Driver Distraction Detection System with Categorized Alerting based on SVM and PCA Optimization

Ruudra Amola   
Department of Computer Science and Engineering  
Graphic Era Deemed to be University Dehradun, India  
ruudraamola@gmail.com

Guru Prasad M S  
Department of Computer Science and Engineering  
Graphic Era Deemed to be University Dehradun, India  
guruprasad.cse@geu.ac.inShailja Bahuguna  
Department of Computer Science and Engineering  
Graphic Era Deemed to be University Dehradun, India  
shailjabahuguna2003@gmail.com

Harsh Pravin Warkari  
Department of Information Technology  
PCCOE PUNEPune, India  
harsh.warkari21@pccoepune.org Rishi Kumar  
Department of Computer Science and Engineering  
Graphic Era Deemed to be University Dehradun, India  
rishikumar.cse@geu.ac.in

*Abstract*— The road accident scenario in India can be understood by simple statistics: every year, more than 150 hundred people lose their lives on the road. It is like crashing a jumbo jet carrying 340 people every day without any survivors. It is the enormity of the situation. Road accidents impact India's GDP by 3.14% (IIT Delhi- TRIPP Study). This paper mainly focuses on the solution to the issue of distracted driving. Distracted driving is indulging in attention-breaking activities that can disrupt the flow of the driver's coordination on the road. These distractions are broadly classified as Manual, Visual, and Cognitive. The following research shows how driver distractions can be detected using computer vision-equipped cameras in the vehicle to detect an act of distraction. We have used the State Farm Distracted Driver Detection Dataset (SFDDD), consisting of images classified into 9 categories of driver distraction activities. We have adopted a machine learning approach and trained a classification model using a linear support vector machine (SVM), achieving an accuracy of 98.95%. The result has a trade-off with the time and space and was further improved by merging SVM with Principal Component Analysis (PCA), significantly reducing the model training time and even enhancing the accuracy to 99%. Using the model, we have also implemented a prototype for a prioritized alerting system that warns the driver when it detects a distraction and sets off an alarm.

Keywords— fatality, GDP, distracted driving, coordination, manual, visual, cognitive, machine learning, SFDDD, SVM, time, space, PCA

# Introduction

This Driving is an act of controlling or operating a vehicle on land. It involves controlling the speed, movement, and direction of a vehicle. Driving is done by a person, known as a driver, manually using equipment attached to the vehicle. Driving is a compassionate activity involving a solid focus and coordination of motor and sensory organs. The driver needs to maintain undivided attention and be calm and tranquil while driving on the road. Any distraction, for instance, if the driver takes his eyes off the road while driving, even for a fraction of a second, can have serious ramifications [1][2].

Distracted driving is indulging in attention-breaking activities while driving that can disrupt the flow of the driver's mental and physical coordination and lead to difficulty in driving and maintaining focus on the road. It creates a perilous situation for the driver and fellow passengers, pedestrians, and people driving other vehicles on the road [3][4].

In recent years, there has been a significant rise in the number of vehicles running on the roads, and the growing use of technology and devices like mobile phones has contributed significantly to the issue of distracted driving. Researchers and experts worldwide have expressed keen interest in solving this problem and have proposed solutions and ideas to tackle the issue[5].

Studies have concluded that the sources of distractions may lie inside the vehicle or outside of it. It can be equipment-related or otherwise and maybe self-encouraged or imposed on the driver under certain circumstances. Sources of distractions are widely classified under three categories: Manual, Visual, and Cognitive [22].

Manual distractions are the ones where the driver may have eyes on the road but is doing some manual activity while driving, like taking his hands off the steering wheels, searching for something in the dashboard, attempting to reach the backseat of the car, and more [1][6].

Visual distractions, as the name suggests, relate to visual distractions in and around the vehicle, like the driver trying to repeatedly look at the GPS navigator, looking at roadside billboards and other surroundings, and looking at the rearview mirror for too long.

Cognitive distractions are fatal, as they cannot be seen or detected. They occur in a person's mind. It can be called mental absence while driving, thinking of something that disturbs the driver's focus[7].

According to the records laid by the World Health Organization, road accidents are the 9th leading cause of death. India is a country where the frequency of accidents is unmatchable to any other nation, an accident happens every minute, and a life is lost every four minutes. One of the most prominent causes of driver distraction is using mobile phones for calling or GPS navigation. A previous study revealed that 31% of the drivers who used a mobile phone while driving met with an accident. The current study analysed and modelled the effects of mobile phone distraction on drivers' reaction time in India's different age groups [8][9].

A research study by the Indian Institute of Technology Bombay (IITB) has shown how using technical equipment like mobile phones while driving can distract drivers and affect their ability to handle fatal situations by reducing their reaction times. The study results have shown that calling and texting while driving degrades the ability to react to a driver [6].

In a significant advancement to embedded distraction detection systems, German technology firm Bosch developed what it describes as an "interior monitoring system" for cars that utilize the extensive computer vision approach that is based on the idea of merging artificial intelligence (AI) and cameras. One of its features included a camera integrated into a steering wheel. It can detect when a driver is distracted, when their eyes are becoming heavy, and when they have drowsiness and are trying to reach out to the rear seats or the person sitting next to them [7][10].

But the solution has yet to reach a broader mass. Firstly, the problem of distraction is still unclear to drivers on the road. It is still casual for people to have snacks and drinks while driving, listen to loud music, and text without realizing the seriousness of the events it can lead up to. Secondly, all these brands have implemented the systems with mostly one feature: fatigue, driver drowsiness, and eye-head movements. However, the reasons for distraction are not limited to drowsiness; there is much more to counter.

Distracted driver classification primarily involves two main approaches. The first approach involves wearable setups with sensors measuring brain signals, heartbeat, and voluntary actions. However, this approach proves to be cost-effective and uses complex hardware. The other idea involves using camera vision systems to detect and predict the type of distraction. It commonly uses deep learning algorithms to perform feature extraction and classification tasks. While the first approach can detect cognitive distractions, the second can detect manual and visual distractions. In this research work, we explore the second approach in a much more open fashion.

State Farm is an insurance firm in the United States. They surveyed with the help of some subjects, both male and female and created a dataset of images depicting different types of distracting activities that drivers tend to indulge in. State Farm aimed to improve these statistics and ensure their customers by testing how dashboard-fixed cameras automatically detect drivers' distracted behaviours.

In April 2016, State Farm initiated a competition on the Kaggle website to collect ideas and solutions to the driver's behaviour problems using their dataset. We have used the same State Farm driver dataset to train our proposed machine learning model.

Our problem focuses on detecting distracted acts of car drivers based on the State Farm driver distraction image dataset from Kaggle, using Machine Learning techniques to achieve maximum accuracy. The dataset is widely classified into 10 classes representing 9 different causes of hindrances in driver attention. The 10th class represents "Safe driving”.

# LITERATURE REVIEW

This section outlines the research on distracted driver detection in the last few years. Authors in [10] presented a review paper on Driver Drowsiness Detection Systems and identified some measures for detecting distracted drivers. "Biological-based measures" involve wearable sensors to detect the driver's drowsiness using brain activity, heart rate, and muscular activities.

While wearable technology is expensive and requires user involvement, "Image-Based Measures" utilizes a camera for vision-based classification. Authors in [11] employed SVM and logistic regression models to develop a real-time approach for detecting driver distraction. The SVM model outperformed traditional logistic regression models, achieving an average accuracy of 81.1%. In the research work proposed by authors in [27], they performed feature extraction through CNN and used PCA before the training of the SVM model, achieving an accuracy of 96.28%. Authors in [12] introduced a deep learning-based CNN model that collected the images for the dataset using a Kinect camera. The study utilized self-made feature extractors integrated with classifiers like SVM and CNN.

However, CNN-based continuously outperformed SVM. The superiority of CNN was attributed to its enhanced efficiency and faster computational speed.

The problem has also been studied using various Deep-learning approaches to achieve better results. Deep learning approaches involved models such as AlexNet and convolutional neural network (CNN) architectures like ResNet50, VGG-16, VGG-19, and GA-weighted ensemble. Individual CNN models tend to improve accuracy over a better scale on raw image datasets. Authors in [13] proposed using the Res-Net 50 network to classify the images of distracted drivers and a tag that represents the type of distraction in a video. Their approach achieved an accuracy of 94%.

In 2023, research [14], which focused on detecting driver behaviour with a combination of artificial deep learning and machine learning models with GA (genetic algorithm), achieved an accuracy of 99.80%.

Overall, when compared to conventional machine learning approaches, deep learning models outperform them. Deep learning can help identify a driver's stress levels and behavioral patterns by tracing eye and physical movements.

Some of the significant research works that have led to a paradigm shift in the study of driver distraction detection frameworks have been summarized in Table I.

# Proposed Model and Framework

## Dataset

Getting the system to work begins with gathering data from the dataset. The images are coloured and are of size 640×480 pixels. Each class encompasses nearly 2300 images, and the distribution per class and label are detailed in Table II below.

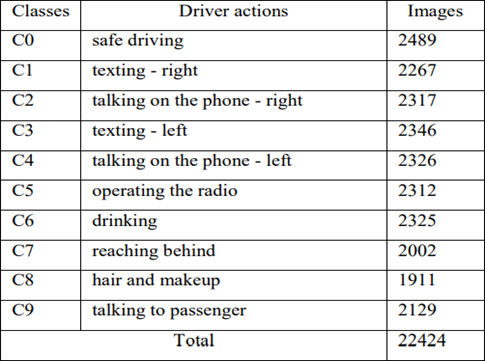
The dataset has been split into train and test subsets. The training subset consists of 10 driver distraction classes (c0- c9) as identified.

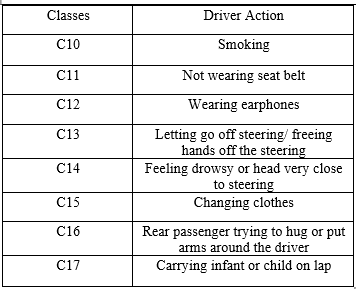
Apart from the classes already available in the State Farm dataset, we have also proposed an extended list of driver distractions that can be incorporated into the database. After thoroughly studying traffic-related statistics, we concluded that many other distraction classes exist, as listed in Table III.

Table I Analysis of related work

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Year** | **Paper Reference No.** | **Contribution** | **Algorithm** | **Advantages** | **Weakness** |
| 1 | 2018 | [21] | Used more than one model to decrease the log loss value. | Used an ensemble of various CNN models, VGG-16, VGG-19 and  InceptionV3. | Log loss score of 0.795. | The accuracy of the model needs improvement. |
| 2. | 2021 | [22] | Reviewed various driver distraction approaches. | Provided an overview of research in driver distraction and distraction detection systems. | Identified driver distraction classes, including manual, visual & and cognitive. | More studies can be done on this. |
| 3. | 2021 | [23] | Identified EfficientDet-D3 as the most optimal model for detecting driver's inattention. | The Efficient-Det model detects objects and the ROI of the body through pictures. | MAP score of 99.16%. | The model can be improved by using dynamic data. |
| 4. | 2022 | [13] | Images of distracted drivers identified along with a tag to represent a distracted driver in a video. | Preprocessed images by resizing to make them suitable for use with the ResNet-50 model. | Accuracy of 94% | Need of better image preprocessing technique. |
| 5. | 2019 | [24] | This work focuses on distracted driver posture recognition as a part of the human action recognition model. | A combination of Residual Neural Network, Inception and the HNN to improve the performance. | Accuracy of 96.23%  achieved on SFDDD. | Dataset can be extended to include more classes of distraction. |
| 6. | 2018 | [25] | Presented a new dataset for "distracted driver" posture estimation. | Proposed a method that used genetic algorithm weighted ensemble hand and face detections of CNN. | Accuracy of 94.29% | Need for performance improvement due to temporal features. |
| 7. | 2020 | [26] | Built a powerful multiple class classification model that detected categories of distraction and trained on SFDDD. | Combination of several augmentation techniques, like skin segmentation, facial blurring, and classical augmentation, to detect distraction. | F1 score of 0.662. | ResNet-50 and ensemble methods performed better than this model. |
| 8. | 2023 | [14] | A real-time driver distraction detection model built to avoid car accidents. | Combination model of [CNN](https://www.sciencedirect.com/topics/engineering/convolutional-neural-network) and [GA.](https://www.sciencedirect.com/topics/engineering/genetic-algorithm) | Accuracy of 94.80% | Need of improvement with more data. |

Table II Driver Distraction Categories from Dataset





In the data visualization and examining phase of our research we first explored the distribution of images in the different classes. There is a total of 10 different driver actions labeled ('c0' through 'c9'). Furthermore, we examined the composition of the dataset, including the division of training images and testing images.

To visually represent the distribution of classes we utilized data visualization tools. The plot is generated using 'Seaborn' and 'Matplotlib' libraries. The Matplotlib library is used for 2D plotting of the Class Distribution which includes labeled axes for class names and image counts. Seaborn's 'countplot' function is used to create a bar plot of the class distribution as shown in Figure 1.

## Dataset Preprocessing and Sampling

Image data is loaded using the 'skimage' library, and each image in the dataset is resized to the exact dimensions of (150,150,3) using the 'resize' function. The resizing of images ensures the uniformity of the input dataset. The pixel values of the resized images were then flattened and stored in an array while corresponding target labels (classes through 'c0' to 'c9') is stored in another array. 67,501 features represent each image after flattening the pixel values.

A random dataset sample is created for enhanced training, which is altered for subsequent model training and evaluation. A random subset of the dataset is generated to improve training efficiency and evaluation accuracy. This subset comprised 10,000 images and is subsequently divided into two sets: a training subset containing 8,000 images and a testing subset containing 2,000 images.

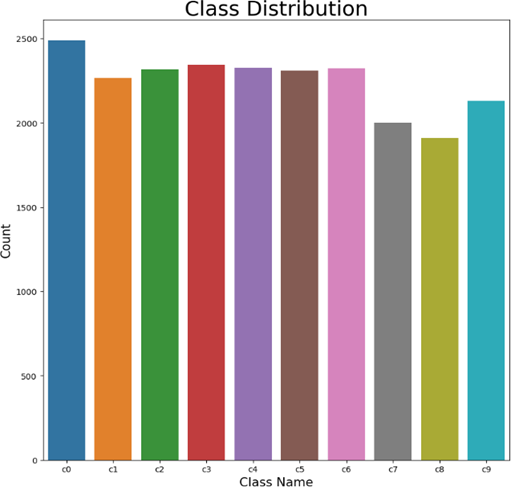


Fig. 1 Bar Plot Sampling for Class Distribution

## Dimesnionality Reduction Using Principal Component Analysis Technique

Datasets comprising images are typically high-dimensional due to the large number of pixels in each image. This high dimensionality requires significant storage space and increases the computational time required for training machine learning models.

PCA is used to mitigate this issue and optimize memory usage. Principal Component Analysis is a dimensionality reduction algorithm that identifies principal components, reducing the complexity of the dataset while retaining the variance.

In the preprocessing step, it is identified that each image represents 67,501 features, which are further processed based on variance. We used the 'sklearn' library to import PCA from the 'sklearn.decomposition' module. To perform PCA the standardization of the data is performed using "Standard Scaler", which involves removing the mean and scaling to unit variance.

The number of principal components selected was 800; we used a plot in Fig. 2, which shows how much of the variation in the data is explained to add more components. Looking at where the curve levels off, we could see that 800 components capture approximately 90% of the variance, encompassing most of the information in the image data. This step causes a dimensionality reduction from 67,501 to 800, resulting in more efficient storage and computation. PCA effectively streamlines the dataset and is faster to process during machine learning model training.

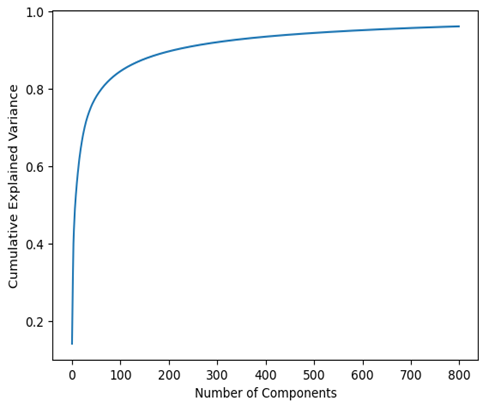


Fig. 2 Number of Principal Components and their cumulative variance

## The Support Vector Machine Algoirthm

SVM is a supervised ML algorithm that is used in classification and regression problems. The algorithm uses an input set of data that includes individual weights associated with each element in the dataset. These weights are labels.

The algorithm's output is an optimal hyperplane that helps separate and classify the dataset points or elements. In a one-dimensional space, the hyperplane is a point, in a two-dimensional space, the hyperplane is a line; similarly, in a 3-D space, the hyperplane is a surface that separates the space into 2 parts where each class of data points lie on either side of the plane.

In the case of 2-D space, we have 2 kinds of hyperplane situations: one is the linearly separable hyperplane, and the other is the non-linearly separable hyperplane, as shown in Fig. 3. Space is said to be linearly separated when we can generate a line as a hyperplane. However, when the data points are inseparable using a line, they are said to be a non-linearly separated plane.

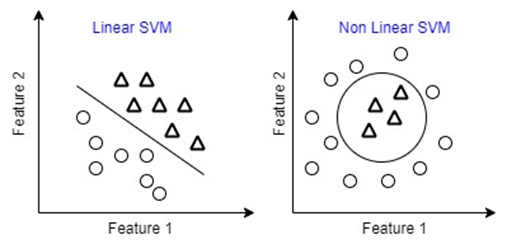


Fig. 3 Linear and Non-Linear Separated Spaces

There are a lot of possible hyperplanes that can exist in space, but what SVM does is help find the best or the optimal one. The best hyperplane maximizes the margin between the positive and the negative samples. This margin is also called the street around the hyperplane that separates the positive and negative data samples.

The support vectors are the data points that lie closest to the hyperplane and lie on the decision boundaries on either side of the hyperplane. These are the most difficult to find as the position of the optimal hyperplane only depends on the support vectors. Moving the support vectors affects the decision boundaries, but moving other points or vectors does not affect the decision boundaries.

A linear SVM model is selected for its effectiveness in multi-class classification tasks and is trained using the training subset. The instance of the SVM classifier is created with a linear kernel and regularization parameter ('c') set to 1.

## Model Training and Approach

In terms of the State Farm dataset, it is evident that the data is divided into multiple classes, indicating an image classification task. SVM can handle the multiclass classification tasks through various strategies such as 'one vs one' and 'one vs rest'. In our proposed framework, we have imported the 'Support Vector Classifier: SVC' from 'sklearn' library, built upon 'libsvm'. This classifier efficiently addresses multi-class scenarios using the 'one vs one' approach, employing pairwise binary classifiers.

In the 'one vs one' strategy, a binary classifier is trained on every possible pair of classes. With 'N' classes, the number of binary classifiers trained is given by the formula N(N-1)/2. In our dataset, comprising 10 classes, a total of 45 binary classifiers were employed. We implemented the 'one vs one' strategy in our SVC using the 'decision\_function\_shape' parameter.

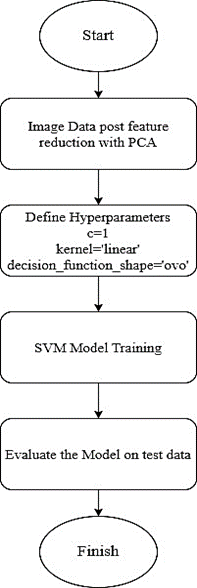


Fig. 4 Flowchart of proposed SVM Framework

The model was trained on a subset of 8,000 using stratified sampling, ensuring that each subgroup's proportion was maintained in the training and testing sets.

The process diagram of our suggested framework using the SVM classifier is depicted in Fig. 4. The SVM is employed to distinguish between various categories of distracted driver actions, specifically identifying whether a driver is distracted. Finally, the trained Linear SVM model was evaluated on the test set. Model predictions are compared against the true labels to evaluate the accuracy of the validation.

# IMPLEMENTATION OF AN ALERTING SYSTEM

We have implemented a prioritized alerting system based on the PCA-supported SVM algorithm. As the PCA-supported model generated a better prediction accuracy rate, we loaded the model onto a pickle file and fed it into our prototype alerting system's web application

## Uploading and predicting the class

To create the UI of the alerting system, we have created a basic web application using the flask framework, as shown in Fig. 5. The application takes any random image as input, which belongs to one of the distraction classes from the state farm dataset. It then predicts the class the image belongs to using the SVM model. There are 2 options in the application; one is to upload the image of the driver that is to be predicted and the other one is to 'predict' the output and then start the warning with the alarm. The system first calls out the warning, then the alarm, and then displays the uploaded image on the screen.

A person driving a car

Description automatically generated

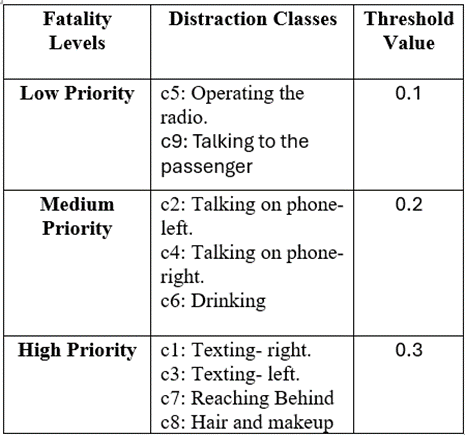
Fig. 5 UI of the system

## Identifying risk levels

We have also introduced a concept of prioritized alerting into our model. Prioritized alerting is related to setting the types of distractions onto a hierarchy scale where the auditory alert and warning based on the level of risk it poses while driving. The scale is defined using a three-level measure of low, medium, and high risk that a particular distraction poses.

In this system, we set different distraction severity thresholds for drivers. As shown in Table IV, we assign numbers (0.1, 0.2, and 0.3) to represent these levels corresponding to the distraction classes. A driver's distraction level below 0.2 is considered low risk and triggers a less urgent alert. If the distraction level is between 0.1 and 0.2, it's more serious, triggering a medium-level alert. These alerts help warn the driver about potential dangers on the road. Using these thresholds, the system can better judge the urgency of the situation and give appropriate warnings to keep the driver safe.

Table IV Hierarchy scale of fatality levels



## Auditory warnings and alerts

To trigger the auditory warning speeches, we have used the 'gTTS' or Google Text-to-Speech API to trigger a speech warning and specify the exact risk or type of distraction activity to the driver. The API utilizes multilingual support in the TTS system to provide warnings in different languages if required. It is followed by an alarm like the one that is triggered when the driver is not wearing the seatbelt.

The difference between a normal alarming system and ours is that our alarming system takes priority or measure of the risk level in consideration, and based on that, it triggers a mild, a bit strong, and a heavy alarm tone. One issue with using the 'gTTS' API is that it requires a stable internet connection. This is causing a delay from the time of detecting the distraction to the point of calling out a warning. But, the delay is minimal, which may not affect the system's aim to alert the driver. The delay ranges from 0.4-0.9 seconds. Also, a conditional check is done to ensure that the prediction is not equal to "c0" since c0 class corresponds to "Safe Driving," for which an alert is not required.

# Results and discussion

## Confusion Matrix

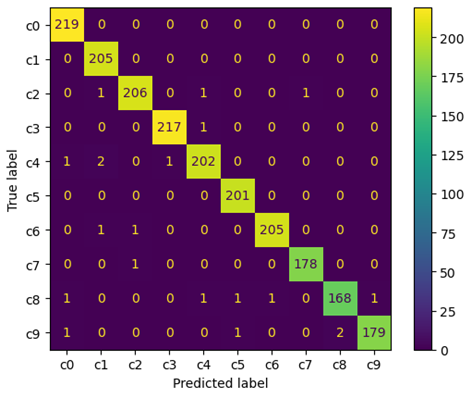


Fig. 6 Confusion Matrix for SVM with PCA

The confusion matrix provides insight into the performance of the multiclass classification model on the testing subset of 2000 images as shown in Fig. 6. Each row represents the actual class, while each column represents the predicted class. The diagonal elements indicate the number of correctly classified instances for each class, while off-diagonal elements represent misclassifications.

However, our model exhibited confusion primarily between the instances from the Hair and Makeup class (c8), and instances from 5 other classes: Talking to Passenger (c9), Talking on the Phone – Left (c4), drinking (c6), operating the radio (c5), and Safe driving (c0). This confusion may stem from similar visual features shared among these classes, such as hand gestures or head movements, which could be present during safe driving and distracted behaviours like interacting with passengers or using a phone.

## Comparative Analysis of Performance: PCA vs. Non-PCA-based Approaches

We conducted our model training and testing with different sets of features. In the initial phase, we implemented an SVM-based framework with 67,501 features, which involved resizing the images to impose uniformity and flattening the pixels into a 1-dimensional array with corresponding labels. Finally, following image preprocessing, we executed a feature reduction step using Principal Component Analysis (PCA), effectively reducing the dimensionality of the image features from 67,501 to 800. Remarkably, the SVM model, operating without dimensionality reduction, achieved a commendable accuracy of 98.95%. However, this came at the expense of a considerable training time of approximately 30 minutes and 27 seconds. Upon integrating PCA into our framework, we observed an enhancement in model accuracy and a substantial reduction in training time, now merely 11.2685 seconds. This optimization transformed our framework to be more efficient, exhibiting improved memory utilization and processing speed. A comprehensive comparison of results is shown in Table V.

Table V Comparative Analysis of SVM performance

|  |  |  |
| --- | --- | --- |
| **Metrics** | **With PCA** | **Without PCA** |
| **Model Accuracy** | 99% | 98.95% |
| **Training Time** | 11. 268573 sec | 30 min and 27.585141 sec |
| **No. of Features** | 800 | 67501 |

In the initial phase of our model implementation, we used 67,501 features to train the SVM classifier, achieving an accuracy of 98.95%. Training the classifier took 30 minutes and approximately 27.5 seconds. Reducing the features to 800 components using Principal Component Analysis increased the accuracy to 99% and reduced the training time to 11.2685 seconds.

The PCA-optimized SVM classifier was serialized into a pickle file for deployment. Leveraging Flask, we locally hosted the algorithm, utilizing the trained model pickle file for real-time prediction. Our system dynamically issues warning messages and alert tones based on predicted risk levels or potential fatalities. Despite a latency range of 0.3 to 0.9 seconds in our alerting system, attributed to the use of gTTS API, which requires a stable internet connection, this latency range doesn't lead to safety risks to the drivers

# CONCLUSION AND FUTURE WORK

State Farm is an extensive dataset, and processing the images requires much time and space. Therefore, there's a need for potential optimizations and changes. The categories identified by the State Farm Dataset are limited and would fail to identify potential categories that can cause a driver to be distracted from the road. Therefore, our study proposes 8 new categories that can enhance the classification.

In the future, we wish to replace the gTTS module that we used to trigger auditory alerts based on warning messages with a module that can operate offline. This would reduce the delay between detection and alerting as it would not depend on internet speed.

Also, we aim to improve the computational performance of the model by integrating our existing PCA-based SVM model with deep learning or CNN approach.

##### References

1. Furkan Omerustaoglu, C. Okan Sakar, Gorkem Kar, "Distracted driver detection by combining in-vehicle and image data using deep learning," Sensors, vol. 96, p. 106657, November 2020.
2. Monagi H. Alkinani, Wazir Zada Khan, Quratulain Arshad, Mudassar Raza, "HSDDD: A Hybrid Scheme for the Detection of Distracted Driving through Fusion of Deep Learning and Handcrafted Features," Sensors, vol. 22, Issue:5, p. 1864, February 2022.
3. Deepthi M. Pisharody, Binu P. Chacko, K.P. Mohamed Basheer, "Driver distraction detection using machine learning techniques," Materials today, Proceedings, vol. 58, part 1, pp. 251-255, February 2022.
4. Y. Yao, X. Zhao, H. Du, Y. Zhang and J. Rong, "Classification of distracted driving based on visual features and behavior data using a random forest method" in Transportation Research Record: J. Trans. Res. Board, Sep. 2018.
5. K. Sengupta, A. Srivastava, S. Shreyansh, S. Aggarwal, and V. N. Kumar, 'Driver sleep detection: A new and accurate approach,' in Proc. Innov. Power Adv. Comput. Technol. (i- PACT), pp. 1–7, 2021.
6. K Sudheevar Reddy, Kiran N Deshmukh, Lokesh M, Prathik Raj RC, Dr. Kavitha KS, "Deep Learning Approaches to Detect Distracted Driver Activities: A Survey", International Journal for Research in Applied Science & Engineering Technology, vol. 11, Issue:2, pp. 1074-1082, February 2023.
7. T. Liu, Y. Yang, G. -B. Huang, Y. K. Yeo and Z. Lin, "Driver Distraction Detection Using Semi-Supervised Machine Learning," in IEEE Transactions on Intelligent Transportation Systems, April 2016 .
8. K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp. 35–43, Jun. 2015.
9. K. Roy, "Unsupervised sparse, nonnegative, low rank dictionary learning for detection of driver cell phone usage," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 10, pp. 18200– 18209, Oct. 2022.
10. Yaman Albadwai, Maen Takruri and Mohammed Awad, "A Review of Recent Developments in Driver Drowsiness Detection Systems", 2022.
11. Yulan Liang, Michelle L. Reyes and John D Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines," IEEE Transactions on Intelligent Transportation Systems, vol. 8, Issue:2, pp. 340-350, July 2007.
12. Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Y. Wang, "Driver activity recognition for intelligent vehicles: A deep learning approach," IEEE Trans. Veh. Technol., vol. 68, no. 6, pp. 5379–5390, June 2019.
13. Muhammad Saiful Haqem Saiful Bahari, Lucyantie Mazalan, "Distracted Driver Detection Using Deep Learning", Proc. IEEE 18th International Colloquium on Signal Processing & Applications (CSPA), May 12-12, 2022.
14. Abeer. A. Aljohani, "Real-time driver distraction recognition: A hybrid genetic deep network-based approach," Alexandria Engineering Journal, vol. 66, pp. 377-389, March 2023.
15. J. Hu, "Automated detection of driver fatigue based on AdaBoost classifier with EEG signals,' Frontiers Comput. Neurosci., vol. 11, p. 72, Aug. 2017.
16. F. Tango and M. Botta, "Real-time detection system of driver distraction using machine learning," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 894–905, June 2013.
17. H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks,' J. Adv. Transp., vol. 2019, pp. 1–12, Feb. 2019.
18. M. D. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis, 'Distracted driver detection: Deep learning vs handcrafted 2017.
19. D. Tran, H. Manh Do, W. Sheng, H. Bai, and G. Chowdhary, "Real-time detection of distracted driving based on deep learning," IET Intell. Transp. Syst., vol. 12, no. 10, pp. 1210–1219, Dec. 2018.
20. H. V. Koay, J. H. Chuah, and C.-O. Chow, 'Convolutional neural network or vision transformer? Benchmarking various machine learning models for distracted driver detection,' in Proc. IEEE Region 10 Conf. (TENCON), pp. 417–422, Dec. 2021.
21. Prof. Pramila M. Chawan, Shreyas Satardekar, Dharmin Shah, Rohit Badugu, Abhishek Pawar, "Distracted Driver Detection and Classification," Int. Journal of Engineering Research and Application, vol. 8, Issue:4 (Part -III), pp. 51- 55, April 2018.
22. Alexey Kashevnik, Roman Shchedrin, Christian Kaiser, Alexander Stocker, "Driver Distraction Detection Methods: A Literature Review and Framework," IEEE Access, vol. 9, pp. 60063 – 60076, April 2021.
23. Faiqa Sajid, [Abdul Rehman Javed,](https://ieeexplore.ieee.org/author/37088389492) Asma Basharat, Natalia Kryvinska, Adil Afzal, Muhammad Rizwan, "An Efficient Deep Learning Framework for Distracted Driver Detection," IEEE Access, vol. 9, pp. 169270-169280, December 2021.
24. Munif Alotaibi, Bandar Alotaibi, "Distracted Driver classification using deep learning," Signal, Image and Video Processing, vol. 14, pp. 617-624, November 2019.
25. Yehya Abouelnaga, Hesham M. Eraqi, Mohamed N. Moustafa, "Real-time Distracted Driver Posture Classification," NIPS Workshop on Machine Learning for Intelligent Transportation Systems, December 2018.
26. Nikka Mofid, Jasmine Bayrooti, Shreya Ravi, "Keep Your AI-es on the Road: Tackling Distracted Driver Detection with Convolutional Neural Networks and Targetted Data Augmentation," June 2020, arXiv preprint arXiv:2006.10955.
27. Achmad Yuneda Alfajr1, Kartini, Anggraini Puspita Sari, "Classification of distracted driver using support vector machine based on principal component analysis feature reduction and convolutional neural network," Jurnal Informatika dan Komputer, vol. 11, No. 2, pp. 237-245, October 2023.