal-based detection such as facial thermal images show promising results in detecting intoxicated faces with an accuracy of 88.39%. The future scope is to add eye gaze and head pose information to the network to explore audio and visual fusion strategies for binary classification. Furthermore, there is scope for transferring learning from a vision-based network to an audio-based network.

Bhuyan et al. [7] proposed a method to classify drunk and sober people using their thermal images and walking patterns. The Curvelet transform is used to grab the fringes of a face to identify intemperate people, and SURF (speeded up robust features) detects the temperature flux in the iris and sclera (more tremendous for intoxicated persons). Optical flow is used to demarcate the movement courses of intoxicated or average persons. This work uses RF and SVM algorithms for classification.

Joshi et al. [8] proposed a method using an embedded system to classify drunk and sober people by scanning their faces using a backpropagation algorithm and reacting according to the results processed. This work uses the neural network algorithm for classification, and stopping or slowing down the car is performed by using other tools embedded in the system. This system is further extended to autopilot mode and can easily integrate with two-wheelers.

Takahashi et al. [9] experimented with detecting alcohol presence in humans using facial and other body details caught by a high-definition camera. They incorporated a three-layered neural network to witness alcohol presence in humans and classified them into three stages: sober, mild, and hefty drinking. They obtained an accuracy of 77.3% through combinations of cheek, neck, and hand; they also created a decision tree to distinguish between them. However, some individuals' body colours do not change after consuming liquor. As mentioned, the future scope is to increase accuracy in the detection of drunk people via feeding eyes to neural networks.

Li et al. [10] devised a method to classify intoxicated and sober people using the information received from their smartphones' accelerometers and gyroscopes. A reliable indicator of an individual's level of intoxication is the drinker's walk. This study uses a bidirectional long short-term memory (biLSTM), a convolutional neural network (CNN), a random forest, a multilayer perceptron, and a gradient boost architecture to perform regression analysis on sensor data. The subsequent experiment demonstrates that, in comparison to other architectures, biLSTM and CNN have the lowest root mean square error values of 0.0167 and 0.0168, respectively. In addition, researchers like to build upon this work by including dynamic segmentation to strengthen the algorithm and employing an ensemble of biLSTM networks to lessen bias and noise and to avoid overfitting in their future work.

Bekkanti et al. [11] concocted a computer-based detector to distinguish between drunk and sober individuals using human emotions. To perform regression computation on data gathered from human sentiments, this study uses a multilayered perceptron-based back propagation neural network (MLP-BBN), a convolutional neural network (CNN), an adaptive neuro-fuzzy inference system (ANFIS), a simple vector machine, and a probabilistic neural network. The subsequent experiment shows that MLP-BBN has the highest accuracy, sensitivity, and specificity of 92.01%, 93.90%, and 93.98%, respectively, followed by CNN with 91.9% and ANFIS with 91%.

Soltuz et al. [12] used deep convolutional neural networks (DCNNs) to evaluate face thermal pictures to provide a novel method for subject-dependent intoxication detection. The analysis presents 3 specific DCNN architectures with fifteen, seven, and twenty layers (GoogLeNet). The architectures are trained via transfer learning, using a sizable dataset that includes thermal infrared snaps of the faces of 41 participants. Each subject has a hundred thermal snaps in the dataset. The dataset is equally contained in the subject's sober and intoxicated states. The samples are before and after an hour of alcohol consumption. DCNNs produce pleasing results when facial thermal images are between 93.17% and 98.54%.

Iamudomchai et al. [13] aimed to develop an innovative alcohol detection system using deep learning and infrared (IR) cameras. The initial component is an infrared camera (FLIR) that can capture both infrared and standard snaps of an individual's face. The following component processes the snapshots for alcohol detection based on deep learning technology on a smartphone with the iPhone operating system. Classification accuracy is 85.10 percent (135 population) with four levels of sobriety and a binary classification accuracy of 74.07%.

Bernstein et al. [14] used heart rate data gathered using sensors from multiple participants who drank alcohol, which we turned from a 1D waveform into a set of spectrograms. CaffeNet and AlexNet, two pretrained CNNs, were fed the spectrograms to identify whether the given spectrogram was an instance of alcohol intake. Using 80 training images (40 positives and 40 negatives) and 20 test images (10 positives and 10 negatives), we achieved a test accuracy of 72 percent ($n = 20$, five trials) after adjusting the learning rate, the number of iterations, gradient descent algorithm, as well as the time window, and coloration of the spectrograms.

Willoughby et al. [15] examined samples of 53 people's face snaps after drinking up to three glasses of liquor, extracted features from the images, and used ML to determine whether the participants were sober or intoxicated. According to the researchers, facial lines changed substantially after consuming liquor, and facial landmark

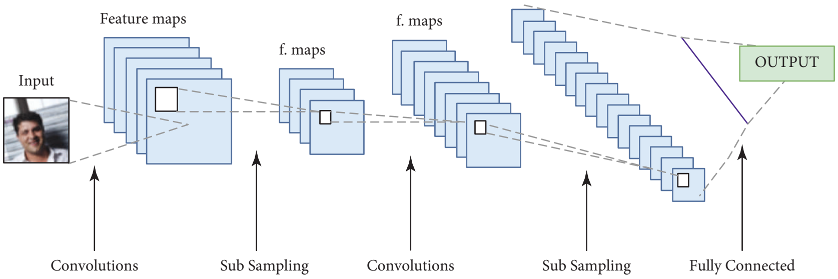


Figure 1: A typical CNN architecture.

vectors showed the most robust predictive features. Using gradient-boosted machines to classify subjects as sober or intoxicated, the regression model achieved an 81 percent classification accuracy. To capture more realistic party/bar scenes, the original dataset was supplemented by blurring, rotating, and adjusting the lighting, which enhanced classification accuracy.

Sajid et al. [16] aimed to design a model for detecting distracted drivers using a publicly accessible dataset, and the dataset is distributed among eight classes: using a cell phone, chatting over a smartphone, driving, operating the radio, tiredness, chitchatting with passengers, peeking back, and consuming alcohol. The proposed methodology uses a pertained model, EfficientNet. This experiment implements five variants of EfficientNet, from which the EfficientDet-D3 is the most acceptable model for detecting distracted drivers with a mean average precision (MAP) of 99.16%.

3. Materials and Methods

This section describes some of the theoretical fundamentals to conduct the experiment which involves the methodologies.

3.1. Convolutional Neural Networks (CNNs/ConvNets). The CNN architecture consists of an input layer, hidden layers, and an output layer. The activation function and final convolution mask the hidden layer's inputs and outputs. The CNN includes a layer that performs a dot product of the convolutional kernel with the layer's input matrix, and the activation function is commonly known as ReLU. The initial convolution operation generates a feature map of the input image, which contributes to the input of the next layer. CNN's hidden layers also contain pooling, dense, and normalization layers [17]. Figure 1 shows the detailed CNN's architecture to explain the convolutions and subsampling to generate the output. The two main parts of CNN's architecture are: Feature extraction is a procedure used by a convolution tool to separate and identify the distinct characteristics of a picture for analysis. There are numerous pairs of convolutional or pooling layers in the feature extraction network. A fully connected layer that makes use of the convolutional process's output and determines the class of the image using the features that were previously extracted. This CNN feature extraction model seeks to minimise the number of features in a dataset. It generates new features that compile the existing features into an original set of features.

3.2. Convolutional Layer. The input image is engrossed in a feature map in the convolutional layer, also known as the activation map. Convolutional layers convolve the input and pass its result to the next layer. Each convolutional neuron processes data only for its receptive field. Fully connected feedforward neural networks are impractical for more significant inputs. FC requires a very high number of neurons, even in a shallow architecture. Convolution reduces the number of free parameters, making the network more profound. Conclusively, CNN is immaculate for data with a grid-like topology like images as convolution or pooling consider spatial relations between features [17].

3.3. Pooling Layer. Pooling layers down sampling the extents of feature maps and it merges the results of neuron tufts into a single neuron. There are two types of pooling layers tuft of neurons in the feature map as shown in [17].

(1) Max pooling uses the utmost value per tuft of neurons in the same manner as shown in Figure 2.

(2) Average pooling uses the moderate value per tuft of neurons.

Average pooling is used in our model by using a moderate value per tuft of neurons.

3.4. Fully Connected Layer or Dense Layer. Figure 3 shows that the fully connected layer is applied to a flattened input where all intakes are connected to all neurons from the next layer. Usually, the end of the architecture contains fully connected layers, and they are used for optimizing objectives such as class scores [18].

3.5. Dropout. The dropout layer unsystematically plunges units and their associations with the neural network during training, discouraging units from co-adapting overly. This is also known as overfitting, as shown in Figure 4 [19].

3.6. Early Stopping. Early stopping is a technique for regularizing model training and is used to sidestep overfitting with an iterative approach, such as gradient descent. Its rules direct how many epochs should be executed before the model commences to overfit, as shown in Figure 5 [20].

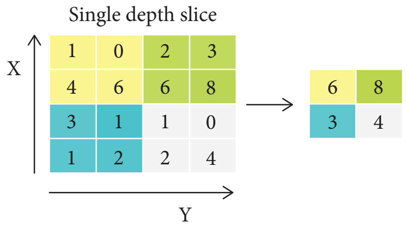


Figure 2: Example of max pooling 2 × 2.

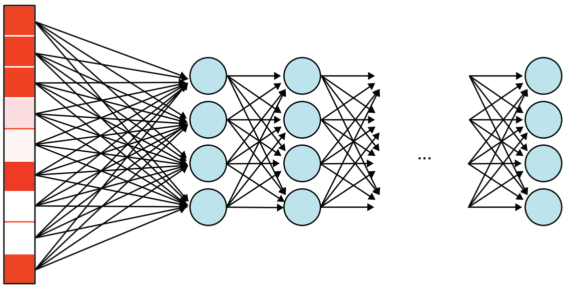


Figure 3: Example of the dense layer at the end of the architecture.

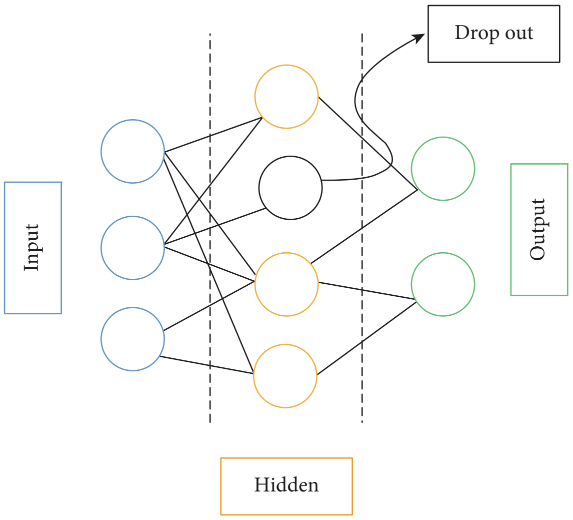


Figure 4: Example of dropout.

3.7. Random Forest Classifier (RF). Figure 6 shows the random forest classifier, which falls under the domain of a supervised machine learning algorithm primarily used for category and deterioration problems [21]. The RF classifier constructs diverse decision trees from a slipshod subset of the training data, and it compiles the votes from given decision trees to settle the final prognosis [22].

3.8. Performance Metrics. Now, let us investigate the basic terminologies of performance metrics..

3.8.1. The Confusion Matrix. It envisions the performance of a classification algorithm in a matrix arrangement of the valid labels versus the model's foreshadowed, where row specimens are a predicted class and column specimens are

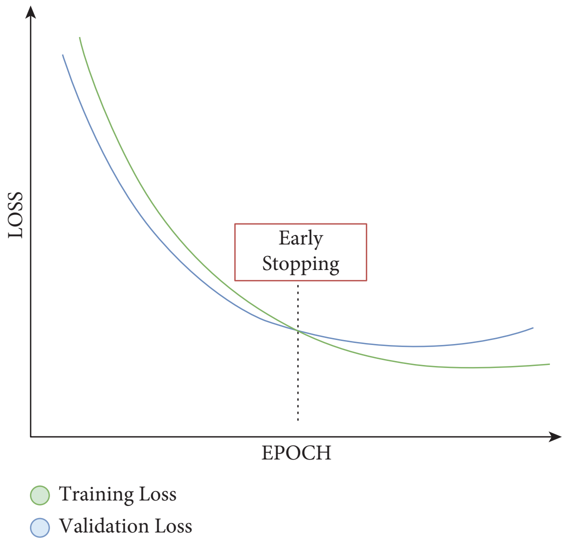


Figure 5: Example of early stopping.

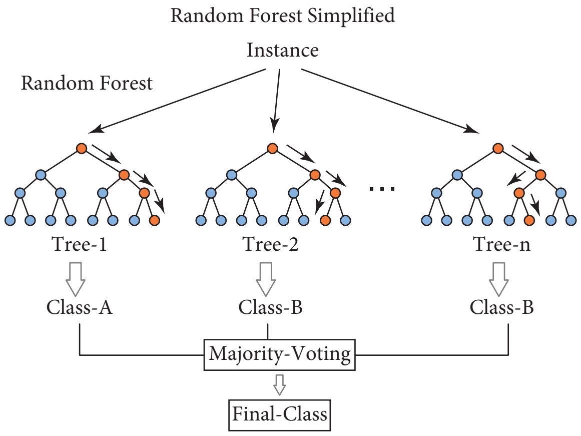


Figure 6: A random forest classifier.

an actual class. The confusion matrix provides the basis of the metrics on which other metrics can evaluate the results [23, 24]. This metric is as follows:

(1) True-positive (TP) is the tally of correctly foreshadowed positive classes over the dataset.

(2) False-positive (FP) is the tally of wrongly foreshadowed positive classes over the dataset, also known as Type I error.

(3) False-negative (FN) is the tally of wrongly foreshadowed negative classes over the dataset, also known as Type II error.

(4) True-negative (TN) is the tally of correctly foreshadowed negative classes over the dataset.

Considering, α as true-positive, then

(1) β: true-negative,

(2) γ: false-positive,

(3) δ: false-negative.

3.8.2. Accuracy. It is the fraction of accurate projections by the count of the input dataset [25].

$$\text{Accuracy} = \frac{(\alpha + \beta)}{(\alpha + \gamma + \beta + \delta)}. \tag{1}$$

3.8.3. Precision. It is a fraction of precisely classified positive classes to the count of predicted positive classes [25].

$$\text{Precision}\left(\text{P}'\right) = \frac{\alpha}{(\alpha + \gamma)}. \tag{2}$$

3.8.4. Recall. It is a fraction of precisely classified positive classes to the count of actual positive classes [25].

$$\text{Recall}\left(\text{R}'\right) = \frac{\alpha}{(\alpha + \delta)}. \tag{3}$$

3.8.5. The F1-Score. The $F1$-score states the equilibrium between precision and recall, ranging from zero to one. This metric usually tells us about classifier accuracy and robustness [25].

$$\text{F1} = \frac{\left(2 * \text{P}' * \text{R}'\right)}{\left(\text{P}' + \text{R}'\right)}\Bigg). \tag{4}$$

3.8.6. Matthew's Correlation Coefficient (MCC). It is a correlation coefficient between the observed and expected binary categories; it returns a value from a negative one to a positive one. The coefficient of a positive one conveys a flawless prediction, zero is better than arbitrary foretelling, and a negative one implies a false projection.

$$\text{MCC} = \alpha * \beta - \gamma * \frac{\delta}{\sqrt{((\alpha + \gamma)(\alpha + \delta)(\beta + \gamma)(\beta + \delta))}}. \tag{5}$$

4. Results and Discussion

This section describes the performance metrics of the machine learning classifier, which were collected during the experiment. The performance metrics are accuracy, precision, recall, $F1$, AUC, and MCC. Random forest, simple vector machine, and K-nearest neighbour were applied over the features extracted from the image process using CNN. Here, RF has surmounted all other classifiers in performance metrics. The AUC of RF is 0.95, and MCC is 0.8783, leading to the highest accuracy of 0.9375, as shown in Table 1.

4.1. Dataset. The dataset [26] contains the facial pictures of human subjects before and after they have consumed alcohol, collected from 53 participants. Each image describes an individual in four states: sober, low drunk, mild drunk, and heavy drunk [27]. Lastly, we have 212 (53 × 4) sets of images that are low in the count for the training model using machine learning algorithms. From there, we turned to image augmentation for experiments, based on which we used Keras APIs for augmentation to multifold given datasets. A sample preview of the dataset is given in Figure 7.

This technique of image augmentation allows the generation of a sample pool of 2,120, which is further split into 80 percent for training and 20 percent for testing. The experiment supplied with a thousand epochs has been limited to 12 epochs to prevent an overfitting model.

The experiment is performed over Google collab with a graphical processing unit (GPU) to enable faster processing of python code. The dataset that resides on Google Drive is mounted to the development environment. Image augmentation is applied while fetching image files from the directory. After that, the augmented data is split into eighty percent training and twenty percent testing. We prepare the CNN model as per the given Figure 8 block diagram, except for the Random Forest Classifier. The CNN model is complied with the Adam optimizer and binary cross-entropy loss. Extract features from the CNN model are sent as input to various classifiers like the random forest, K-nearest neighbour, and support vector machine. The result section incorporates the calculated matters in Table 1.

4.2. Observation of Training of Model Using CNN Alone. A learning curve plots the loss and accuracy of the machine learning model's performance over an epoch. Here, we were able to plot the loss and accuracy curves for the convolutional neural network model with the help of the sklearn metrics library.

In Figure 9, the yellow line shows the curve of training loss throughout epochs, and similarly, the red line shows validation loss over epochs. Here, training loss and validation loss decline at a point of stability, and validation loss has a small deficient area compared to the training loss that deduces that the CNN model is a good fit. Since continued training of a good fit will likely lead to an overfit, an early stopping call back helps restrain the same.

In Figure 10, the yellow line shows the curve of training accuracy throughout epochs, and similarly, the red line shows validation accuracy over epochs. Training accuracy shows strength after a few passes and remains at 100 percent throughout the remaining epoch passes. However, validation accuracy starts with more than 60 percent but slides down to less than 40 percent in the subsequent passes. After ten passes, validation accuracy moves upward to 60 percent and remains constant until the end of the cycle.

Figure 11 shows the confusion metric of the convolutional neural network model. Here, we see that the true-negative is 2, the false-positive is 32, the false-negative is 3, and the true-positive is 31. Using these values, other performance metrics are calculated in the Results section.

The area under the curve (AUC) or receiver operating characteristics curve (ROC) plot envisions a classification model's performance based on accurate and inaccurate classifications. The ROC curve plots the true-positive rate versus the false-positive rate at distinct classification verges.

Table 1: The performance of classifiers over the same dataset.

Classifier	Accuracy	Precision	Recall	$F1$-score	AUC	MCC
CNN	0.5441	0.5238	0.9705	0.6804	0.5441	−0.0563
CNN + KNN	0.625	0.6538	0.85	0.7391	0.55	0.1240
CNN + SVM	0.6562	0.8	0.6	0.6857	0.675	0.3395
CNN + RF	0.9375	1.0	0.9	0.9473	0.95	0.8783

Figure 7: Preview of the dataset.

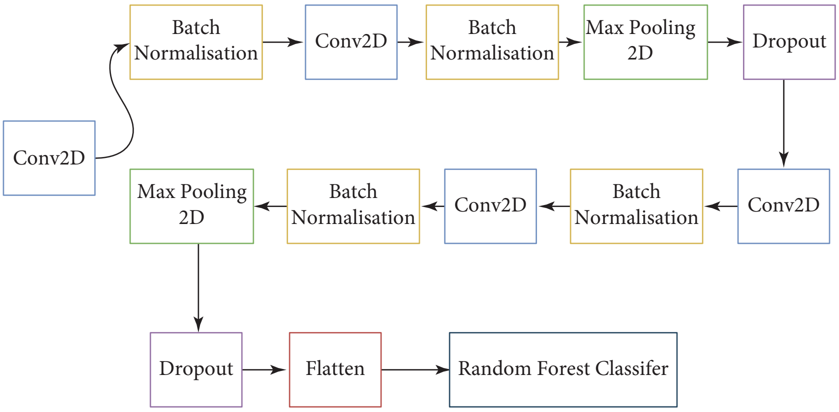


Figure 8: The comprehensive structure of the CNN-RF model from this work.

The area under the curve (AUC) furnishes the capability for a model classifier to discriminate between actual or predicted classes and summaries of the ROC curve [28]. The AUC score is estimated between 0 and 1. The AUC score value near zero represents the classifier model's unsatisfactory version and the score value near one represents the excellent version of the classifier model.

Figure 12 shows the AUC-ROC curve of convolutional neural network model performance. Here, we can see that the AUC-ROC curve moves downward with the mean curve; thus, we conclude that the mentioned model has below-average performance. The AUC score is 0.5441.

In the following observation, feature extraction from the CNN model is passed to other classifiers [29].

4.3. Observation of the Training of Model Using CNN with RF.

Figure 13 shows the confusion metric of the convolutional neural network with the random forest model. Here, we see that the true-negative is 12, the false-positive is 0, the false-

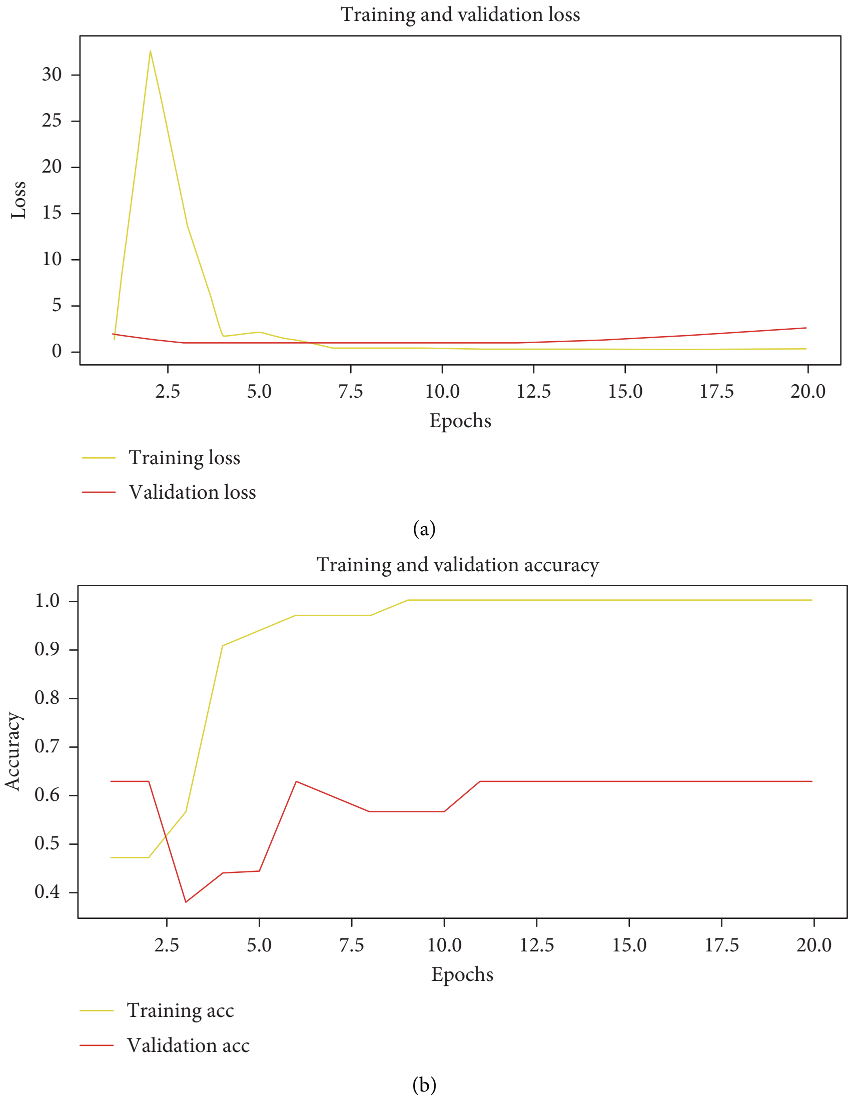


Figure 9: Training and validation loss curves for the convolutional neural network.

negative is 2, and the true-positive is 18. Using these values, other performance metrics are calculated in the Results section.

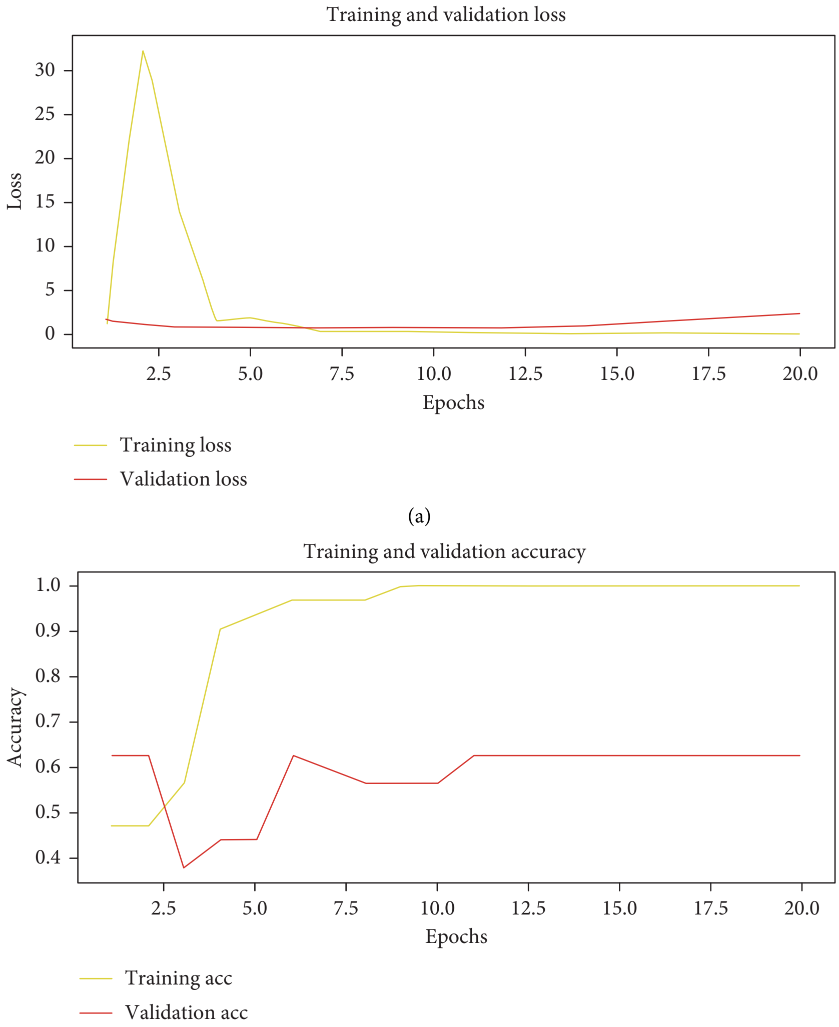
Figure 14 shows the AUC-ROC curve of a convolutional neural network with the random forest model performance. Here, we can see that the AUC-ROC curve stays at the top left corner of the plot; thus, we conclude that the mentioned model has better performance. The AUC score is 0.95, which is close to 1.

4.4. Observation of The Training of Model Using CNN with KNN. Figure 15 shows the confusion metric of the convolutional neural network with the K-nearest neighbours model. Here, we see that the true-negative is 3, the false-positive is 9, the false-negative is 3, and the true-positive is 17. Using these values, other performance metrics are calculated in the Results section.

Figure 16 shows the AUC-ROC curve of the convolutional neural network with the K-nearest neighbour model performance. Here, we can see the AUC-ROC curve move up with the mean curve; thus, we conclude that the mentioned model has above-average performance. The AUC score is 0.55.

4.5. Observation of Training of Model Using CNN with SVM. Figure 17 shows the confusion metric of the convolutional neural network with the simple vector classifier model. Here, the true-negative is 9, the false-positive is 3, the false-negative is 8, and the true-positive is 12. Using these values, other performance metrics are calculated in the Results section.

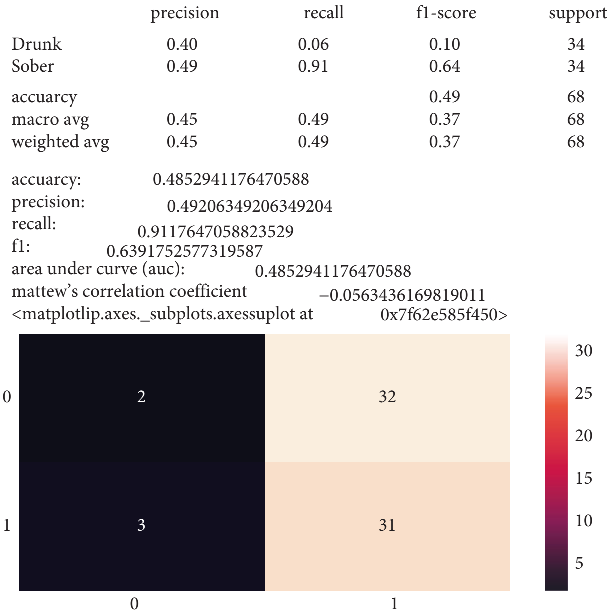
Figure 18 shows the AUC-ROC curve of the convolutional neural network with simple vector classifier model performance. Here, we can see the AUC-ROC curve move up with the mean curve; thus, we conclude that the aforementioned model has better than average performance. The AUC score is 0.675.



(a)

(b)

Figure 10: Training and validation accuracy curves for the convolutional neural network.



```
              precision    recall    f1-score    support

Drunk           0.40        0.06       0.10         34
Sober           0.49        0.91       0.64         34

accuarcy                               0.49         68
macro avg       0.45        0.49       0.37         68
weighted avg    0.45        0.49       0.37         68

accuarcy:       0.48529411176470588
precision:        0.49206349206349204
recall:         0.9117647058823529
f1:         0.6391752577319587
area under curve (auc):        0.48529411176470588
mattew's correlation coefficient        −0.0563436169819011
<matplotlib.axes._subplots.axessuplot at         0x7f62e585f450>
```

Figure 11: Confusion metric for the convolutional neural network.

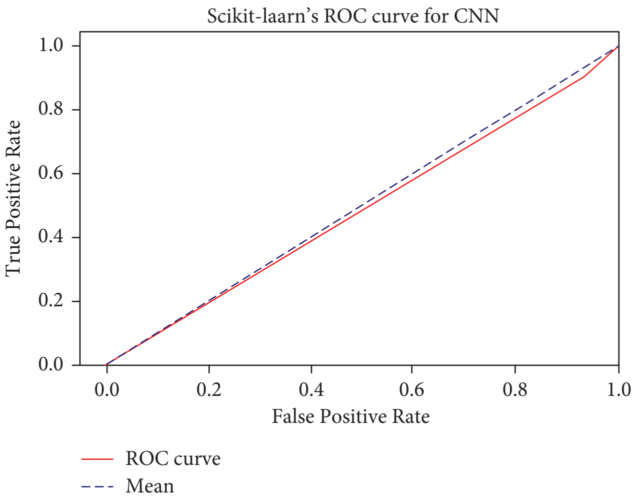


Figure 12: The ROC curve of the CNN model.

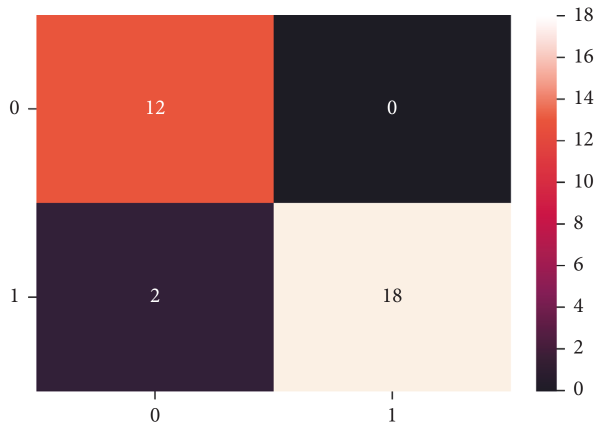


Figure 13: Confusion metrics of CNN with the RF model.

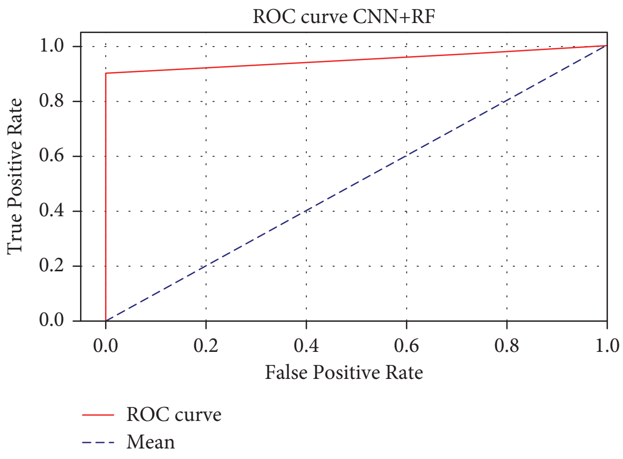


Figure 14: The ROC curve of CNN with the RF model.

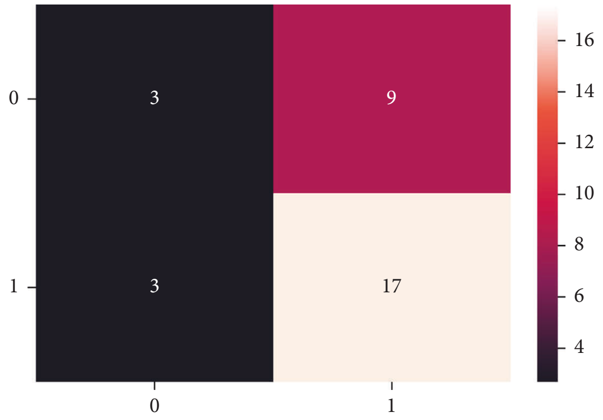


Figure 15: Confusion metric of CNN with the KNN model.

4.6. Validation. Since validation is a vital step to reckon the algorithm's correctness, we validate the same data over two machine learning models: CNN and CNN + RF. The image input to the CNN model erroneously predicts an image of a

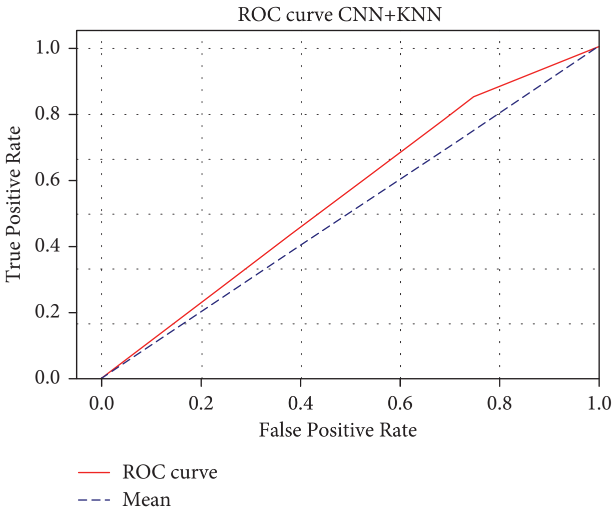


Figure 16: The ROC curve of CNN with the KNN model.

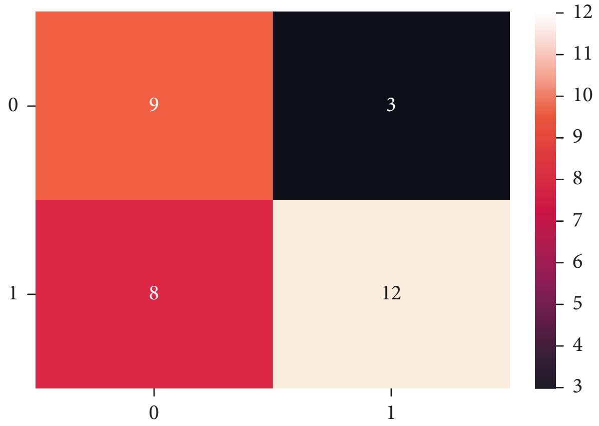


Figure 17: Confusion metric of CNN with the SVC model.

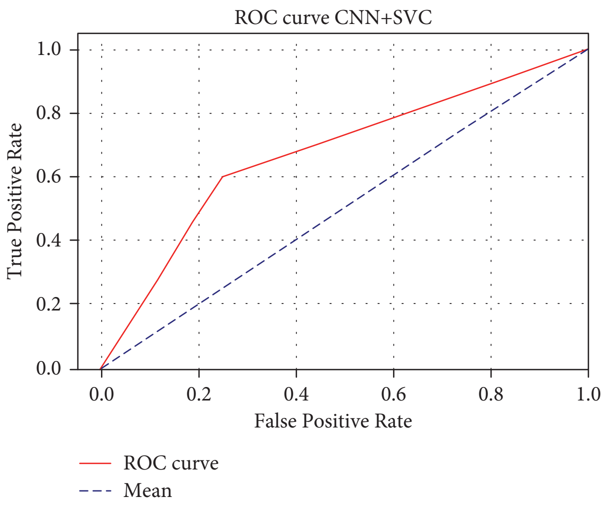


Figure 18: The ROC curve of CNN with the SVC model.

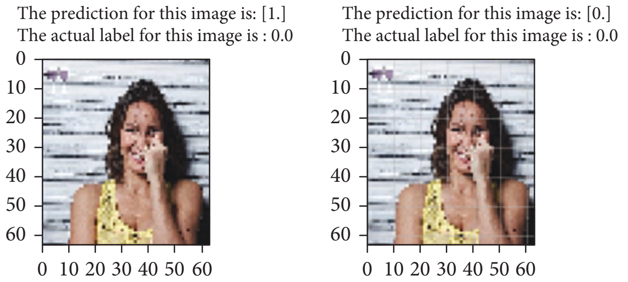

The prediction for this image is: [1.]
The actual label for this image is : 0.0
0
10
20
30
40
50
60
0 10 20 30 40 50 60
The prediction for this image is: [0.]
The actual label for this image is : 0.0
0
10
20
30
40
50
60
0 10 20 30 40 50 60

Figure 19: Validation of the same dataset with the CNN model alone on (a) and the CNN + RF model on (b).

drunk individual, and when the same image is entered into the CNN + RF model, it predicts the image precisely as a sober individual, as shown in Figure 19.

5. Conclusions

This work has documented various techniques and algorithms to classify drunk and sober individuals regardless of vehicle collisions, which account for many fatal and nonfatal injuries. Algorithms with low performance, ungainly integration techniques, poor training datasets, and less responsive systems are fundamental reasons for collisions. Here, we propose a machine learning algorithm with higher accuracy, precision, recall, and $F1$-score that can easily integrate with the mobile ecosystem with the minimum sophistication of hardware structure. Most importantly, it focuses on a noninvasive and portable approach. The proposed technique should reduce the number of crash results, lowering the burden on traffic police, hospitals, and other safety workers. This technique is limited to people of a particular age group (above 18 yrs.). It is yet to be discovered whether it will be applicable to people of the age group below 18 yrs.) [30, 31].

6. Future Scope

Future scope starts with category drivers based on multiple factors like age, gender, geography, experience, and working on sensors that may impact system decisions such as acceleration or brake. Most importantly, the machine learning models should use the proper dataset for training and validation. After that, these machine learning models can be combined with other noninvasive sensor readings to achieve better results and reliability. Our algorithm is fast enough to recognize intoxicated faces with a high-performance rate.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Road Accident Dashboard, "Ministry of road Transport and Highways," 2022, https://morth.nic.in/.

[2] Contributors to Wikimedia projects, "Drunk driving law by country, Wikipedia," 2022, https://en.wikipedia.org/wiki/Drunk_driving_law_by_country.

[3] H. Harkous and H. Artail, "A two-stage machine learning method for highly-accurate drunk driving detection," in *Proceedings of the 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Spain, October 2019.

[4] R. C.-H. Chang, C.-Y. Wang, H.-H. Li, and C.-D. Chiu, "Drunk driving detection using two-stage deep neural network," *IEEE Access*, vol. 9, pp. 116564–116571, 2021.

[5] Z. Li, H. Wang, Y. Zhang, and X. Zhao, "Random forest–based feature selection and detection method for drunk driving recognition," *International Journal of Distributed Sensor Networks*, vol. 16, no. 2, 2020.

[6] V. Mehta, S. S. Katta, D. P. Yadav, and A. Dhall, "Dif: dataset of perceived intoxicated faces for drunk person identification," 2019, https://arxiv.org/pdf/1805.10030.pdf.

[7] M. K. Bhuyan, S. Dhawle, P. Sasmal, and G. Koukiou, "Intoxicated person identification using thermal infrared images and gait," in *Proceedings of the 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, March 2018.

[8] S. Joshi, S. Kate, V. Ketkar, and A. Bharti, "Alcohol detection using face recognition technique integrated with embedded system," *International Journal of Computer Applications*, vol. 158, no. 7, pp. 1–3, 2017.

[9] K. Takahashi, K. Hiramatasu, and M. Tetsuishi, "Experiments on detection of drinking from face images using neural networks," in *Proceedings of the 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*, Hong Kong, China, November 2015.

[10] R. Li, G. P. Balakrishnan, J. Nie et al., "Estimation of blood alcohol concentration from smartphone gait data using neural networks," *IEEE Access*, vol. 9, pp. 61237–61255, 2021.

[11] A. Bekkanti, G. Parasa, A. Krishna, S. Karimunnisa, and C. Z. Basha, "Computer based detection of alcohol consumed candidates using face expressions with SIFT and bag of words," in *Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, June 2021.

[12] A.-I. Soltuz and V.-E. Neagoe, "Facial thermal image analysis with deep convolutional neural network architectures for subject dependent drunkenness diagnosis," in *Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Pitesti, Romania, July 2021.

[13] P. Iamudomchai, P. Seelaso, S. Pattanasak, and W. Piyawattanametha, "Deep learning technology for drunks detection with infrared camera," in *Proceedings of the 2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, Chiang Mai, Thailand, July 2020.

[14] J. P. Bernstein, B. J. Mendez, P. Sun, Y. Liu, and Y. Shang, "Using deep learning for alcohol consumption recognition," in *Proceedings of the 2017 14th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, January 2017.

[15] C. Willoughby, I. Banatoski, P. Roberts, and E. Agu, "DrunkSelfie: intoxication detection from smartphone facial images," in *Proceedings of the 2019 IEEE 43rd Annual*

Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, July 2019.

[16] F. Sajid, A. R. Javed, A. Basharat, N. Kryvinska, A. Afzal, and M. Rizwan, "An efficient deep learning framework for distracted driver detection," *IEEE Access*, vol. 9, pp. 169270–169280, 2021.

[17] H. Wu and J. Zhao, "Deep convolutional neural network model based chemical process fault diagnosis," *Computers and Chemical Engineering*, vol. 115, pp. 185–197, 2018.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, "First steps towards an intelligent laser welding architecture usingdeep neural networks and reinforcement learning," *Procedia Technology*, vol. 15, pp. 474–483, 2014.

[20] Lutz, *Early Stopping — but when? | SpringerLink*Springer, Berlin Heidelberg, 2012, https://link.springer.com/chapter/10.1007/978-3-642-35289-8_5.

[21] L. Breiman, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] D. Mohapatra, B. Subudhi, and R. Daniel, "Real-time sensor fault detection in tokamak using different machine learning algorithms," *Fusion Engineering and Design*, vol. 151, Article ID 111401, 2020.

[23] Z. Noshad, N. Javaid, T. Saba et al., "Fault detection in wireless sensor networks through the randomforest classifier," *Sensors*, vol. 19, no. 7, p. 1568, 2019.

[24] J. Hu, M. Schilder, and X. Yan, "Poster: a neighbor-weighted K-nearest neighbor method for predicting protein subnuclear localizations," in *Proceedings of the 2011 IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, p. 247, Orlando, FL, USA, February 2011.

[25] Y. Sun, H. Zhang, T. Zhao, Z. Zou, B. Shen, and L. Yang, "A new convolutional neural network with random forest method for hydrogen sensor fault diagnosis," *IEEE Access*, vol. 8, pp. 85421–85430, 2020.

[26] anishgollakota, "GitHub - anishgollakota/Intoxication-Identifier: an iOS application that takes live facial data and predicts whether one is intoxicated," GitHub, 2022, https://github.com/anishgollakota/Intoxication-Identifier.

[27] Dividing Images Into Equal Parts Using OpenCV In Python, "GeeksforGeeks," 2021, https://www.geeksforgeeks.org/dividing-images-into-equal-parts-using-opencv-in-python/.

[28] B. V. Dey, "Understanding the AUC-ROC curve in machine learning classification," Analytics India Magazine, 2021, https://analyticsindiamag.com/understanding-the-auc-roc-curve-in-machine-learning-classification/.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co- adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

[30] A. C. Braun, U. Weidner, and S. Hinz, "Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — a comparison," in *Proceedings of the 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–4, Lisbon, Portugal, June 2011.

[31] D. Tanouz, R. R. Subramanian, D. Eswar, G. V. P. Reddy, A. R. Kumar, and C. V. N. M. Praneeth, "Credit card fraud detection using machine learning," in *Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 967–972, China, May 2021.