International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Detection of Distracted Driver Using Machine Learning Techniques.

Pallavi Nimbargi[1], Sagar Hiremath[2], Shambhavi Belagali[3], Sharanya Sakri[4], Mrs. Arpita Patil[5]

[1]Dept. of Computer Science and Engineering S. G. Balekundri Institute of Technology Belagavi, Karnataka, India pallavinimbaragi1@gamil.com
[2]Dept. of Computer Science and Engineering S. G. Balekundri Institute of Technology Belagavi, Karnataka, India sagarhiremathssh@gamil.com
[3]Dept. of Computer Science and Engineering S. G. Balekundri Institute of Technology Belagavi, Karnataka, India belagalishambhavi6648@gamil.com
[4]Dept. of Computer Science and Engineering S. G. Balekundri Institute of Technology Belagavi, Karnataka, India sharanya7sakri@gamil.com
[5]Assistance Professor Dept. of Computer Science and Engineering S.G Balekundri Institute of Technology Belagavi, Karnataka, India arpita.i.patil@gmail.com

ABSTRACT —

The rise of advanced technology and its integration into modern society has brought numerous benefits and conveniences. However, it has also presented significant challenges, particularly concerning road safety. The phenomenon of distracted driving, characterized by drivers engaging in activities that divert their attention from the primary task of driving, has become a pressing issue worldwide. This abstract aims to explore the delectation or pleasure experienced by distracted drivers, shedding light on the underlying psychological and behavioral factors that contribute to this dangerous behavior.

Distracted driving is currently a major contributor to traffic accidents, highlighting the growing importance of intelligent vehicle driving systems. Recently, there has been a rising interest in developing driver- assistance systems that can detect driver actions and enhance overall safety. While these studies utilize various data types, including driver's physical conditions, audio and visual features, and vehicle information, the primary data source often consists of images captured within the car, encompassing the driver's face, arms, and hands. This research proposes an architecture based on convolutional neural networks (CNN) to effectively classify and detect driver distraction. An efficient CNN model with exceptional accuracy is implemented, emphasizing the utilization of intensive convolutional techniques.Keywords: Distracted driver analysis, predictive analytics, VGG-16, image analysis video analysis, Head position

I. Introduction

Distracted driving poses a significant problem causing many collisions and fatalities. New learning has become a viable answer to this issue. These methods can detect and identify distracted drivers by utilising cutting-edge algorithms and data analysis, enabling prompt intervention and potential accident prevention.

Machine learning models have the capability to be trained to spot a variety of distracted driving indicators. These models gather input from diverse sources such as visual cues from dashboard cameras, sensory data from accelerometers, and even aural impulses from microphones. These algorithms learn to recognize the patterns and anomalies connected to distracted behaviors through rigorous examination

Additionally, Machine learning algorithms can identify patterns or trends. and sensor data. The acceleration, deceleration, and general movement of the vehicle can be measured by gyroscopes and accelerometers. Machine learning models have the ability to recognize or detect. reckless driving, abrupt lane changes, or inconsistent driving habits that may indicate distraction by examining these patterns When combined with supplementary contextual data, such as location and time of day, these models can enhance their capabilities.

These models can be trained utilizing a variety of datasets gathered from various scenarios and environments in order to increase their accuracy. For training and verifying the models, real-world driving data collected by specialized data gathering tools or simulators is a crucial resource.

The primary goal of utilizing machine learning techniques to detect distracted drivers is to develop intelligent systems that can provide timely alerts or interventions. These systems have the ability to warn the driver via audible or visual cues, or even interact with other safety elements in the car to initiate corrective measures like turning off distracting devices for a while or turning on cutting-edge driver aid systems.

Machine learning can be divided into three categories: reinforcement learning, unsupervised learning, and supervised learning in supervised machine learning, a machine learning model is trained using a labeled dataset that contains known outputs for the purpose of guidance. After learning from input data, the model can then predict or decide. When a model is trained on an unlabelled dataset, it must find patterns and relationships in the data by using unsupervised learning. Training a model to make decisions based on a reward system is the process of reinforcement learning.

The machine learning process typically involves several stages. Initially, data collection and preprocessing take place, which may include data cleaning, outlier removal, and data normalization to make it suitable for machine learning algorithms. Subsequently, the next step involves selecting an appropriate algorithm or model that best fits the specific task at hand.

The type of data, the desired output, and the difficulty of the challenge will all play a role in this. After being chosen, the model is trained on the data using an optimization approach to minimize the difference between the predicted output and the actual outcome. Once trained, the model is employed to make predictions or assessments on new data.

There are numerous distinct machine learning models and algorithms, each having advantages and disadvantages. Support vector machines, decision trees, random forests, neural networks, and logistic regression are some of the commonly employed algorithms in machine learning. Every algorithm is built to handle a certain set of data and generate a particular set of results. For instance, logistic regression is used to forecast binary values, whereas linear regression forecasts continuous values.

II. Problem Statement

Given a dataset of 2D dashboard camera images, an algorithm requires developed to classify each driver's behaviors and determine driving attentively, wearing their seatbelt, or taking a selfie with their friends in the backseat etc. Subsequently, this information can be utilized. to automatically detect drivers engaging in distracted behaviors from dashboard cameras.

III. Methodology

In this case, we employ machine learning techniques to forecast distracted driver behaviors. These techniques generally follow a similar set of procedures illustrated below in a block diagram.

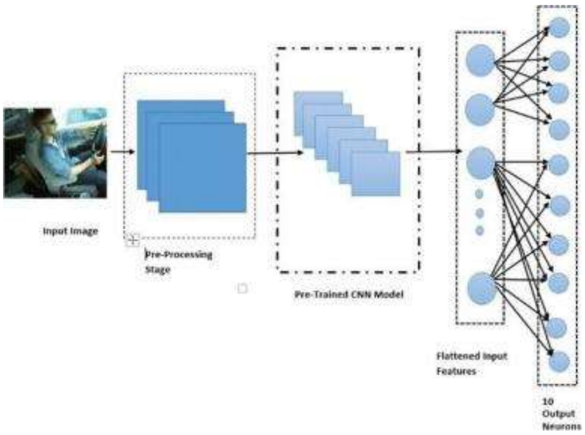

Fig III.1: Proposed Architectural Diagram

Data collection: Compile a the dataset consists of instances that are. labelled and include both distracted and un- distracted driving situations. Data might come from a various sources, such as dashcam film, car sensor data, or computer-generated driving scenarios.

2. Data preprocessing: By carrying out cleaning and preprocessing procedures, ensure the consistency and quality of the data. This stage involves correcting any class imbalances in the dataset as well as noise reduction and sensor data normalisation. relevant elements from the preprocessed data that allow for the distinction between distracted and non-distracted driving situations. Features could include visual signals from pictures or videos, sensor readings (such steering angle or acceleration), and contextual data (like the time of day or the weather).

4. Feature Selection/Dimensionality Reduction: To handle the size and complexity of the feature set, it is necessary to reduce dimensionality or choose the most informative features. This can be achieved through techniques such as feature significance analysis and principal component analysis (PCA).
5. Model selection: A suitable machine learning model needs to be chosen to effectively identify patterns and categorize instances of distracted driving. Popular models for this task include support vector machines (SVM), decision trees, random forests, as well as deep learning architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
6. Training and Evaluating: The dataset is divided into subsets, specifically for training and testing purposes. The training data is utilized to train the selected model on the extracted features, while the testing data is employed to assess its performance and determine its effectiveness. Accuracy, precision, recall, and F1-score are typical evaluation criteria.
7. Implementation: The model is integrated into a real-time system capable of processing streaming data from moving objects. This system is designed to efficiently detect instances of distracted driving and promptly issue appropriate warnings or notifications.

Continuous Improvement: Track the performance of the deployed system, get user input, and keep improving the model's precision and flexibility to changing driving conditions.

This might entail periodically retraining the model with fresh data.

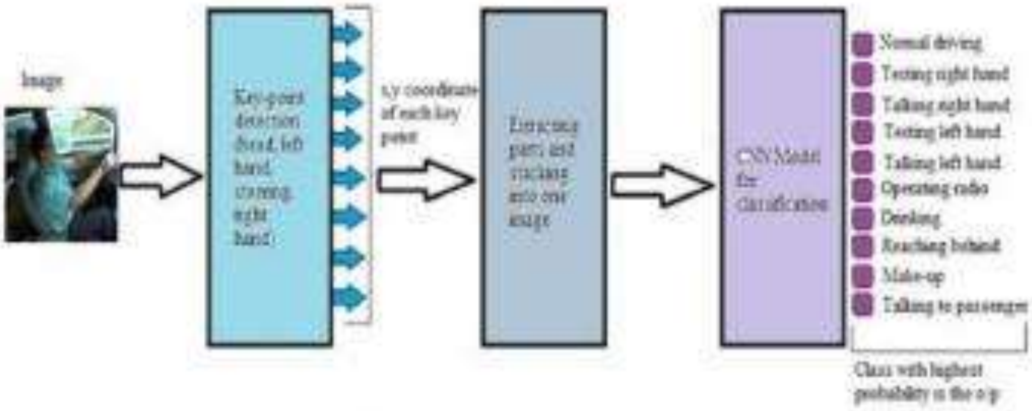

Fig III.2: Proposed Flow Diagram

The model receives input in the form of static images or video streams captured by cameras installed in the vehicle. These images undergo processing through a sequence of convolutional layers, enabling the extraction of relevant features from the input images. Subsequently, these extracted features are fed into a series of fully connected layers, ultimately generating a prediction regarding the driver's state of distraction.

The CNN model was trained on a dataset of labelled images or video frames, where each image is labelled as either a distracted driving behaviour or not. During training, the CNN model learns to recognise patterns and Features that are indicative of distracted driving behaviour such as talking on the phone, texting, or eating while driving.

After the model is trained, it becomes capable of real- time detection of distracted driving behavior. The model processes the input images, leveraging the learned patterns and features to make predictions. If the model predicts that the driver is distracted, an alert can be sent to the driver to help prevent potential accidents. But we are displaying the predicted result as of now.

TESTING

1. Load the saved model weights using the `load weights()` function. These weights represent the trained model, and they should be saved after the training is completed.
2. Compile the model using the same optimizer, loss function, and evaluation metric as during training. This ensures that the model will be evaluated using the same settings as during training.
3. 3To prepare the test data, the `ImageDataGenerator` and `flow_from_directory()` functions are employed. The `ImageDataGenerator` is utilized to preprocess the test images, including tasks like rescaling the pixel values. On the other hand, the `flow_from_directory()` function allows loading the test images from a specific directory and generating batches of images to be fed into the model during the testing phase.
4. To assess the model's performance on the test data, the `evaluate()` function is utilized. This function calculates both the loss and accuracy of the model by taking a data generator as input, which generates batches of test images. The function then returns the computed test loss and accuracy, providing valuable insights into the model's effectiveness on the test dataset.

Upon executing the `evaluate()` function, an estimation of the model's performance on unseen data is obtained. The test accuracy serves as an indicator of the model's ability to generalize effectively to new data. A high test accuracy implies that the model can accurately classify distracted and non-distracted drivers in real-world scenarios, demonstrating its capability to handle real-world situations with proficiency.

Layer Explanations:

1. A Conv2D layer with 32 filters and a kernel size of (3,3) that accepts input pictures with a dimension of (DIM,DIM,NB_CHANNELS) and generates feature maps with a dimension of (DIM,DIM,32). The padding is set to "same", which indicates that zeros are added to the input image to guarantee that output feature maps have the same spatial dimensions as the input.

2. An activation layer with a ReLU activation function that transforms the Conv2D layer's output nonlinearly.
3. The batch-normalization layer, which uniformly normalises activations across batch sizes and spatial dimensions.
4. A layer called MaxPooling2D that down samples feature maps in both dimensions by a factor of 3 by performing a max pooling operation over a spatial neighborhood of size
5. A Conv2D layer with 64 filters and a kernel size of (3,3) results in feature maps that are (DIM/3,DIM/3,64) pixels in size.
6. An activation layer that uses ReLU activation. Layer of batch normalisation 7.
7. A Conv2D layer with 64 filters and a kernel size of (3,3) results in feature maps that are (DIM/3,DIM/3,64) in size.
8. An activation layer that uses ReLU activation.
9. Layer for batch normalisation.
10. A layer called MaxPooling2D with a pool size of (2,2) down samples the feature maps in both dimensions.
11. A Conv2D layer with 128 filters and a kernel size of (3,3) results in feature maps that are (DIM/6,DIM/6,128) in size.
12. An activation layer that uses ReLU activation.
13. Layer for batch normalisation.
14. A Conv2D layer with 128 filters and a kernel size of (3,3) results in feature maps that are (DIM/6,DIM/6,128) in size.
15. The activation layer employs the ReLU activation function, while the 17th layer incorporates batch normalization.
16. To prevent overfitting, a dropout layer is included, randomly deactivating 50% of the activations.
17. A layer called MaxPooling2D with a pool size of (2,2) down samples the feature maps in both dimensions by a factor of 2.
18. In the network architecture, the 20th layer serves as a flattening layer, transforming the output of the preceding layer into a one-dimensional (1D) vector.
19. A dense layer with 1024 units is employed to perform a linear transformation on the input vector. ReLU activation function in the activation layer, 22. Batch Normalization layer, number 23.
20. The model includes a dense layer with 10 units, which produces the final output and aligns with the 10 classes present in the image dataset.
21. The SoftMax activation layer is utilized to generate class probabilities for every input image, enabling the model to assign probabilities to each class.

IV. Conclusion:

The application of machine learning techniques has demonstrated their effectiveness in detecting distracted drivers. Through the utilization of diverse data sources including video feeds, sensor inputs, and other vehicle- related data, machine learning models can be trained to accurately recognize behaviors linked to driver distraction. These techniques hold the promise of improving road safety by delivering real-time alerts and interventions to mitigate accidents caused by distracted driving.

Machine learning algorithms have the capability to analyze and examine a wide range of features, including driver behavior patterns, eye movements, head pose, facial expressions, and vehicle dynamics, to identify signs of distraction. These algorithms can be trained using large labeled datasets,

where human experts annotate instances of distraction or non-distraction. Through the learning process, the models can recognize complex patterns and make accurate predictions based on new data.

Using machine learning for detecting distracted drivers offers several advantages. It enables fast processing of large volumes of data, adjusts effectively to evolving environments and contexts, and enhances performance through continuous learning. Additionally, machine learning can accommodate individual differences and identify distinct patterns of distraction. Nonetheless, there are certain challenges associated with the implementation of machine learning techniques for detecting distracted drivers. One challenge is acquiring diverse and representative datasets that capture various types of distractions and driving conditions. Another challenge is ensuring the privacy and ethical use of the collected data, as it often involves capturing sensitive information about individuals. Furthermore, the deployment The successful deployment of machine learning systems for detecting distracted drivers necessitates meticulous validation, rigorous testing, and seamless integration into established vehicle systems or infrastructure. The models need to be robust, reliable, and capable of handling real-time processing to provide timely warnings and interventions.

VGG architectures include a vast number of parameters. In this study, this parameter.

References:

[1] Nemcova et al., "Multimodal features for detection of driver stress and fatigue," IEEE Trans. Intell. Transp. *Syst.*, vol. 22, no. 6, pp. 3214–3233, Jun. 2021.

[2] Kashevnik, R. Shchedrin, C. Kaiser, and A. Stocker, "Driver distraction detection methods: A literature review and framework," IEEE Access, vol. 9, pp. 60063– 60076, 2021.

[3] Talbar, and S. Gajre, "Towards computationally efficient and realtime distracted driver detection with MobileVGG network," Vehicles, vol. 5, no. 4, pp. 565– 574, Dec. 2020.

[4] R. Liu, and M. Gong, "CA-PMG: Channel attention and progressive multi-granularity training network for fine-grained visual classification," *IET Image Process.*, vol. 15, no. 14, pp. 3718–3727, Dec. 2021.

[5] J. Chen et al., "Fine-grained detection of driver distraction based on neuralarchitecture search," IEEE *Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5783– 5801, Sep. 2021.

[6] R. H. Faisal, and M. M. Rahman, To effectively implement machine learning systems for detecting distracted drivers, it is crucial to conduct thorough validation, extensive testing, and smooth integration into pre-existing vehicle systems or infrastructure.Feb. 2019

[7] Kabbai, M. Abdellaoui, and A. Douik, "Image classification by combining local and global features," *Vis. Comput.* 679–693, May 2019.

[8] kabbai "Detection of distracted driver using convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018

[9] Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from asingle natural image," in *Proc.* IEEE/CVF Int. Conf. Comput. Vis., Oct.2019