Synopsis on
Drowsy Driver Detection

Department of Information Technology

By

Sagnik Sinha (Roll no. 12200219011)

Dhiraj Ojha (Roll no. 12200219017)

Alok Kumar Jha (Roll no. 12200219021)

Rohit Kumar Choubey (Roll no. 12200219025)

Under the guidance of

Dr. Ranjit Ghoshal

St. Thomas’ College of Engineering and Technology

Affiliated to

Maulana Abul Kalam Azad University of Technology, West Bengal

We are submitting the synopsis on "Drowsy Driver Detection" as a part of my final year seventh semester project under the guidance of Dr. Ranjit Ghoshal.

Sagnik Sinha **s** Dhiraj Ojha Alok kumar Jha Rohit Choubey

**3333333333333333333333333333333333333333333333333333333333333333333333333333333333333
3333333333333333333333333333333333333333333333333333333333333333333333333333333333333
3333333333333333333333333333333333333333333333**

<Signature> **<Signature>** **<Signature>** **<Signature>**

Sagnik Sinha **Dhiraj Ojha** **Alok Kumar Jha** **Rohit Kumar Choubey**

Guide's Signature (for approval)

Dr. Ranjit Ghoshal

Departmental Stamp

Vision: To promote the advancement of learning in Information Technology through research-oriented dissemination of knowledge which will lead to innovative applications of information in industry and society.

Mission: To incubate students, grow into industry ready professionals, proficient research scholars and enterprising entrepreneurs.

To create a learner-centric environment that motivates the students in adopting emerging technologies of the rapidly changing information society.

To promote social, environmental and technological responsiveness among the members of the faculty and students.

PEO:

PEO1: Exhibit the skills and knowledge required to design, develop and implement IT solutions for real life problems.

PEO2: Excel in professional career, higher education and research.

PEO3: Demonstrate professionalism, entrepreneurship, ethical behavior, communication skills and collaborative team work to adapt the emerging trends by engaging in lifelong learning.

PO : Project Mapping with Program Outcomes

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
3	3	2	3	3	3	-	3	2	3	3	3

Enter correlation levels 1,2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Justification:

1. Engineering knowledge: Application of the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems is required in this project, hence, it satisfies PO1.
2. Problem analysis: Identifying, formulating, reviewing research literature, and analyzing complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences is highly required in this project, hence, it satisfies PO2.
3. Design/development of solutions: Designing solutions for complex engineering problems and designing system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations is required to be done in this project, thus, PO3 stands applicable.
4. Conduct investigations of complex problems: Using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions is required to a great extent. PO4 stands applicable.
5. Modern tool usage: Appropriate modern tools and methods are used. Hence PO5 stands applicable.
6. The engineer and society: Application of reasoning informed by the contextual knowledge to assess cultural issues and the consequent responsibilities relevant to the professional engineering practice is recognized in this project, hence, PO6 stands applicable.
7. Environment and sustainability: This project's impact is more cultural and technology based, hence, PO7 is not applicable.
8. Ethics: Applying ethical principles and commitment to professional ethics and responsibilities and norms of the engineering practice is required. PO8 is applicable.
9. Individual and team work: Functioning effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings: this is applicable here.
10. Communication: Communicating effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, making effective presentations, and giving and receiving clear instructions: PO10 is substantially applicable in this project.
11. Project management and finance: Demonstrating knowledge and understanding of the engineering and management principles and applying these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments: this is highly applicable in this project, hence PO11 has high correlation level (3)
12. Life-long learning: Recognizing the need for, and having the preparation and ability to engage in independent and life-long learning in the broadest context of technological change: this is relevant with respect to this project, hence, PO12 has high correlation level of 3.

PSO : Project Mapping with Program Specific outcomes

PSO1	PSO2	PSO3
3	2	3

Enter correlation levels 1,2 or 3 as defined below:

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)**

Justification:

This project requires me to apply programming knowledge to build an efficient and effective solution of the problem with an error free, well documented and reusable code, user friendly interface and well-organized database. Hence, the project substantially satisfies PSO1.

Creation of multimedia enabled web solutions using information in different forms for business, education and the society at large isn't applicable in this project, hence, PSO2 stands irrelevant/ not applicable.

Understanding and analyzing a big complex problem and decomposing it into relatively smaller and independent modules algorithmically is done in this project, hence, PSO3 is satisfied substantially.

Index Page

Contents

1. Introduction

1.1 Problem Statement: Drowsy Driver Detection System using Machine Learning.

1.2 Problem Definition: The driver of a vehicle is categorized as drowsy or not based on classification algorithms.

1.3 Objective: The aim of this project is to build a model which correctly recognizes the driver of a vehicle as fit to drive or not using Machine Learning. This project extends to building an interface which automobiles can use to detect drowsiness in drivers and prevent accidents.

1.4 Literature Survey:

1. E.Mary Bearly, R. Chitra, "Automatic Drowsiness Detection for Preventing Road Accidents", 2021 Smart Technologies, Communication and Robotics (STCR)
2. Al Mamun Mizan, A. Z. M. Tahmidul Kabir, Nadim Zinnurayen, Tawsif Abrar, Akib Jawad Ta-sin, Mahfuzar, "The Smart Vehicle Management System for Accident Prevention by Using Drowsiness, Alcohol, and Overload Detection", 2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), pp.173-177, 2020.

1.5 Brief Discussion on Problem:

Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads.Some of the current systems learn driver patterns and can detect when a driver is becoming drowsy.[1]

2. Concepts and Problem Analysis

This project is an application of Machine Learning. Machine Learning is the use of algorithms so as to compel a system to learn and adapt on its own from previous experiences and draw inferences from patterns in data. A general machine learning block diagram is illustrated in Fig 1. A typical machine learning problem starts with data acquisition followed by pre-processing and preparation of the data suitably. The prepared data is used for extracting features and the dataset is split into training and testing sets. The machine learning model is trained using this training set and the trained model is tested using the test set. This is followed by model evaluation which involves calculating the accuracy and precision of the trained model. This evaluation can be further used to fine tune the model until a desired accuracy is achieved that can be deployed.[1]

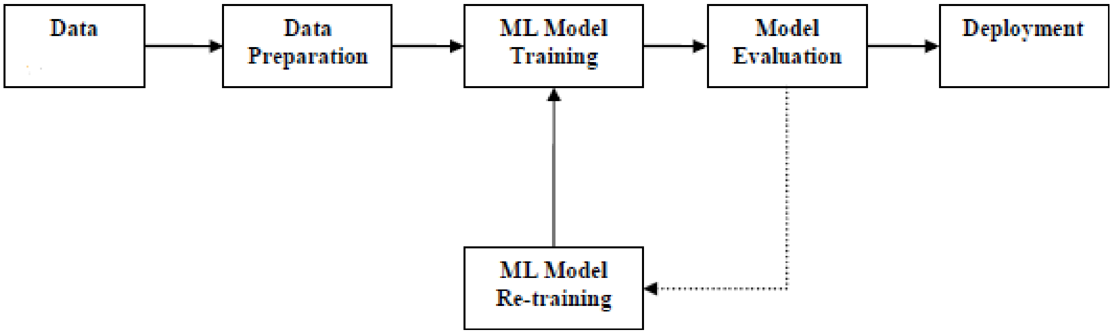


Fig. 1

2.1 Dataset:

The dataset used in this project is the MRL Eye Dataset. This dataset contains infrared images in low and high resolution, all captured in various lightning conditions and by different devices. The dataset is suitable for testing several features or trainable classifiers. In order to simplify the comparison of algorithms, the images are divided into several categories, which also makes them suitable for training and testing classifiers.

Dataset consists of 84,898 images taken across 37 different persons (33 men and 4 women). It contains all types of images like open, closed, with glasses, without glasses, and in different lighting conditions.[2]

2.2. Feature Extraction:

Digital image processing is the use of a digital computer to process digital images through an algorithm. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Python provides lots of libraries for image processing, including −

- OpenCV − Image processing library mainly focused on real-time computer vision with application in wide-range of areas like 2D and 3D feature toolkits, facial & gesture recognition, Human-computer interaction, Mobile robotics, Object identification and others.
- Numpy and Scipy libraries − For image manipulation and processing.
- Sckikit − Provides lots of algorithms for image processing.
- Python Imaging Library (PIL) − To perform basic operations on images like create thumbnails, resize, rotation, convert between different file formats etc.[3]

2.2.1 Reading the data using Python: OpenCV is an image processing library mainly focused on real-time computer vision with application in wide-range of areas like 2D and 3D feature toolkits, facial & gesture recognition, Human-computer interaction, Mobile robotics, Object identification and others.[3]

2.2.2 Visualizing the data: The image is resized to 224 x 224 , converted to grayscale and represented as a graphical plot for feature extraction. Viewed using plt.imshow() and plt.show().

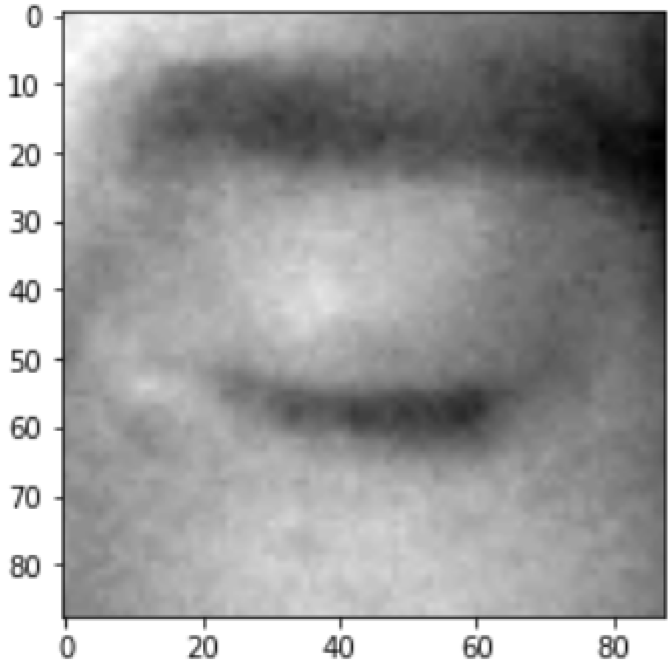


Fig. 2 (Sample Data)

2.2.3 Features Used:

Convolutional Neural Network

A neural network which was designed to process multi-dimensional data like image and time series data is called a convolutional neural network (CNN). It includes feature extraction and weight computation during the training process. The name of such networks is obtained by applying a convolution operator which is useful for solving complex operations. The true fact is that CNNs provide automatic feature extraction, which is the primary advantage. The specified input data is initially forwarded to a feature extraction network, and then the resultant extracted features are forwarded to a classifier network. The feature extraction network comprises loads of convolutional and pooling layer pairs. Convolutional layer consists of a collection of digital filters to perform the convolution operation on the input data. The pooling layer is used as a dimensionality reduction layer and decides the threshold. During backpropagation, a number of parameters are required to be adjusted, which in turn minimizes the connections within the neural network architecture.[4]

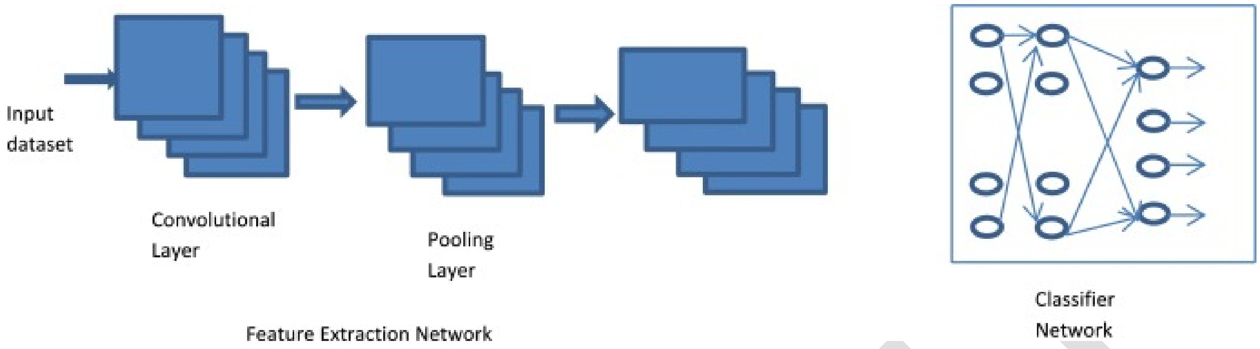


Fig. 3 (Feature Extraction Network)

2.3 Classification:

In machine learning , classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. A model will use the training dataset and will calculate how to best map examples of input data to specific class labels. As such the training dataset must be sufficiently representative of the problem and have many examples of each class label.

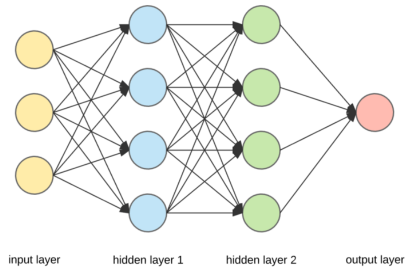
Types of classification techniques:

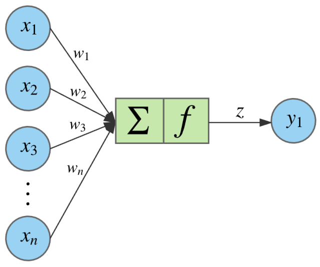
1. K Neighbors Classification
2. Logistic Regression
3. Support Vector Machines
4. Decision Trees
5. Artificial Neural Network

2.3.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are multi-layer fully-connected neural nets that look like the figure below. They consist of an input layer, multiple hidden layers, and an output layer. Every node in one layer is connected to every other node in the next layer. We make the network deeper by increasing the number of hidden layers.

2.3.1.1 Working:





A given node takes the weighted sum of its inputs, and passes it through a non-linear activation function. This is the output of the node, which then becomes the input of another node in the next layer. The signal flows from left to right, and the final output is calculated by performing this procedure for all the nodes. Training this deep neural network means learning the weights associated with all the edges.The equation for a given node looks as follows. The weighted sum of its inputs passed through a non-linear activation function. It can be represented as a vector dot product, where *n* is the number of inputs for the node.

$$z = f(x \cdot w) = f\left(\sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ z \in d_{1\times 1}$$

Bias is an input to all the nodes and always has the value 1. It allows to shift the result of the activation function to the left or right. It also helps the model to train when all the input features are 0. If this sounds complicated right now you can safely ignore the bias terms. For completeness, the above equation looks as follows with the bias included.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ b \in d_{1\times 1},\ z \in d_{1\times 1}$$

So far we have described the *forward pass*, meaning given an input and weights how the output is computed. After the training is complete, we only run the forward pass to make the predictions. But we first need to train our model to actually learn the weights, and the training procedure works as follows:

- Randomly initialize the weights for all the nodes. There are smart initialization methods which we will explore in another article.

- For every training example, perform a forward pass using the current weights, and calculate the output of each node going from left to right. The final output is the value of the last node.
- Compare the final output with the actual target in the training data, and measure the error using a *loss function.*
- Perform a *backwards pass* from right to left and propagate the error to every individual node using *backpropagation*. Calculate each weight's contribution to the error, and adjust the weights accordingly using *gradient descent*. Propagate the error gradients back starting from the last layer.

Backpropagation with gradient descent is literally the "magic" behind the deep learning models. It's a rather long topic and involves some calculus, so we won't go into the specifics in this applied deep learning series. For a detailed explanation of gradient descent refer here. A basic overview of backpropagation is available here. For a detailed mathematical treatment refer here and here. And for more advanced optimization algorithms refer here.

If we take a classification problem as an example, we want to separate out the classes by drawing a decision boundary. The input data in its given form is not separable. By performing non-linear transformations at each layer, we are able to project the input to a new vector space, and draw a complex decision boundary to separate the classes.

Let's visualize what we just described with a concrete example. Given the following data we can see that it isn't linearly separable.

So we project it to a higher dimensional space by performing a non-linear transformation, and then it becomes linearly separable. The green hyperplane is the decision boundary.

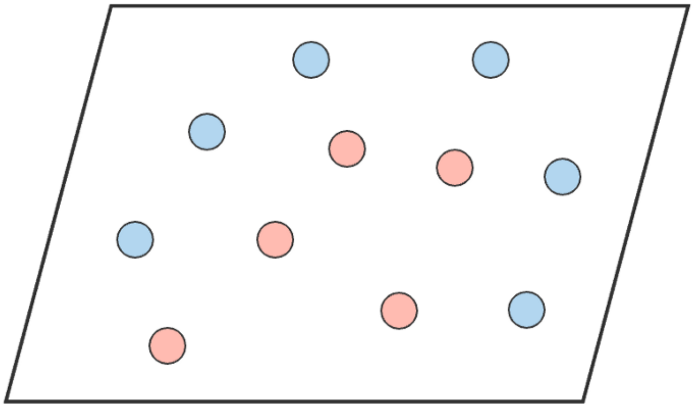


Fig. 5 (Feature Space)

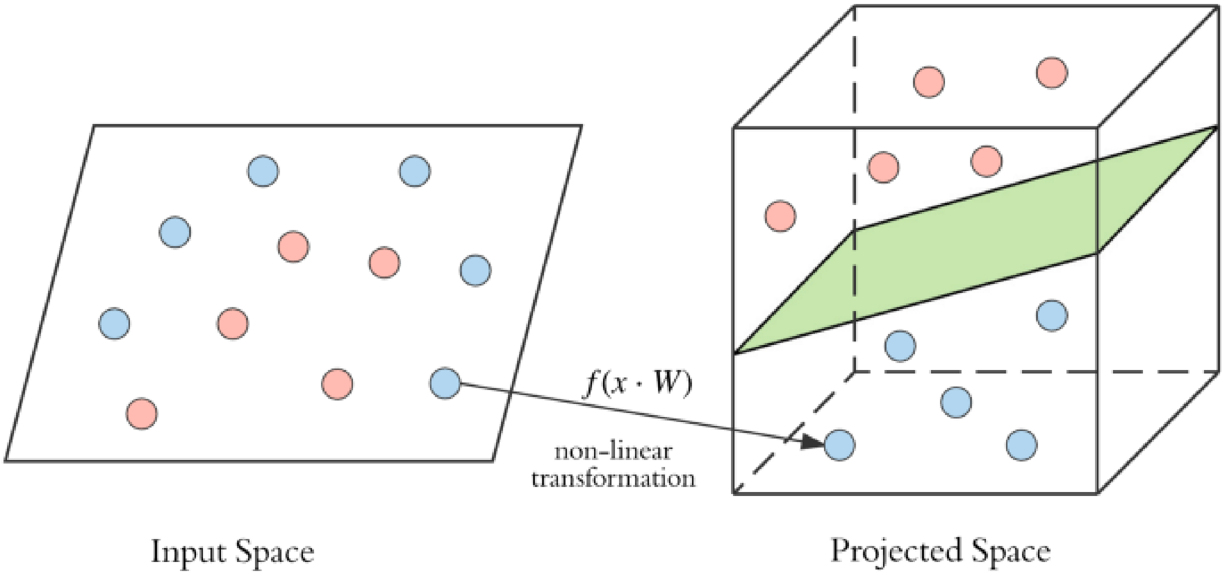


Fig. 6 (Projected Feature Space)

This is equivalent to drawing a complex decision boundary in the original input space.

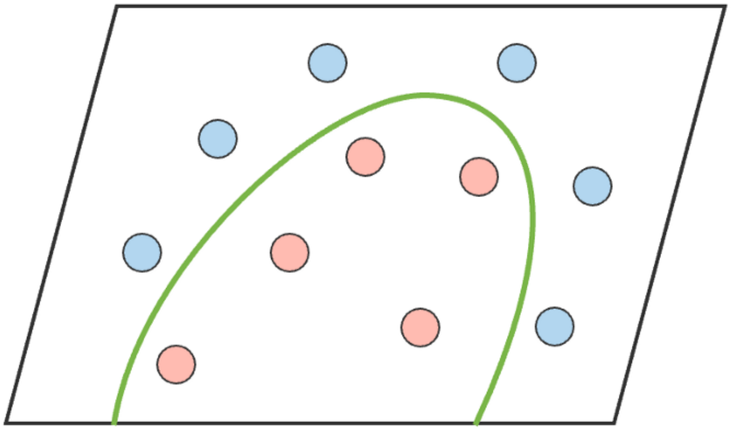


Fig. 7 (Non-linear Decision Boundary)

So the main benefit of having a deeper model is being able to do more non-linear transformations of the input and drawing a more complex decision boundary.[5]

2.3.1.2 Drowsiness Detection System :

Requirements

●Python

●OpenCV: OpenCV is a great tool for image processing and performing many computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, object tracking, and many more tasks.

●TensorFlow: Tensorflow is a free and open-source library, developed by the Google Brain team for machine learning and artificial intelligence. Tensorflow has a particular focus on the training and inference of deep neural networks.

●Keras: Keras is an open-source software library and it provides a Python interface for artificial neural networks. Keras is more user-friendly because it is an inbuilt python library.[6]

Methodology

The methodology of this project is the first video is captured using a webcam and from the video first face is detected using the Haar Cascade algorithm and then the eyes are detected. Then we use our deep learning model which is built using transfer learning to know the status of the eye. If it is an open eye then it will say Active and if it is a closed eye then it will check for a few seconds and then it will say the driver is drowsy and will beep an alarm.

We will use Python, OpenCV, TensorFlow, and Keras to build a system that can detect the closed eyes of drivers and alert them if ever they fall asleep while driving. If the driver's eyes are closed, this system will immediately inform the driver. OpenCV that we are going to use now will monitor and collect the driver's images via a webcam that was attached and feed them into the deep learning model and then the model will classify the driver's eyes as 'open' or 'closed.'[6]

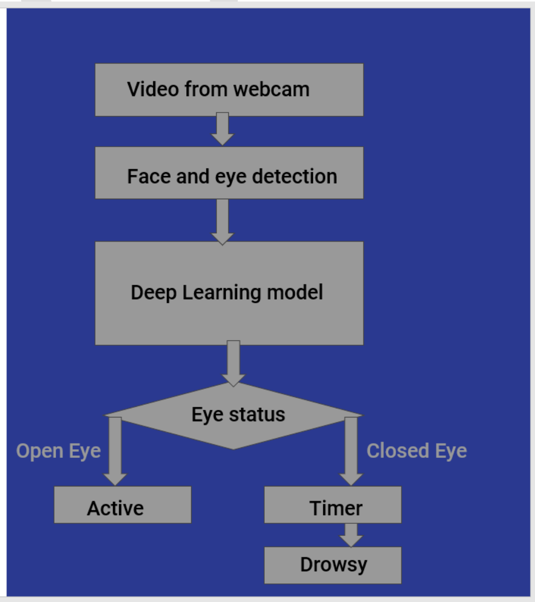


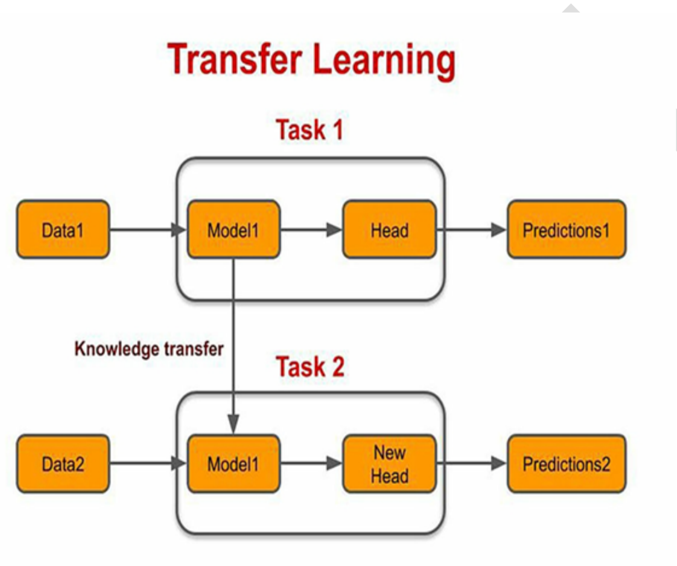
Fig. 8 (Workflow)

Transfer Learning

In this project, we will use transfer learning to build the model. Transfer Learning is a machine learning method where we use a pre-trained model for a new model with the related problem statement.

For example, a model that is used for the recognition of cars can be used for the recognition of trucks.

Here mainly it focuses on the knowledge that is gained while solving one problem and it applies to a different but related problem.[6]



<u>Fig. 9 (Transfer Learning)</u>

2.3.2 Support Vector Machine(SVM):

Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems.[7]

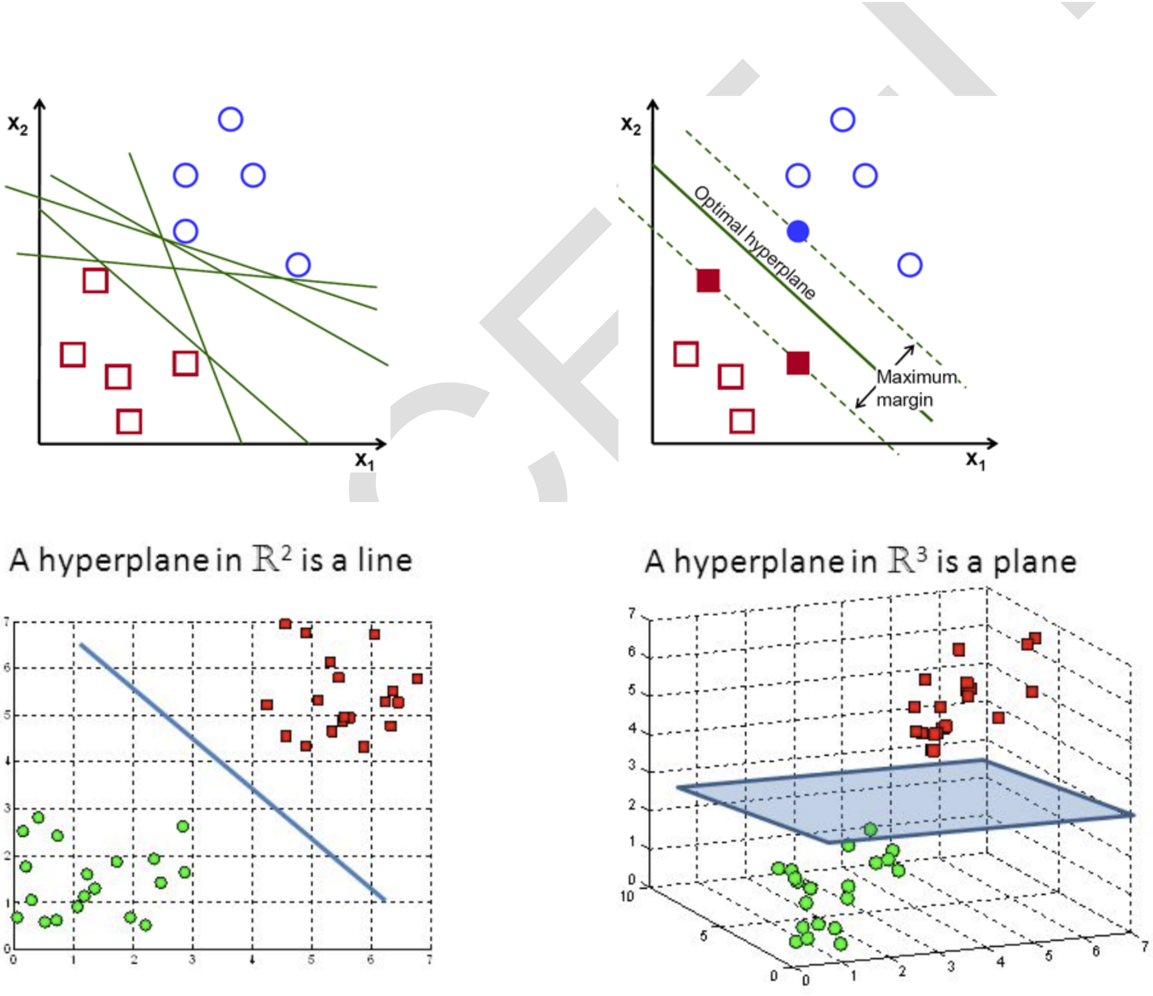
2.3.2.1 Brief Discussion

The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.[7]

Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.[7]

So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.[7]

The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.[7]



A hyperplane in $\mathbb{R}^2$ is a line

A hyperplane in $\mathbb{R}^3$ is a plane

Fig. 10 (Optimal Hyperplane)

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.[8]

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.[8]

3.Conclusion

Drowsy driving is one of the major causes of road accidents and death. Hence, detection of driver's fatigue and its indication is an active research area. Most of the conventional methods are either vehicle based, or behavioral based or physiological based. Few methods are intrusive and distract the driver, some require expensive sensors and data handling. Therefore, in this study, a low cost, real time driver's drowsiness detection system is developed with acceptable accuracy. In the developed system, a webcam records the video and the driver's face is detected in each frame employing image processing techniques. Facial landmarks on the detected face are pointed and subsequently the eye aspect ratio, mouth opening ratio and nose length ratio are computed and depending on their values, drowsiness is detected based on developed adaptive thresholding.

4.Future Planning:

The application is made using leading and the best classification methods (Artificial Neural Networks and Support Vector Machine). Performance upgrades, better optimization of the feature extraction algorithms used is a priority. The application can also be ported to work in the aviation industry where fatigue of pilots is a major concern.

Gnatt Chart for Project Planning:

REQUIREMENT ANALYSIS, FEASIBILITY STUDY							
		LITERATURE SURVEY					
			DATA ACQUISITION & FEATURE EXTRACTION				
					CODING		
						TESTING & ANALYSIS	
							DEPLOYMENT & MAINTENANCE
SEPTEMBER 2022	**OCTOBER 2022**	**NOVEMBER 2022**	**DECEMBER 2022**	**JANUARY 2023**	**FEBRUARY 2023**	**MARCH 2023**	**APRIL-MAY 2023**

References

[1] Driver drowsiness detection - Wikipedia en.wikipedia.org
[2] MRL Eye Dataset | MRL mrl.cs.vsb.cz
[3] Image processing in Python? www.tutorialspoint.com
[4] Binary Classification with Neural Networks - Atmosera www.atmosera.com
[5] Applied Deep Learning - Part 1: Artificial Neural Networks by Towards Data Science

[6] Drowsiness Detection System Amrutha K — Published On May 31, 2022- analytics vidyalaya

[7]Venu Madhav/Driver-Drowsiness-Detection-github.com

[8]Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science