This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society to the acceptance and publication.

Machine Learning Models for Drowsiness Detection

Harshit Meda
Department of Mechanical Engineering
Indian Institute of Technology, Ropar
Rupnagar, India
2018meb1227@iitrpr.ac.in

Janapareddy Mohan Padmanabha Ganesh
Department of Mechanical Engineering
Indian Institute of Technology, Ropar
Rupnagar, India
2018meb1230@iitrpr.ac.in

Ashish Sahani
Center for Bio-Medical Engineering
Indian Institute of Technology, Ropar
Rupnagar, India
ashish.sahani@iitrpr.ac.in

***Abstract*— Road crashes and other accidents have become the common cause of fatalities and injuries in the human world. According to data from the World Health Organisation (WHO) in 2015, nearly 1.25 million people died worldwide due to road accidents. Driver fatigue is a significant factor in many road accidents. A sleepy driver is more dangerous than a driver driving at high speeds as he is victim of less sleep. Many researchers and manufactures are trying to solve this using various technologies. Driver drowsiness detection can help prevent a huge number of sleep induced road accidents. We will be using computer vision algorithms to extract facial features such as eye closure and yawning, followed by machine learning techniques to effectively detect driver state. We will be comparing multiple machine learning models and will be using the most effective one to develop a real-time drowsiness detector. This system will warn the driver if it detects a drowsy state, hence preventing any harm that may have been caused to the driver and the passengers otherwise.**

Keywords—Drowsiness Detection, Machine Learning, KNN, Logistic Regression, Decision Tree Classifier, Naïve Bayes Classifier, LSTM, CNN

I. INTRODUCTION

According to the statistics releases by National Highway Traffic Safety Administration (NHTSA), every year about 100,000 crashes are caused due to drowsy driving. These crashes resulted in more than 1550 fatalities, 71,000 injuries and $12.5 billion money losses [1]. According to a survey, half of American adult's report consistently that they have driven drowsy. These startling figures show that how impactful the drowsy driving is.

It is an industrial and academic challenge to develop drowsiness detection technologies. In the Automotive industry, Volvo developed Driver Alert Control which alerts the driver when he is drowsy using a camera mounted to lane departure warning system (LDWS). Mercedes Benz has developed Attention Assist system, which warns the driver before it's too late if he starts to become drowsy. Similarly, many other big automobile companies like Audi, Hyundai, Mazda etc. have developed various technologies that helps prevent accidents caused due to drowsy driving.

Autonomous systems designed to analyse driver exhaustion and detect driver drowsiness can be an integral part of the future intelligent vehicle to prevent accidents caused by sleep. Multiple techniques have been developed in recent years.

1. Driver operation and vehicle behaviour can be implemented by monitoring the steering wheel movement, accelerator or brake patterns, vehicle speed, lateral acceleration, and lateral displacement.
2. Another set of techniques focuses on monitoring of physiological characteristics of the driver such as heart rate, pulse rate, and Electroencephalography.
3. The third set of techniques is based on computer vision systems which can recognize the facial appearance changes occurring during drowsiness.

The 1st method is limited by the type and model of the car. The 2nd method though more accurate results it has widely been downplayed due to impracticality in deploying it on a large scale and its intrusive nature. The 3rd method is a very promising one and we have followed this route and developed a model on the same.

II. DATA SOURCE

We used UTA Real -Life Drowsiness Dataset created by a research team from University of Texas for multi-stage drowsiness detection [2]. The RLDD data source consists of 10 min videos of 60 participants in alert and drowsy states. All the participants are over 18 years of age and they also belong to different ethnicities. Since the data set was too large, we took a sample of 25 people. From the total data set of 50 videos, we extracted 1 frame every second from the 2-minute mark.

Fig 1. Sample frames from the UTA-RLDD dataset in the alert and drowsy states

III. FEATURES

We extract the face landmarks from each frame using the extract_face_landmarks function from the mlxtend function. There is a total of 68 landmarks per frame but as we are only extracting features of the eyes and mouth, we decided to keep the landmarks 37–68.

2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) | 978-1-7281-9539-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2MTC50364.2021.9460017

978-1-7281-9539-1/21/$31.00 ©2021 IEEE

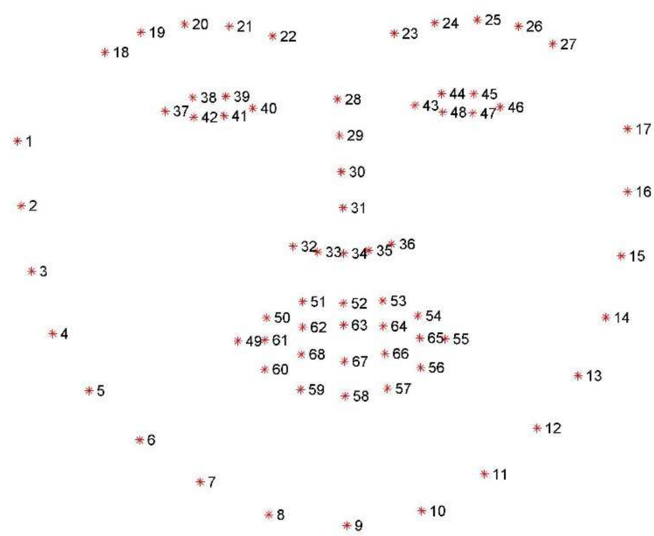
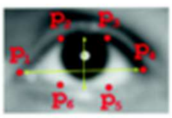


Fig 2: Face Landmarks as detected by extract_face_landmarks [2]

We will first extract 4 facial features namely

A. Eye Aspect Ratio(EAR)

EAR is the length to width ratio of the eyes. We know that when an individual is drowsy, their eyes are likely to contract, and the frequency of blinks increases drastically. Based on this hypothesis, we expected our model to predict the state as drowsy if the eye aspect ratio for an individual drops below a critical threshold. This threshold will be decided while training the model.



$$EAR = \frac{|\,p_2 - p_6| + |\,p_3 - p_5\,|}{2|\,p_1 - p_4\,|} \quad (1)$$

B. Mouth Aspect Ratio(MAR)

Like the EAR, the MAR is the length to width ratio of the mouth. We hypothesize that as an individual becomes drowsy, they are likely to yawn, making their MAR to be higher than in the alert state.

$$MAR = \frac{|EF|}{|AB|} \quad (2)$$

C. Pupil Circularity (PUR)

PUC is a measure complementary to the EAR. Like the EAR, a drowsy person is likely to have a smaller value of PUC as compared to an alert person.

$$Circularity = \frac{4 * \pi * Area}{perimeter^2} \quad (3)$$

$$\begin{aligned} Perimeter = {} & |\,p_1 - p_2| + |\,p_2 - p_3| \\ & + |\,p_3 - p_4| + |\,p_4 - p_5| \\ & + |\,p_5 - p_6| + |\,p_6 - p_1| \end{aligned} \quad (4)$$

$$Area = \left(\frac{|p_2 - p_5|}{2}\right)^2 * \pi \quad (5)$$

D. Mouth aspect ratio over Eye aspect ratio (MOE)

MOE is the ratio of the MAR to the EAR. Depending on the state of the person, the MOE moves in the opposite direction and this leads to an amplification of the relative change in the two values. i.e. an alert person has low MAR and high EAR, whereas, for a drowsy person, it is the other way round. Hence, the MOE is more responsive to change.

$$MOE = \frac{MAR}{EAR} \quad (6)$$

IV. Solutions and Methodology

With the 4 features extracted, we train different machine learning models to predict the state of the person in the frame as alert or drowsy. We will be performing cross-validation along with hyperparameter tuning to maximize accuracy and precision.

The models used are:

A. K Nearest Neighbours

The k-nearest-neighbour classifier uses the Euclidean distance between a test sample and the given training samples to find the nearest observations to the test sample and predict its class. The Euclidean distance between sample xi and xl (l=1,2,…,n) with p features is defined as:

$$d(x_i\,,x_l\,) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \cdots + \left(x_{ip} - x_{lp}\right)^2} \quad (7)$$

Let xi be a training sample and x be a test sample and let ω be the true class of a training sample and ωt be the predicted class for a test sample (ω,ωt=1,2,…, Ω). Here, Ω is the total number of classes.

During training, only the true class ω of the training sample is used to train the classifier, while during testing we predict the class ωt of all the test samples [4]. KNN is a "supervised" machine learning classification method at it uses the class labels of the training data to predict the class of test data.

The predicted class of test sample x is set equal to the most frequent true class among k nearest training samples. This forms the decision rule

D:x→ωt

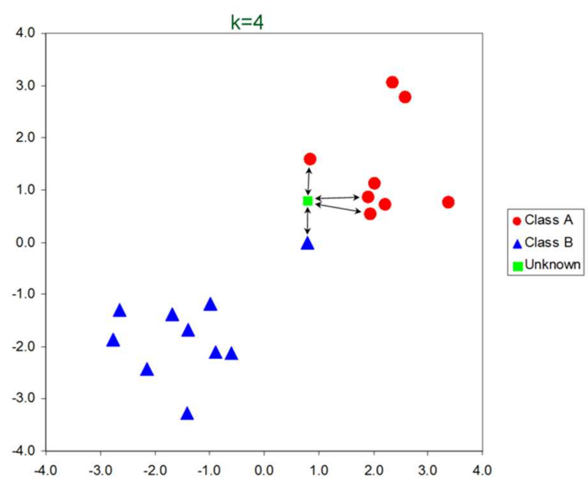


Fig 3: kNN classifier with k=4, Among the 4 nearest neighbours of the unknown sample, the most frequent class is A, thus it is categorized as class A [4]

B. Logistic Regression

Logistic Regression is a binary classifier predicting the target categorically dependent variables and ideal for detection of alert or drowsy state. The sigmoid function is used to predict the probability of a state of 1. Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{8}$$

Model:

Output = 0 or 1
Hypothesis => Z = WX + B
hΘ(x) = sigmoid (Z)
hΘ(x) = P(Y=1 |X; theta)
P(Y=1 |X; theta) + P(Y=0 |X; theta) = 1

Cost Function:

$$Cost(h_\theta(x), y) = -ylog\left(h_\theta(x)\right) - (1-y)\log\left(1-h_\theta(x)\right) \tag{9}$$

It must be noted that mean square error is not the cost function for logistic regression as the sigmoid function is non-convex. This implies that gradient descent will not converge to the global minima but the local minima if Mean Squared Error is employed as the cost function.

C. Decision Tree Classifier

A decision tree is a Machine Learning Algorithm that resembles a tree where each node shows a feature (attribute), each link (branch) shows a decision (rule) and each leaf shows an outcome (categorical or continuous value) [5]. It also closely resembles the human level thinking; hence it is really handy for easy interpretation of data.

TABLE I Metrics used to train the Decision Tree Classifier

Metrics	Equation
Information Gain	$I(p,n) = \left(-\frac{p}{p+n}\right)\log_2\left(\frac{p}{p+n}\right) - \left(\frac{n}{p+n}\right)\log_2\left(\frac{n}{p+n}\right)$
Gain Ratio	$Gain\ Ratio = I(p,n) - E(A)$ I(p,n) = Information before splitting E(A) = Information after splitting
Gini Index	$GI = 1 - \sum_{j=1}^{c} p_j^2$

According to the values of the splitting attribute, the training data are partitioned into several subsets. Until all instances in a subset belong to the same class, the Decision Tree algorithm proceeds recursively [6]. This however can lead to overfitting of the model and to prevent this, we can restrict the depth of the decision tree. The Gini Index represents the homogeneity of data at each node and this value is 0 for a uniform dataset. Thus, the Gini index decreases with successive split in data at each node (p_j in the formula for Gini index is the probability of observation to belong to class j out of c classes).

D. Naive Bayes Classifier

Naive Bayes classifier is supervised machine learning algorithm, which uses the Bayes theorem of Probability. The input features are assumed to be statistically independent in this algorithm [7]. In probability theory and statistics, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event [8]. The simple form of Bayes' theorem is given as the following equation:

For any two events A and B, provided P(B) ≠ 0,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{10}$$

Naive Bayes Classifier assigns the most likely class (either drowsy or non- drowsy) for a given feature vector. Learning such classifiers can be greatly simplified by assuming that features are independent given class, that is, P(X|C) = π P(Xi|C) , where X = (X1, X2, . . ., XN) is a feature vector and C is a class (either drowsy or non-drowsy) [7]. Our feature vector, which is X = (MOE, PAR, EAR, MAR), is input vector to the model.

E. Long Short-Term Memory (LSTM) Networks

Long short-term memory networks are artificial networks which deal with sequential data. These are special type of Recurrent Neural networks (RNN), which are well-suited for classifying time series data. In RNN's, inputs are related on each other whereas in other neural networks they are independent. We chose LSTM model as there will be no gradient-vanishing problems and it can also study long sequences. A LSTM network consists of three gates and they are Forget gate, input gate and output gate.

The features which are extracted from each video frame is passed into the LSTM model. The features are divided into batches of data of size 5 before feeding to the network. Then, each batch of data is sent through the first LSTM layer which consists of 128 neurons which returns the sequences. The next layer consists of 100 neurons using SoftMax activation. The next LSTM layer, which is the output layer, consists of single neuron using SoftMax activation indicating '0' if the person is non-drowsy and '1' if the person is drowsy.

TABLE II LSTM Parameters

Activation Function	SoftMax
Optimizer	Adam
Loss Function	Mean Squared Error
Number of Epochs	100
Timestep	5

F. Deep CNN Based Drowsiness Detection System

1) Step 1 Extracting Features from Video Frames:

As it is not possible to train the whole neural network on videos, we have extracted one frame every second starting from the 2 minute-mark. With help of dib library landmark coordinates were extracted and from these coordinates the features were extracted. Features extraction is kind of dimensionality reduction where only important features are extracted. It makes easy for the model to predict when you extract important features of an image and pass into the model instead of passing a whole image array.

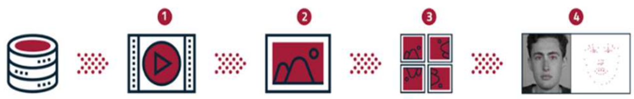


Fig 4: Video pre-processing outline [9]

2) Step 2 Normalisation of Features:

As different people have different sizes of eyes and mouth, features like MOE and PUC can vary a lot so it is required to normalise their features. After extracting the features from the frames, the four features were normalised.

3) Step 3 Drowsiness Detection using CNN:

We have built a one-dimensional CNN. Our CNN model consists of 5 layers which includes one convolutional layer with 32 filters with RELU activation, 1 flattened layer, 2 fully connected dense layers with 16 and 8 neurons respectively, and an output layer which consists of two neurons one predicts drowsy and the other to predict non-drowsy with sigmoid activation. The normalised features were passed as input to the CNN.

TABLE III CNN Parameters

Activation Function	RELU /Sigmoid
Loss Function	Binary Cross Entropy
Number of epochs	100
Optimizer	Adam
Learning Rate	0.00001

Here, instead of passing features as input, video frames can also be passes as an array input to the two-dimensional CNN and train the model.

V. Results

After training the different models, these are the results obtained:

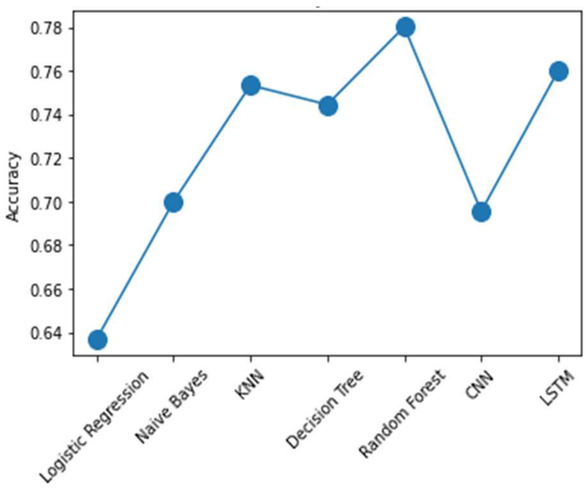


Fig 5: Accuracy vs Model

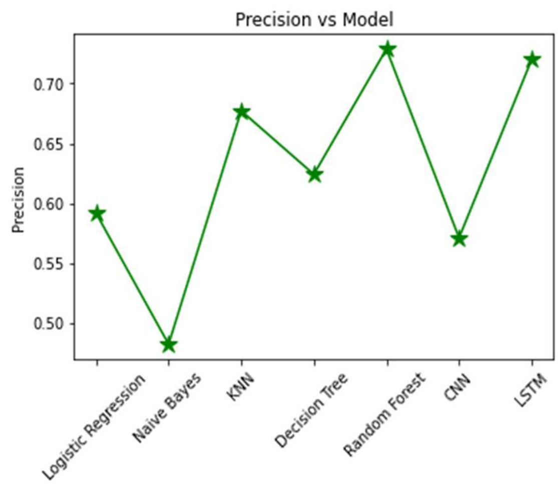


Fig 6: Precision vs Model

From the Accuracy and Precision of all the models trained, the Random Forest Classifier is the superior one. The LSTM model which works well on time-series data with a feedback loop has performed almost as good as the Random-Forest Tree.

Here are some of live images, which are taken through webcam, detected as drowsy and alert by the Random Forest Classifier.

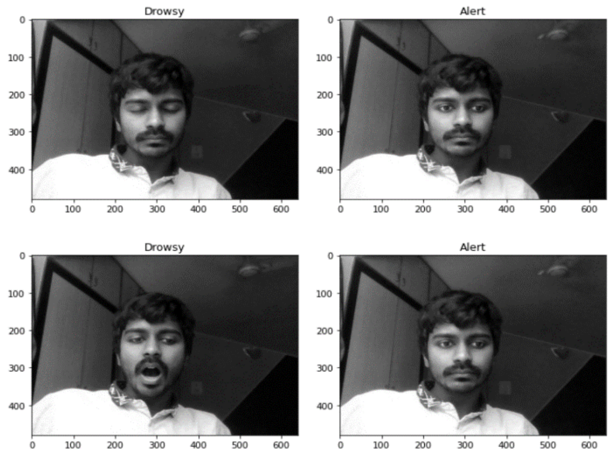


Fig 7: Live Detection of Drowsiness

A. Feature Importance

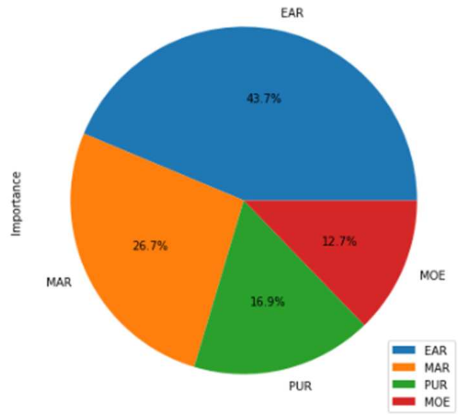


Fig 8: Feature Importance

From the Decision Tree Classifier, we were able to find the feature importance associated with the 4 input features. From the data obtained, it can be concluded that the Eye Aspect Ratio (EAR) is the single most significant indicator of Drowsiness. This is a very intuitive result as closed eyes are a sure indicator of drowsiness as opposed to an open mouth. Also, the change in pupil circularity (PUR) though significant, has lesser importance as compared to the MAR. The least significant feature turns out to be MOE, contrary to our assumption that it will be more responsive to change in state as hypothesized initially.

VI. Conclusion

We have successfully developed an algorithm for driver drowsiness detection using machine learning and computer vision. We used hand-engineered features detecting driver drowsiness based on human facial expressions. We were able to come up with an effective solution, with little inconvenience caused to the driver in the form of close to body sensors and instruments. Also, it will work efficiently irrespective of the model and age of the car. We developed an algorithm to detect a drowsy state in real-time.

VII. FUTURE WORK

Although the proposed machine-learning based driver drowsiness detection can detect drowsiness with reasonable accuracy, still there is a scope for improvement in its performance. Drowsiness induces involuntary rolling or falling of the driver's head which was not considered in the developed model. Also, to ensure that we can detect drowsiness during nocturnal hours, an I.R. LED-based tracker can be employed to help detect drowsiness in all illumination conditions.

REFERENCES

[1] United States Department of Transportation, "Drowsy Driving," *National Highway Traffic Safety Administration,* 2020. [Online]. Available: https://www.nhtsa.gov/risky-driving/drowsy-driving. [Accessed: July 2, 2020].

[2] Zhong.G, "Drowsiness Detection With Machine Learning," *Medium*, Dec.14, 2019. [Online]. Available: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a. [Accessed: June 10, 2020].

[3] R.Ghoddoosian, M.Galib and V.Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0-0, 2019.

[4] Peterson E Leif, "K-nearest neighbor," *Scholarpedia* , vol.4, no.2 , pp. 1183 - 1186, 2009.

[5] X.Zhang and S.Jiang, "A Splitting Criteria Based on Similarity in Decision Tree Learning," *Journal of software*, vol.7, no.8, pp. 1775-1782, August 2012.

[6] S.D. Jadhavand H.P. Channe , "Efficient recommendation system using decision tree classifier and collaborative filtering," *Int. Res. J. Eng Technol,* vol.3, no.8, pp. 2113-2118, August 2016.

[7] A.C.Bahnsen, "Machine Learning Algorithms Explained-Naive Bayes Classifier", *Appgate*, Mar.9,2018. [Online]. Available: https://blog.easysol.net/machine-learning-algorithms-4/. [Accessed: July 2, 2020].

[8] Wikipedia contributors, "Bayes' theorem," *In Wikipedia The Free Encyclopedia, Jun.29,2020.* [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=966530798. [Accessed: July 3,2020].

[9] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen and K. Barkaoui, "Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, 2020, pp. 237-242.

[10] I.Rish, "An empirical study of the naive Bayes classifier, " *IJCAI 2001 workshop on empirical methods in artificial intelligence,* vol.3, no.22, pp. 41-46, 2001.

[11] K. Dwivedi, K. Biswaranjan and A. Sethi, "Drowsy driver detection using representation learning," *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, 2014, pp. 995-999, doi: 10.1109/IAdCC.2014.6779459.

[12] R.Jabbar, et al, "Real-time driver drowsiness detection for android application using deep neural networks techniques," *Procedia computer science 130* , vol. 130, pp. 400-407, 2018.