Real-Time Detection and Classification of Distracted Driver using Deep Learning

Harshita Gupta
IIIT Delhi
Delhi, India
harshita19467@iiitd.ac.in

Shreya Tomar
IIIT Delhi
Delhi, India
shreya19110@iiitd.ac.in

Suyashi Singhal
IIIT Delhi
Delhi, India
suyashi19478@iiitd.ac.in

The number of road accidents has risen steadily in recent years. Distracted driving is a major cause of accidents. According to a report released by the Union Road Transport and Highways Ministry in 2016, 17 people were killed in India every hour due to road accidents. This indicated the need of the hour. We want to create a reliable and precise system that can record the image of a driver conducting multiple tasks such as texting and radio operations. The data will be collected using cameras in the cars that will capture aspects such as the face, arms, and hands. Since convolutional neural networks works great on image datasets, it is used for not only detecting the distracted driver but also for identifying the cause of distraction. We have trained multiple CNNs including scratch models as well as fine tuned pre-trained models to come up with an apt ensemble model for accurate classification. This classification model enables us to detect and alert the drivers in case of distractions and thus reduce disastrous and life threatening road accidents.

1 INTRODUCTION

Major road accidents take place due to distracted drivers. These include texting, talking on the phone, operating the radio, drinking, reaching behind, hair and makeup, talking to passengers, etc. The moment a driver's attention is off the road, accidents are likely to happen. The World Health Organization (WHO) published the 2015 Global Status Report, which revealed that about 1.25 million people die each year in car accidents around the world. Every year, distracted driving behavior kills over a million people and causes 50 million significant injuries around the world.

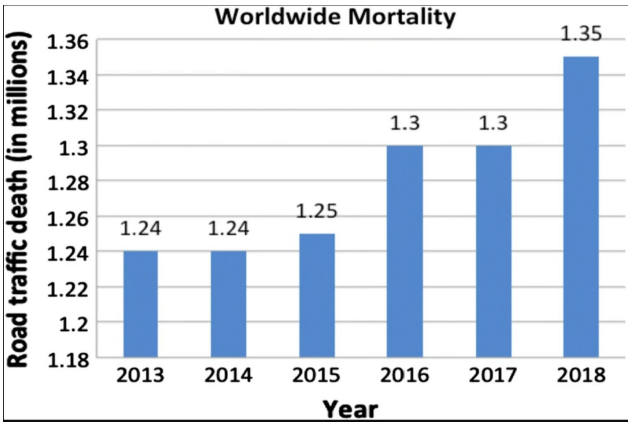

Figure 1: Worldwide Mortality due to Road Accidents

The National Highway Traffic Safety Administration (NHTSA) reported that 36,750 people died in motor vehicle crashes in 2018. Out of this, 12% was due to distracted driving. Our idea can detect the driver's action using the CNN model and can predict whether the driver is distracted in some activities or not. Hence the government can have the information about it in the form of surveillance analysis of the data. As a result, major accidents can be avoided, and our country can be a safer place.

Any activity that takes away the driver's attention from the road is distracted driving. The Centre for Disease Control and Prevention(CDC) provided a more precise definition of distracted driving by classifying it into three types, namely visual (not being visually attentive on the road ahead), cognitive (mentally distracted by eye gaze is on the road), and manual (physically distracted like driver's hands off the wheel). This project aims to develop a deep learning-based model using computer vision that can detect the distracted manual activities of the driver. The model can use images of the driver on the wheel effectively to assess and identify physical distractions like talking on the phone, eating, texting, etc.

We have included novelty in our project as compared to the existing studies. By combining two datasets into one, our project alleviates the problem of overfitting. This would provide us with additional training data and a more significant number of labelled photos. This merged dataset would be used to improve the model's generalizability and decrease overfitting. Moreover, we have added the novel technique of using MediaPipe Landmark Detection for making the model generalizable and being able to classify the actions of the driver based on images taken from various different angles.

Further, we would make the application capable of working on real-time data. It would allow a video or image to be used as an input in real-time, and if the driver is distracted for a significant amount of time, they can be warned by using some kind of alertness signal. Hence, our model would work not only for the dataset being used here but can also be integrated with a camera that captures real-time video and photographs. We would employ some metrics to suggest whether to alert the driver or not by capturing multiple frames within a few minutes of driving and generating the signal accordingly. The police can use the data from such a model to impose penalties on people not following driving rules and regulations and prevent accidents in the long run.

Figure 2: Example Images of each class

2 RELATED WORKS

In this paper[1], the best ensemble was created after averaging the probabilities generated by the models, VGG-16, VGG-19, and InceptionV3. The final log loss value obtained was 0.795. CPU resources provided by Google Cloud Platform were used. If more computing resources were available then the results could have been better. This could have been achieved by using KNN to find out the K nearest neighbours of an image and then generating a final probability by considering the average of the probabilities of these images.

In the paper[2], a CNN-based system was implemented to detect a distracted driver with the specific reason for that distraction. They modified the VGG-16 architecture by applying several regularization techniques to prevent the overfitting of the training data. The accuracy of the model was 96.31

In the paper[3], a publicly available dataset was used. Methods such as transfer learning and fine tuning were used to build a highly accurate vision based detection model. A data collection application that accessed in vehicle sensor data through a CAN bus interface and collected a multimodal dataset by performing real world trips. The model obtained on the public dataset was applied to the vision part of the multimodal dataset. The predictions of vision and sensor based modalities with two different fusion approaches to map the inputs to one of the ten actions.

In the paper[4], a method is proposed to combine the three deep learning models, the residual network, the Inception module and the hierarchical recurrent neural network to improve the performance of detecting a distracted driver's behavior.. The approach was then evaluated with the help of two types of datasets, namely, State Farm Distracted driver dataset and AUC distracted driver dataset. The images examined in this study were captured using a dashboard camera to detect distracted drivers from the 2D images. Satisfactory results were achieved from this.

3 METHODOLOGY

The flowchart in Figure 3 depicts the pipeline of our project and is followed by description of individual components of the pipeline.

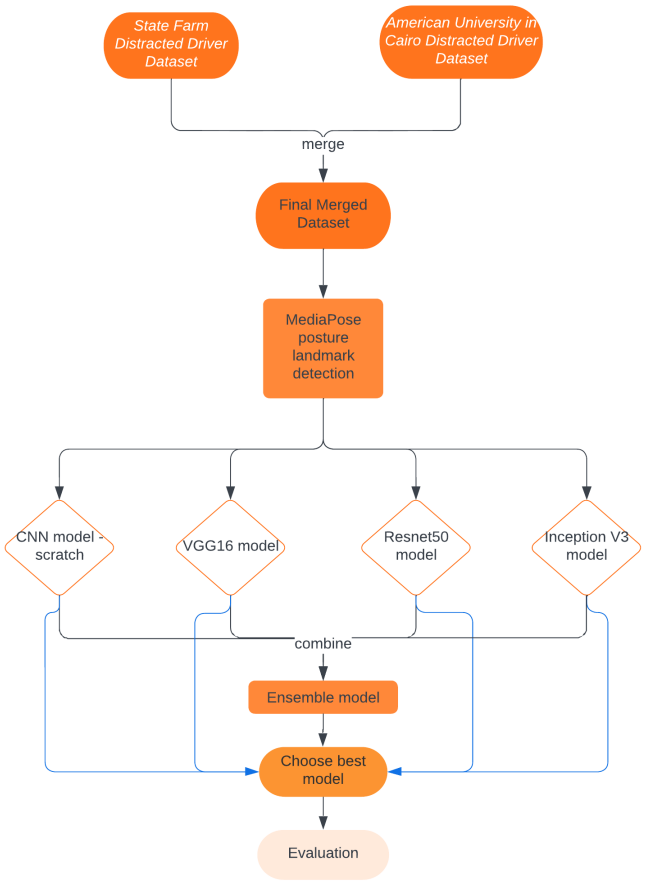

Figure 3: Model Pipeline

3.1 Dataset Creation and Description

We have a dataset of images captured from a camera mounted on the driver's dashboard. We form this dataset by combining two datasets to obtain a more extensive training set and thereby aim to avoid overfitting. The first dataset is the State Farm Distracted Driver Dataset, available on the following website - https://www.kaggle.com /competitions/state-farm-distracted-driver-detection/overview/ evaluation. In this data, the training set contains 22.4K images that have been labelled into ten classes, as given in table 1.

Class	Description
0	Normal Driving
1	Texting while driving using right hand
2	Talking to somebody on the phone using right hand
3	Texting while driving using left hand
4	Talking to somebody on the phone using left hand
5	Reaching the dashboard to operate the radio
6	Drinking or eating
7	Reaching behind
8	Fixing hair and makeup
9	Talking to passenger

Table 1: Description of Classes

The other dataset is - The American University in Cairo Distracted Driver Dataset. It contains images in the form of frames from videos captured while driving. It also includes extensive training and test sets. The combined dataset can help minimize the problem of overfitting seen in many studies on this topic. This also forms one of the novel contributions of our project. Our algorithm will detect the distracted activity of the driver and provide the most appropriate label as the output. Our dataset has performed basic preprocessing of images like creating validation sets, resizing and normalization of the images.

In a real-world scenario, the camera will capture a video of the driver while driving. In our demo we will concurrently segment the obtained video into various frames in real-time that can be used as an input to the model to detect the distracted activity of the driver. The driver can thus be alerted in case of an event of prolonged distraction. This model can be ingrained with each driving vehicle system to alert the driver in case of any distractions and prevent accidents.

3.2 MediaPipe Pose Detection

Our dataset contains images with the side view of the drivers only. In order to make the model more generalizable, we have made use of the MediaPipe Pose Detection model for predicting and drawing the pose landmarks on the image of the drivers. This would enable us to attain good performance even when the driver video is taken from a different angle than those in the dataset.

MediaPipe Posture is a machine learning approach for high-fidelity body pose tracking that uses RGB video frames to infer 33 3D landmarks and a background segmentation mask for the entire body. A two-step detector-tracker ML pipeline is used in the solution. The pipeline initially locates the person/pose region-of-interest (ROI) inside the frame using a detector. Using the ROI-cropped frame as input, the tracker then predicts the posture landmarks and segmentation mask inside the ROI. Figure 4 depicts the Landmark detection and annotation of one of the images in the dataset.

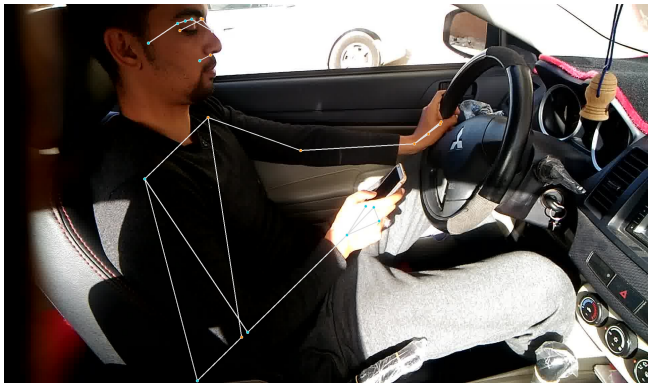

Figure 4: MediaPipe Landmark Detection and Annotation

3.3 Models

Numerous studies have used handcrafted features and Machine Learning models to train the classifier. However, the ones based on Convolutional Neural Networks have shown more promising performance. CNN's are based on the premise that a local understanding of a picture is sufficient, with the practical advantage of fewer parameters and hence require less computing time and data needed to train the model.

Training CNN models with a few layers and an extensive training set can overfit. Increasing the number of layers and training the models from scratch is computationally expensive. Hence we employ the pre-trained models since they provide us with the best of both worlds in this case. They not only contain a large number of hidden layers but also are comparatively less computationally costly as the weights in only the final layers need to be trained. The majority of the initial layers are generalized for edge and shape detection. Thus the initial layers are robust in detecting the basic pixel and patch level features like edges. The final layers are kept trainable to make them application-specific. We have trained a CNN model from scratch and used transfer learning to finetuning pre-trained models. We finally create an ensemble of the fine-tuned models to get our final results. Details about some of the models used by us are as follows -

(1) **CNN from scratch** - The convolutional neural network is a neural network specifically used for images. There are input layers, hidden layers, and an output layer. The input layer comprises the individual pixel values of the resized photos from our combined dataset, while the output layer gives the probability of each class.

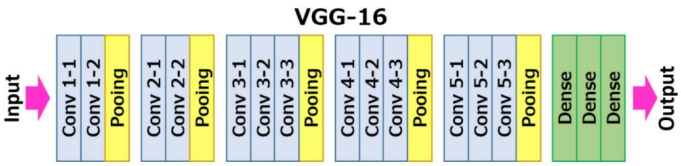

Figure 5: VGG-16 network (Example Neural Network)

(2) **VGG16** - VGG16 is a 16-layer Convolutional Neural Network with 13 convolutional layers and two fully connected layers at the top, followed by the softmax output. The most distinctive feature of VGG16 is that, rather than having a massive number of hyper-parameters, they concentrated on having 3x3 filter convolution layers with a stride one and always utilized the same padding and max pool layer of 2x2 filter. The convolution and max pool layers are arranged in the same way throughout the whole architecture.

(3) **Resnet 50** - The ResNet-50 model is a 50-layer convolutional neural network (CNN). A Residual Neural Network (ResNet) is a type of Artificial Neural Network (ANN) that builds a network by stacking residual blocks on top of each other. Neural networks with a large number of layers suffer from the problem of accuracy saturation and degradation after a point due to vanishing and exploding gradients. ResNets tackle this problem by introducing the concept of skip connections. Skip connections solve the problem of disappearing gradients by creating an alternate path for the gradient to follow. They also allow the model to learn an identity function. This guarantees that the model's upper levels perform equally well as the lower layers.

(4) **Inception V3** - The symmetric and asymmetric construction components utilized in Inception v3 include convolutions, average pooling, max pooling, concatenations, dropouts, and completely connected layers. The activation inputs are batch normalized and used frequently throughout the model. Loss is calculated using Softmax.

Ensemble Model

We ensemble the above models and perform a weighted average of the results in order to make a more accurate and generalizable classifier. Each model is given an equal weightage while combining the results. In this way, our final model combines a few deep learning models with some of the most complex architectures to perform precise categorization.

4 EXPERIMENTAL ANALYSIS

4.1 Evaluation Metrics

We used accuracy, precision, recall and F1-score as the evaluation metrics to test our models :

(1) *Accuracy*: Accuracy measures the overall efficiency of a classifier. We require that most of the fetal states are classified correctly for a good performance. It is worth noting that even predicting normal state correctly is important because wrong predictions lead to more cesarean sections.

$$\text{Accuracy} = TP / (TN + FP + FN + TP)$$

(2) *Precision*: It is the ratio of true positives to the total of the true positives and false positives. Here, it is the measure of the number of fetal states classified correctly out of all the positive samples.

$$\text{Precision} = TP / (TP + FP)$$

(3) *Recall*: It is the ability of a classifier to categorize positively labeled data. Hence, for all fetuses, it tells us how many we correctly classified.

$$\text{Recall} = TP / (TP + FN)$$

(4) *F1 score*: It is the harmonic mean between precision and recall. High F1 score enables us to get a good trade-off between precision and recall.

$$\text{F1-Score} = TP / (TP + 0.5\,(FP + FN))$$

4.2 Cost Function

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

Figure 6: Evaluation Metric (Log Loss)

We will use the multi-class cross-entropy logarithmic loss as the cost function. The log-loss is used when the performance of a classification model with a prediction input of a probability value between 0 and 1 is to be measured. Our model aims to minimise this loss. We will use a set of predicted probabilities for each image to compute the loss and optimise the model's performance. Figure 6 depicts the log-loss formula. N is the number of images in the test set, M is the number of image class labels, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, $p_{ij}$is the predicted probability that observation i belongs to class j.

4.3 Experimental Results

Table 2 lists down the accuracies of the various CNN models trained on the dataset. We observe that ResNet gives the least validation accuracy and poor performance. Hence we exclude it from our ensemble model and only use the remaining three models for ensemble.

Figure 7 and 8 demonstrates how the model's accuracy and categorical cross entropy (loss) vary over epochs when the model is trained. Both the training and validation accuracies in Figure 7 increase with the number of epochs until they reach a particular threshold, indicating that no more improvements can be made and training must be halted. Similarly, when the model learns, both

Models	Training loss	Training accuracy	Validation loss	Validation accuracy
CNN	0.0339	0.9903	0.0337	0.9935
Inception	0.1038	0.9814	0.0841	0.9801
Resnet-50	4.6218	0.8750	3.7207	0.4913
VGG	0.1710	0.9548	0.3300	0.9137

Table 2: Performance of models

validation and training losses decrease, eventually disappearing when the model begins to overfit.

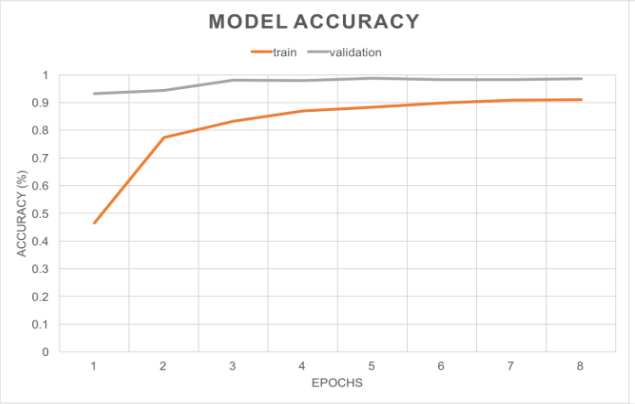

Figure 7: Ensemble Model Accuracy

Moreover, these findings indicate that training accuracy is significantly greater than validation accuracy. This leads to overfitting. Dropout layers are used to prevent overfitting. We see this behaviour because the model has a small number of layers for a big amount of training data. It causes the model to not fit accurately on the training set.

The Precision, Recall and F1 scores are comparable to the accuracies of the model. For instance for the CNN scratch model, the F1 score was 99.33, precision and recall were approximately 99.3. Similar inferences can be drawn for them as well.

4.4 Demo

We made a demo video by sitting in a car and mimicking distracted activities as well as actions of normal driving. We divided this video into various frames and classified each frame using the above model. We alert the driver using multiple beep sounds in case of prolonged distraction to prevent accidents. Figure 9 depicts a frame from one such video.

5 CHALLENGES

Road safety is a significant concern worldwide. The number of road accidents has only increased in the past years. According to a survey conducted by the National Highway Traffic Safety Administration, distracted drivers are responsible for roughly one-fifth of

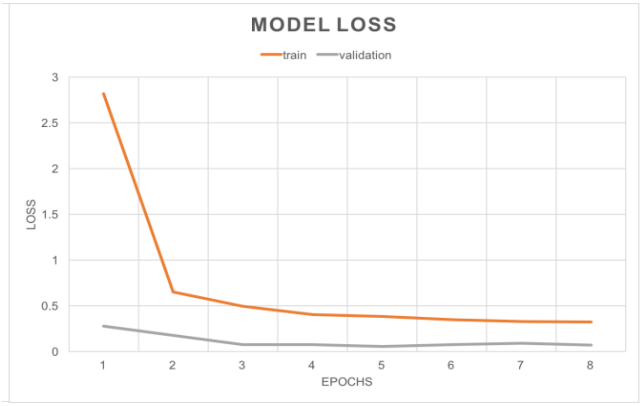

Figure 8: Ensemble Model Loss

Figure 9: Frame from Demo video

all motor vehicle accidents, and Indian roadways account for the most significant number of fatal accidents globally. Daily, around nine people are killed in the US, and more than 1,000 are wounded in traffic accidents involving a distracted driver. The situation can be especially problematic for individuals driving alone or those driving at night or late to reach somewhere.

Prevention is always better than cure, especially when someone's life is at stake. We aim to develop a robust and accurate system that can capture the driver's video while driving and perform real-time classification of their actions. These models can help prevent accidents to a great extent. The algorithm can be integrated as part of the vehicle driving system. Thus, if the car could identify such distractions and notify the driver, it might reduce road crashes.

One of the significant technical challenges in developing such a model is the overfitting of the deep learning network. Since these neural networks have very complex architectures with many parameters, more often than not, they overfit. Most researchers in this domain have mentioned this as a drawback of their model. This

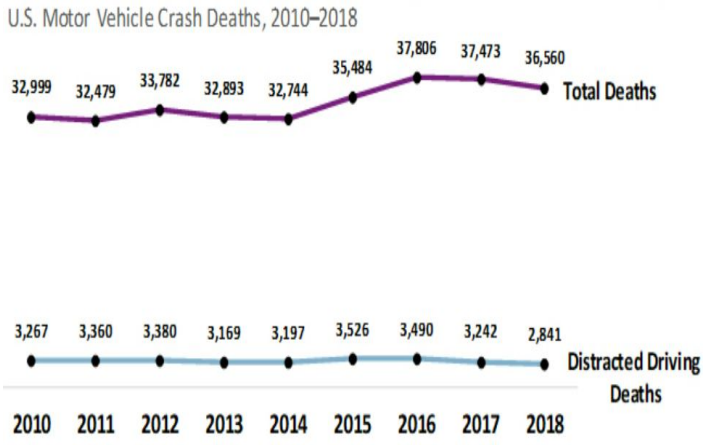

Figure 10: US Motor Vehicle Crash Deaths due to Distracted Driving

reduces the generalizability of the model and hence diminishes its performance. Moreover, training large neural networks from scratch requires a lot of computational power that might not be available readily. There is an added problem of data leakage encountered while using the dataset. With slight changes, multiple images of the same person within the training dataset leaked into the validation set. This can lead to the model being trained on the same information it would predict. Generation of blurry frames due to the vehicle's movement and problems of using cameras in two-wheelers are some of the other technical challenges that need to be overcome while dealing with this problem.

6 CONTRIBUTIONS

The main aim is to improve the performance of detecting drivers' distracting actions. A dashboard camera capable of detecting distracted drivers through 2D camera images is used as the dataset to capture the distracted driver's actions. We will perform high-performing CNN training on a very large number of images. It allows us to learn good low-level features extraction, which applies to our dataset. Higher-level features are less but are suitable for fine-tuning. The accuracy of the distracted driver classification is improved by using transfer learning and combining various deep learning models to develop an ensemble of convolutional neural networks. The final evaluation metric will be generated by averaging out the results of all the models.

The data of the distracted drivers can be useful for the government to take control over reckless driving as well as real-time alerting systems for the driver itself. Good accuracy is maintained using a Convolutional Neural Network-based approach and also maintaining a good computational complexity. The memory requirements are reduced to tackle real-world applications.

7 CONCLUSION

Through this study, deep learning model techniques of both from scratch and pre-trained models are used. The models used are Resnet-50, VGG-16, Inception, and a convolutional neural network model from scratch. ResNet, AlexNet, and VGG-16 are just a handful of the Deep Convolutional neural network models that have been pre-trained on ImageNet. We have achieved almost perfect accuracy, precision, recall, and F1 score metrics of around 90-98% in 3 out of 4 models. In the case of the Resnet-50 we get the values of the prediction metrics of around 49% for 10-class prediction into classes from 0 to 9, depicting normal driving, texting -right, taking on the phone -left, etc. This reiterates that the feature engineering performed by us has performed well.

Our research used photographs from the State Farm Distracted Driver Detection[4] competition on Kaggle to solve the problem of distracted driver detection. The model achieved 98.4% accuracy on test data by leveraging the pre-trained VGG16 network, Resnet-50, Inception, and our scratch model.

Further analysis showed that the class which was mis-classified wrong in most of the classes was reaching behind. It is often confused with the driver talking on their phone with the right hand. The overall results were successful and effective at predicting driver who gets distracted while driving. If everything goes according to plan, these results could be proven to be very effective in preventing further accidents and deaths resulting from distracted driving. In the future, we plan to fine-tune a few of the VGG16 network's lower layers by freezing them, while we retrain the remaining ones on the dataset.

For further studies, we are attempting to reduce the amount of parameters which would eventually reduce the computation time. The accuracy can be increased by incorporating temporal context which may aid in the reduction of misclassification mistakes. We also want to develop a system that can identify visual and cognitive distractions. This would bring a new change to our society in the future.

REFERENCES

[1] Prof. Pramila M. Chawan, Shreyas Satardekar, Dharmin Shah, Rohit Badugu, Abhishek Pawar. 2018. *Distracted Driver Detection and Classification.* [online] Available at: https://www.researchgate.net/profile/Pramila-Chawan/publication/337275106_Distracted_Driver_Detection_and_Classification/links/5dce5afc299bf1b74b426c58/Distracted-Driver-Detection-and-Classification.pdf

[2] Bhakti Baheti, Suhas Gajre, Sanjay Talbar. 2018. *Detection of Distracted Driver using Convolutional Neural Network.* Center of Excellence in Signal and Image Processing, SGGS Institute of Engineering and Technology, Nanded, Maharashtra, India https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w14/Baheti_Detection_of_Distracted_CVPR_2018_paper.pdf

[3] Furkan Omerustaoglu, C. Okan Sakar, Gorkem Kar. 2020. *Distracted driver detection by combining in-vehicle and image*

data using deep learning. https://sci-hub.hkvisa.net/10.1016/j.asoc.2020.106657

[4] Kaggle.com. 2016. *State Farm Distracted Driver Detection | Kaggle*. [online] Available at: https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data

[5] Distracted Driver Detection using Deep Learning https://towardsdatascience.com/distracted-driver-detection-using-deep-learning-e893715e02a4