DRUNKENNESS FACE DETECTION USING GRAPH NEURAL NETWORKS

Vighnesh Bhaskar Kamath
dept. Computer Science and Engineering
PES University
Bengaluru, India
vighneshkamath43@gmail.com

Sagar S Pai
dept. Computer Science and Engineering
PES University
Bengaluru, India
sagarspai61@gmail.com

Shwetha S Poojary
dept. Computer Science and Engineering
PES University
Bengaluru, India
spoojary016@gmail.com

Ananth Rastogi
dept. Computer Science and Engineering
PES University
Bengaluru, India
ananth.rastogi1729@gmail.com

Srinivas K S
dept. Computer Science and Engineering
PES University
Bengaluru, India
srinivasks@pes.edu

***Abstract*—The fatalities associated with driving while intoxicated (DWI) are on the rise, leading to a staggering twelve thousand people dying from it and nine lakh people getting arrested every year. DWIs are usually confirmed with the use of breathalyzers, which require the subject to blow into the machine. In light of the current pandemic caused by COVID-19, a susceptible individual may deny blowing into the machine. Thus, the need for a contactless method to detect if someone is drunk arises, so that suspects are prevented from taking advantage of the situation. This also assists law enforcement in the detection of DWI cases. The proposed study is the method to detect intoxication in a given suspect through Graph Neural Networks using facial landmarks. We also present a labeled dataset as a complementary dataset for intoxication detection. This dataset is the first graph-based data available for the detection of alcohol intoxication. Extensive experiments were carried out to validate this approach.**

***Keywords*—** DWI, Intoxication, Alcohol, GNN, Landmarks, Delaunay, Augmentation

I. Introduction

National statistics show that around 12000 people die every year and around 9 lakh individuals are arrested in DWI (driving while intoxicated) incidents [1]. Alcohol consumption in the workplace greatly affects performance and is noted to have a negative impact on staff morale and confidence. Alcohol, especially in the present era, has an excessively awful effect on any event. The number of reports of harassment and abuse inflicted under intoxication is increasing every day.

There are numerous consequences of alcohol consumption on the body, with two of these being decreased motor function and coordination as well as impaired selection [2]. According to C.M. Steele's study, intoxicated human beings have a hard time multitasking, and alcohol may lead them to only remember the most outstanding cue after drinking [2]. Inebriated drivers are likely to overlook many consequences of choosing to drive or not drive after drinking, which contributes to the high number of DWI cases. A few studies have focused on impact caused by alcohol on riding or driving abilities of individuals with very high blood alcohol levels, despite the existence of numerous studies on alcohol damage. Likewise, fewer studies have made distinctions between drinking and driving based on driving ability. Furthermore, discrimination based on riding or driving performance that is, the process of non-contact could also be more applicable.

Currently, law enforcement officers use breath-analyzers (or breathalyzers) to determine the level of intoxication by measuring the Blood Alcohol Content (BAC), while other techniques involve Bio signal captures — such as electrocardiograms [3] and thermal images [4] of faces, requiring specialized devices.

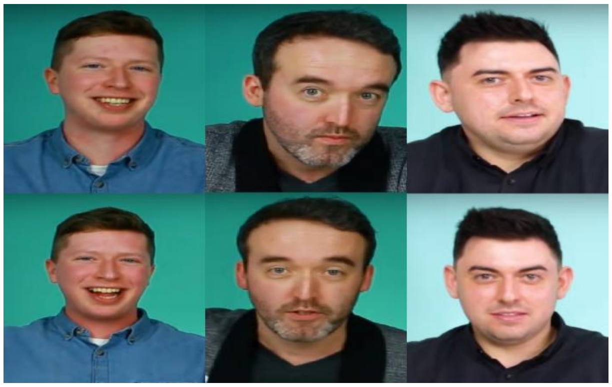


Figure 1. Face images representing the changes in facial expressions post intoxication. The top row shows face images in a sober state and the bottom row shows face images post intoxication.

The study explores the changes that occur in a person's facial features due to alcohol consumption. There is evidence to indicate that people's facial expressions change when they consume alcohol [5]; furthermore, the person feels drowsy and shows signs of fatigue [6]. Here, Deep Learning is used to make the machine identify if a person is intoxicated as a result of changes in his or her features. The key contributions of this paper are as follows:

- A contact less method to predict if the suspect is drunk or sober using face images.
- We believe that this is the first graph-based dataset for developing learning systems to detect if someone is sober or drunk.

II. Literature Review

"Drunk person identification using thermal infrared images" [4], identification of a drunk person is done with help of thermal images. Two approaches are presented by the author by analyzing the radiometric values of the face to determine intoxication. An initial approach involves using pixel values of specific points on the face as features. Eventually, these features were reduced into two dimensions, which revealed clusters move toward the same direction when alcohol consumption increases. A drunk and sober feature space was created for classification out of this feature space. According to the author, the second hypothesis was based on the observation that a particular facial location increased in thermal value with alcohol consumption. The thermal images were divided into small chunks of 100 pixels area to produce of matrix of dimension 8x5. The researchers observed an increase in temperature near the mouth and nose as opposed to the forehead in the experiment. Despite the advantages, the experiment was inconvenient due to the high hardware costs. Moreover, thermal images differed according to individual skin color.

"Drunken Selfie Detection" [7], shows the use of machine learning models to classify photographs as drunk or sober. In that article, it is revealed how smartphones can be used for detecting drunkenness. Additionally, the author details all the methods that have been employed to find an answer to the given problem. It also explores how the removal of certain features affects the performance of the model, specifically discussing the different data augmentation techniques they used and the different photos taken sober and after drinking one, two, three glasses of wine that was created as part of a project named "Three glass Later". They employed technique like random forest classifiers, SVCs, k-neighbors algorithm, etc. The pipeline used by the researchers performed face detection and alignment, feature extraction, feature selection, and machine learning classification. An app for the Android platform that could take photos and classify them using machine learning was also developed, with a level of accuracy of 81±3%. Having limited dataset size and using data generated in a very constrained environment, this dataset could not be used in real-time analytics.

A state-of-the-art dataset was proposed in the paper "DIF: Data set of Perceived Intoxicated Faces for Drunk Person Identification" [8] which contains audio-visual material obtained from online sources of sober and intoxicated people. As part of the intoxication detection research, Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) were trained to compute audio and video baselines, respectively. In order to extract spatio-temporal features from a video , Convolutional Recurrent Neural Network (CRNN) and 3D CNN are made use of, this is for the visual analysis. Next, a two-layer Perceptron with ReLU activation on the OpenSmile generated feature, using dropout and batch normalization was trained as the audio baseline. Finally, an ensemble approach which involved decision tree was used for optimization. The final probability $\bar{p}$ is computed using Equation 1,

$$\bar{p} = \sum_{i=0}^{m} w_i \, p_i \tag{1}$$

where p_i and w_i are the probability and the weight by the i^{th} model. Experiments are done on weighted and average ensemble strategies.

III. Background

In the following section, we discuss various techniques that we used to design the learning system for intoxication detection.

A. Facial Landmark Localization

The detection of facial landmarks can be viewed as a subset of the problem of predicting shapes. An image will be given to a shape prediction model which will localize some key points along the border of the shape. When we apply this task to facial landmark detection our goal is to detect important facial structures on the face by predicting their shapes. Therefore, we can split up analysis of facial landmarks into two parts.

1) Face detection
2) Detect the key facial structures on the face

The first task can be achieved using various methods. The goal of the algorithm is to obtain the face bounding box. Many strategies have been developed to address the problem of landmark localisation. The approach described by Kazemi et al. [9] returns 68 key facial landmarks when trained on iBUG 300-W data-set [10]. The landmark equation is given as shown in Equation 2.

$$\mathrm{LM} = \{ [x_i, y_i] \mid 1 \le i \le 68 \} \tag{2}$$

Where LM represents Landmarks, x and y represents the coordinates.

B. Delaunay Triangulation method

Delaunay method [11] in computational geometry is defined as the triangulation method for a set of P discrete points such that there are no points within the circumcircle of any triangle. We make use of Scipy's Delaunay method for further implementation.

C. Graph Neural Network

Graphs are better at representing entities and relations between them. Usually, an adjacency matrix, node features, edge features, and labels are required to give any graph data as an input to a neural network.

The intuition behind GNN [12] is that a given node is identified by connections and neighbors it. The common operation that follows is node feature updating and aggregation as defined in Equations 3 and 4.

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}) \tag{3}$$

$$o_v = g(h_v, x_v) \tag{4}$$

where f is the aggregation function and g is the update function. g when used at last layer becomes output function, x is the input feature, h is hidden state, co[v] is set of edges connected to node v, $ne_{[v]}$ is set of neighbors of node v, x_v represents features of v, $x_{co[v]}$ represents the features of its edges, $h_{ne[v]}$ is the states of the nodes in the neighborhood of v and $x_{ne[v]}$ is the features of the nodes in the neighborhood of v. The learning algorithm is based on a gradient descent strategy.

D. Graph Convolution Network

Nodes in a graph can receive messages along with their connections with their neighbors. GCNs [13] are different from label propagation as they pass the node features rather than the label. GCNs operate on one hop of their neighborhood. The neighbor features are aggregated and projected to another dimension which refers to the new feature representation of the node, which is shown by Equation 5. This process has to take place for all the individual nodes of the graph. Multiple GCN layers can be stacked to capture long-range information.

$$H^{(l+1)} = \mathrm{ReLU}(\tilde{D}^{1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}) \tag{5}$$

where, $\tilde{A}$ is the adjacency matrix A with self-loops for the individual nodes, $\tilde{D} = \Sigma_j \tilde{A}_{i,j}$ sum of the degrees of the individual nodes along the row, $W^{(l)}$ is the trainable weight matrix. $H^{(l)} \in R^{NxD}$ is a matrix of activation's in the l^{th} layer.

IV. Data Preparation and Pre-Processing

In the following sections, we will examine the step-by-step approach to creating graphs and its pre-processing techniques, after which these graphs will be used as inputs to the models for training and validation.

A. Data set Collection

The data was collected by taking snapshots of video from YouTube and Periscope, using the labelled links [8]. Videos were downloaded and a haar-cascade face detection model was used so that only clear faces are cropped from videos. In addition to this relevant term such as “alcohol”, “drunk” and “whiskey” were used to search other videos. This classification of cropped images was manually verified to ensure there is no mislabeling of data. Around 160 videos, 80 under each class were used, we were able to accumulate around 50 valid images from each video and create a large dataset for developing a classification model.

B. Image to Graph Conversion

Each image was initially passed through a face detector which gave a bounding box of the face. If no face was found the sample was discarded. The cropped face was fed into a facial-landmark locator. The landmark locator returns 68 key-facial landmarks, which were treated as nodes or vertex as shown in Equation 6. The corresponding x and y coordinates were used as node features as defined in Equation 7.

$$V = \{i \quad |1 \le i \le 68\} \qquad (6)$$

$$X_{v_i} = [\, x_i, y_i] \qquad (7)$$

Where X represents feature, V represents set of nodes, V_i represents i^{th} vertex and [x, y] represents x and y coordinates of the i^{th} vertex. These coordinates are later used as input Delaunay Method (DM) which returns a set T, which consists of coordinates of M distinct triangles as described in Equations 8 and 9.

$$T = DM(X_V) \qquad (8)$$

$$T = \{[a_i, b_i, c_i] \mid 1 \le \; i \; \le \; 68 \,, a \; \ne b \ne c\} \qquad (9)$$

Since we focus on creating a non-directed graph, we consider each side of the triangle as a non-directed edge and form a set E which contains all the edges as shown in Equation 10. Figure 2 shows how an input image with a face is converted into a graph using series of steps.

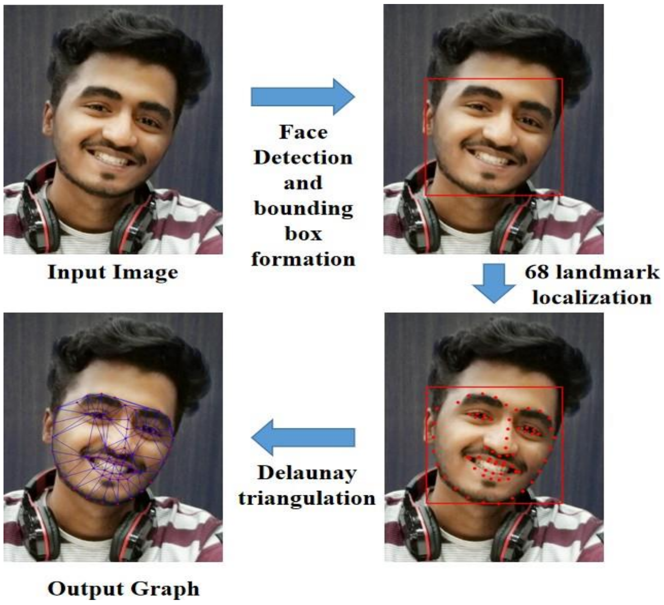


Figure 2. Figure representing conversion of an input face image to graph using series of steps

$$E = \{[a, b], [b, a], [b, c], [c, b], [c, a], [a, c] \quad |\forall[a, b, c] \in T\} \qquad (10)$$

We later create a 2-D binary adjacency matrix A of dimension 68x68 as described by Equation 11.

$$A_{i,j} = \{1 \;\; \text{if } [i, j] \in E, \text{else } 0\} \qquad (11)$$

C. Graph Normalisation

Normalisation becomes important as the photos may come from different resolutions and the node features may vary a lot. For the following data, z-score normalisation cannot be used since it becomes important to maintain the structure of the graph. Hence, we perform graph-wise normalisation. Each of the graphs which were formed from the image was normalised based on the techniques mention in the paper [14]. The individual node feature was normalised graph-wise using equations 12,13 and 14.

$$\hat{X}_v = \frac{X_v - \mu}{\sigma} \qquad (12)$$

$$\mu = \frac{1}{|V|} \quad \sum X_v \qquad (13)(7)$$

$$\sigma = \sqrt{\frac{1}{|V|}\sum(X_v - \mu)^2} \qquad (14)$$

Where X_v is the node feature vector of the graph, μ is the mean calculated across the nodes of the graph G and σ is the standard deviation calculated across nodes of the graph.

D. Gaussian Noise Augmentation

For an image graph, the node features are the x and y coordinates of facial landmarks on Euclidean space. To this Gaussian noise with mean 1 and standard deviation within the range of 0 to 0.15 is added. The reason behind specifying the range of standard deviation is to ensure that the internal structure of the graph is not disturbed and also so that noise doesn’t deviate from the Gaussian distribution. In the real-time application of the facial landmark recognition model, some errors may crop up in locating landmarks. This augmentation procedure aims to reduce impact of these errors. Hence, we input noise in our data so that the model does not deviate in the decision due to these errors. For a given node V, $X_V = [x, y]$ represents its node feature. For every graph G = (V, E) in the data, we create another graph $\widehat{G} = (\widehat{V}, \widehat{E})$ by applying Equation 15 to every node of the graph G.

$$\hat{X}_{v_i} = \hat{X}_{v_i} + \text{GaussianNoise}(\mu, \sigma) \qquad (15)$$

where μ and σ represent mean and standard deviation of the Gaussian distribution respectively.

E. Rotational Augmentation

Rotational Augmentation is the rotation of the node features about a given reference point. For every graph G=(V, E) in the data, we create another graph $\widehat{G} = (\widehat{V}, \widehat{E})$ by applying equations 16, 17, 18, and 19 to every node of the graph G.

$$k^0 = (X_v{}^0 - X_{v_{ref}}{}^0) \qquad (16)$$

$$k^1 = (X_v{}^1 - X_{v_{ref}}{}^1) \qquad (17)$$

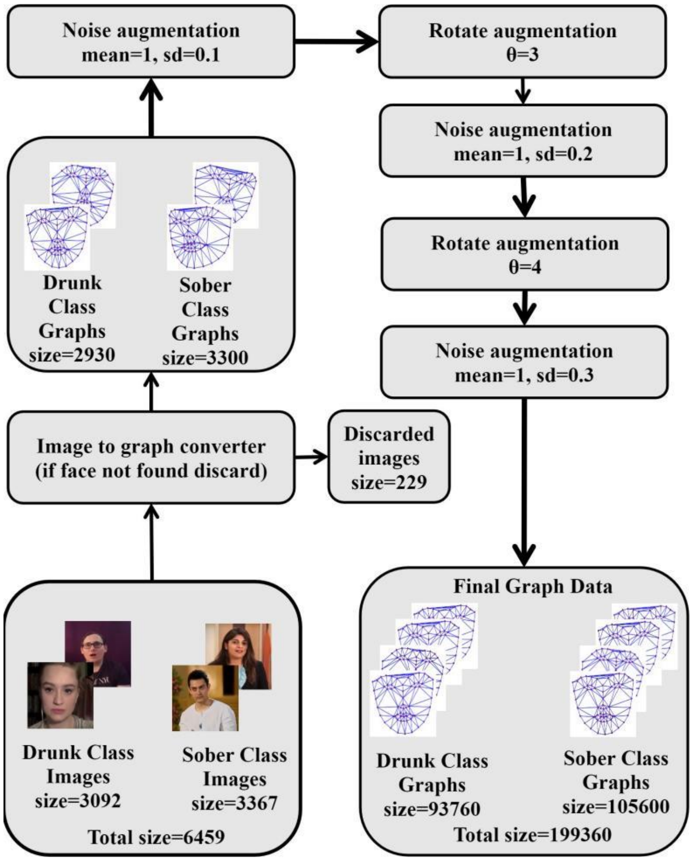
$$\hat{X}_v^0 \;+\; \cos(\theta)\times k^0 \; - \sin(\theta) \times k \qquad (18)$$

$$\hat{X}_v{}^1 = X_{v_{ref}}{}^1 + \sin(\theta) \times k^0 + \cos(\theta) \times k^1 \qquad (19)$$

where θ is the angle of rotation, $X_{v_{ref}}$ is the [x, y] values about which the nodes are to be rotated, $X_v{}^i$ is the feature vector containing only the i^{th} value i.e i = 0 only x values.

F. Final Graph Data Preparation Pipeline

After data collection, we obtained 3092 images of drunk people and 3367 images of sober people, the size of the dataset being 6459 images. These data were pre-processed and then augmented using the techniques as described in Figure 3 to obtain final data of size 199360 graphs.



Noise augmentation
mean=1, sd=0.1
Rotate augmentation
θ=3
Noise augmentation
mean=1, sd=0.2
Rotate augmentation
θ=4
Noise augmentation
mean=1, sd=0.3
Drunk Class Graphs size=2930
Sober Class Graphs size=3300
Image to graph converter
(if face not found discard)
Discarded images size=229
Final Graph Data
Drunk Class Images size=3092
Sober Class Images size=3367
Total size=6459
Drunk Class Graphs size=93760
Sober Class Graphs size=105600
Total size=199360

Figure 3. Graph Data Preparation Pipeline

Which consists of 93760 graphs of sober class and 105600 graphs of drunk class.

V. Proposed Intoxication Detection Model

In this section, we discuss the proposed model and its architecture for intoxication detection. We use variants of GNN which takes the prepared graph as the input and outputs the Drunk and Sober as shown in Figure 4.

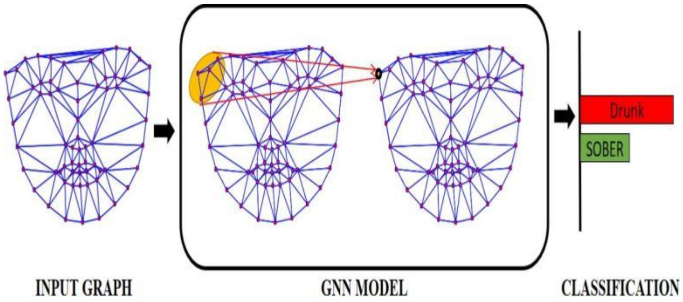
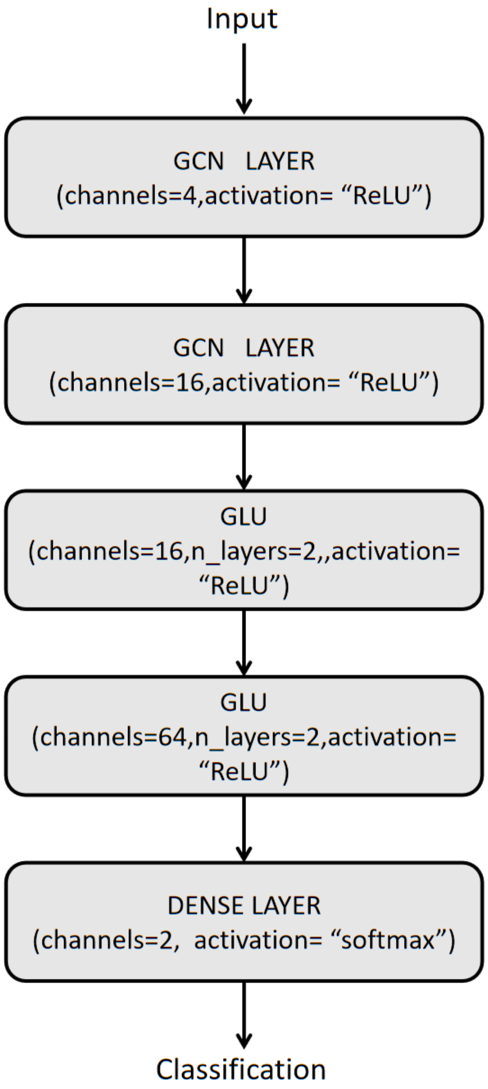


Figure 4. Proposed Classification Model

To create a GNN model we use stacked GCN layers followed by a Gated Linear Unit which uses GRU for aggregation. Gated Linear Unit is used to prevent vanishing gradients. The detailed architecture is shown in Figure 5.



Input
GCN LAYER
(channels=4,activation= "ReLU")
GCN LAYER
(channels=16,activation= "ReLU")
GLU
(channels=16,n_layers=2,,activation= "ReLU")
GLU
(channels=64,n_layers=2,activation= "ReLU")
DENSE LAYER
(channels=2, activation= "softmax")
Classification

Figure 5. Architecture of GNN mode

VI. Results

The following sections details the results achieved. A random seed of 2 was used to shuffle the data three times, and the train, test, and validation sets were divided in the ratio 7:2:1. The same should be used for future improvement. The entire process of training and validation was conducted in Google Colab. Adam optimizer is used, with learning rate of 0.001, epochs of 200, patience of 10 and batch size of 64.

Training accuracy of 86.69 ± 2.29 was achieved with validation and testing accuracy of 86.73 ± 2.6 and 86.4 ± 2.4 respectively. The training and validation accuracy curve is shown in Figure 6.

In every graph, we introduce a special node known as the master node. In the graph, the master node is connected to every node, i.e., every node has an edge with the master node. As a master node, it ensures that messages from two distant nodes are passed correctly. We choose the nose node as master node as shown in Figure 7.

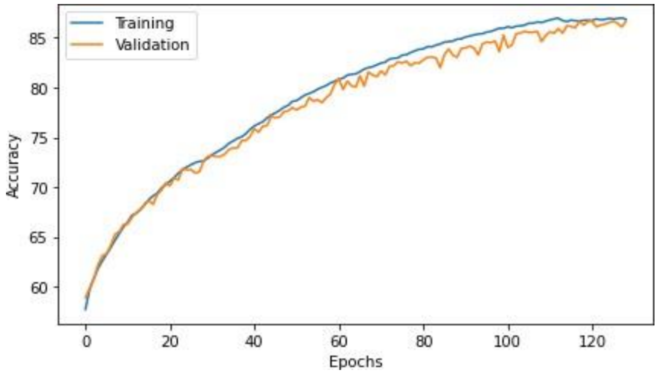


Figure 6. Training and Validation, Accuracy of the model

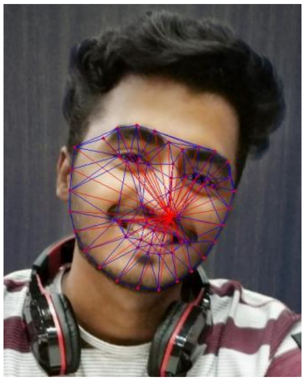


Figure 7. Graph with master node. Edges in red represent the master node connections

We use the same training configuration as mentioned above and train the model after introducing the master node. We achieve Training accuracy of 88.7 ± 2.21 was achieved with validation and testing accuracy of 89.09 ± 2.46 and 89.5 ± 2.51 respectively. The training and validation accuracy curve for the same is shown in Figure 8. The confusion matrix of the test data is represented in Figure 9. Table 1 shows a clear comparison between the results of the two training approaches.

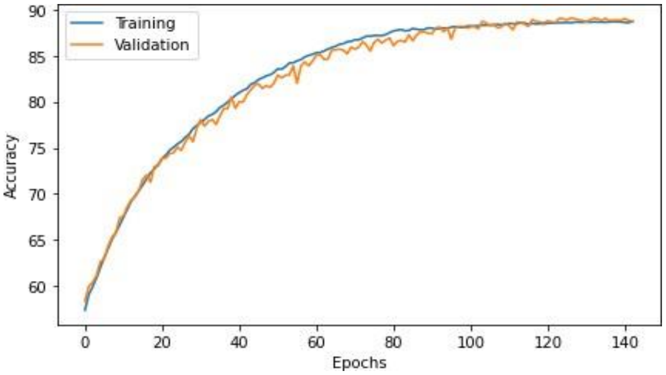


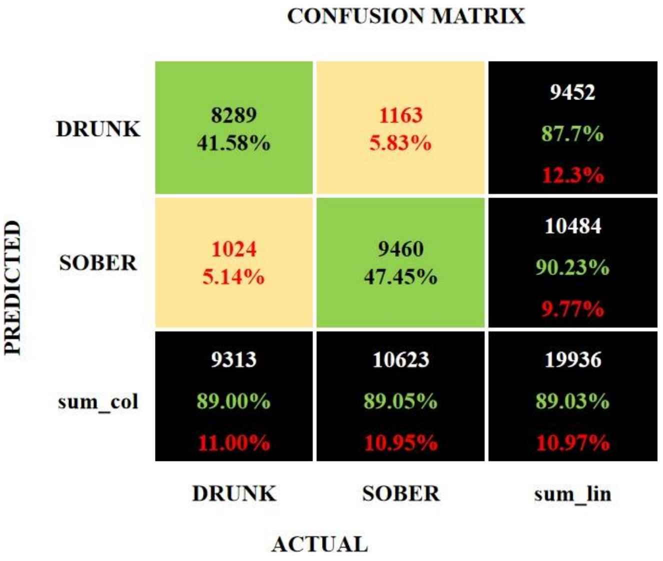
Figure 8. Training and Validation, Accuracy of the model with master node

Results		
MODEL	Drunk classifier	Drunk classifier -with master Node
Training Accuracy	86.69±2.29	**88.7±2.21**
Training Loss	0.304±0.04	**0.26±0.04**
Validation Accuracy	86.73±2.6	**89.09±2.46**
Validation Loss	0.302±0.051	**0.25±0.05**
Test Accuracy	86.4±2.4	**89.5±2.51**
Test Loss	0.303±0.04	**0.252±0.54**

Hyperparameter	
Learning Rate	0.001
Epochs	200
Patience	10
Batch Size	64
Train : Test : Val	7:2:1
Seed for data shuffling	3

Table I
RESULTS OBTAINED FROM THE TWO MODEL. RESULTS IN RED REPRESENT THE BEST RESULT OBTAINED.

From the confusion matrix, it is clear that our model is not inclined towards a certain class as both the True Positive Rate and True Negative rate are greater than 0.89. Moreover, the data was not skewed, hence the accuracy itself is proof that the model is able to recognize drunk and sober people using their face image.



CONFUSION MATRIX

PREDICTED \ ACTUAL	DRUNK	SOBER	sum_lin
DRUNK	8289 41.58%	1163 5.83%	9452 87.7% 12.3%
SOBER	1024 5.14%	9460 47.45%	10484 90.23% 9.77%
sum_col	9313 89.00% 11.00%	10623 89.05% 10.95%	19936 89.03% 10.97%

Figure 9. Confusion Matrix of the model with master node in input on testing set

VII. CONCLUSIONS

Based on our results, our model can correctly classify drunken and sober people from their facial photographs with an average accuracy of 88%. The proposed study develops a GNN-based model for intoxication detection in addition to a large graph database of drunken and sober faces. The model is independent of the skin color of the sample that was mentioned as a limitation in previous works. It’s important to note that YouTube videos provide these data, so the model could fail if an individual has an untrained tolerance to alcohol.

REFERENCES

[1] Mondal, D., 2021. ‘Drunk driving led to 38,000 road mishaps in three years’ - The Sunday Guardian Live. [online] The Sunday Guardian Live. Available at: ¡https://www.sundayguardianlive.com/news/drunk-driving-led-38000-road-mishaps-three-years¿ [Accessed 28 August 2021].

[2] Steele, C. M., Josephs, R. A. (1990). Alcohol myopia: Its prized and dangerous effects. American Psychologist, 45(8), 921–933. https://doi.org/10.1037/0003-066X.45.8.921

[3] Wu, C., Tsang, K., Chi, H. and Hung, F., 2016. A Precise Drunk Driving Detection Using Weighted Kernel Based on Electrocardiogram. Sensors, 16(5), p.659.

[4] G. Koukiou, G. Panagopoulos and V. Anastassopoulos, ”Drunk person identification using thermal infrared images,” 2009 16th International Conference on Digital Signal Processing, 2009, pp. 1-4, doi: 10.1109/ICDSP.2009.5201249.

[5] Capito, E., Lautenbacher, S. and Horn-Hofmann, C., 2017. Acute alcohol effects on facial expressions of emotions in social drinkers: a systematic review. Psychology Research and Behavior Management, Volume 10, pp.369-385.

[6] Maity, Suman Mullick, Ankan Ghosh, Surjya Kumar, Anil Dhamnani, Sunny Bahety, Sudhanshu Mukherjee, Animesh. (2018). Understanding Psycholinguistic Behavior of predominant drunk texters in Social Media.

[7] C. Willoughby, I. Banatoski, P. Roberts and E. Agu, ”DrunkSelfie: Intoxication Detection from Smartphone Facial Images,” 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, pp. 496-501, doi: 10.1109/COMPSAC.2019.10255.

[8] V. Mehta, S. S. Katta, D. P. Yadav and A. Dhall, ”Dif dataset of perceived intoxicated faces for drunk person identification,” 2019 International Conference on Multimodal Interaction, pp. 367-374, 2019.

[9] V. Kazemi and J. Sullivan, ”One millisecond face alignment with an ensemble of regression trees,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867-1874, doi: 10.1109/CVPR.2014.241.

[10] Sagonas, Christos Tzimiropoulos, Georgios Zafeiriou, Stefanos Pantic, Maja. (2013). 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. 397-403. 10.1109/ICCVW.2013.59.

[11] Sun, Shuli Sui, Jie Chen, Bin Yuan, Mingwu. (2013). An Efficient Mesh Generation Method for Fractured Network System Based on Dynamic Grid Deformation. Mathematical Problems in Engineering. 2013. 10.1155/2013/834908.

[12] Scarselli, F., Gori, M., Tsoi, A., Hagenbuchner, M. Monfardini, G. 2009, ’The graph neural network model’, IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80.

[13] Zhang, H., Lu, G., Zhan, M. and Zhang, B., 2021. Semi-Supervised Classification of Graph Convolutional Networks with Laplacian Rank Constraints. Neural Processing Letters,.

[14] Chen, Yihao Tang, Xin Qi, Xianbiao Li, Chun-Guang Xiao, Rong. (2020). Learning Graph Normalization for Graph Neural Networks.