NEGLEGENT DRIVER

### A Project Work Report

*Submitted in the partial fulfillment for the award of the degree of*

# BACHELOROFENGINEERING

### IN

**COMPUTER SCIENCE WITH BIG DATA ANALYTICS**

### Submitted by:

**Parminder Singh**

**19BCS3757**

### Under the Supervision of:

**KUNDAN MUNJAL**



### CHANDIGARHUNIVERSITY,GHARUAN,MOHALI-140413,

**PUNJAB**

#### APRIL2021

# TableofContents

TitlePage

Abstract

#### INTRODUCTION

1. LITERATURE REVIEW
2. PROBLEMS FACED
3. DATASET DESIGN
4. PROPOSED METHOD
5. CONCLUSION
6. RESEARCH OBJECTIVES
7. **METHODOLOGY**
8. **REFRENCES**

**ABSTRACT**

The world Health Organization (WHO) reported 1.35 million deaths each year as result of traffic crashes. The digits mentioned are growing rapidly over past few years. Almost one fifth of these accidents are caused by distracted drivers. With such rapid increasing population and thereby increasing vehicle counts, the numbers will eventually grow out of the limits. The growth in technologies is prime suspect of being distracted.

Approximately drivers using mobile phones are 4 times more likely to be involved in a crash than one not using mobile phones. To avoid such tremendous and fatal accidents we propose a deep learning and machine learning based solution to achieve a very high rate of success in avoiding such reckless accidents. We also study the effects of different visual elements in distraction detection by means of facial recognition, hand localization and skin segmentation. And hereby we present our version of ensemble with high classification accuracy and wield real time environment.

# INTRODUCTION

Amid these 20 years starting from 2000 to 2020, there has been a 35.7% increase in the number of licensed drivers in India, reaching a total of 190.6 million licenses. From 2000 to 2020, the miles covered by vehicles incremented by 80% while the urban vehicles' miles traveled increased by 80%, while road construction increased by only 37%. The increase in the number of roads lagged behind to the number of vehicles, hence the gridlock. Besides, In-Vehicle Information Systems (IVISs) namely the navigations and technologies cause more distractions and thus lead to more accidents.

Regardless of all the road improvements and vehicle upgrades, lethal crashes persist. According to the Road Accident Report for 2019, a total number of 449,002 accidents took place in the country during the calendar year 2019 leading to 151,113 deaths and 451,361 injuries. The 2019 Global Status Report of the World Health Organization (WHO) reported approximately 1.86 million annual deaths due to road traffic accidents worldwide. The same year, the numbers of global deaths due to Hepatitis and HIV are estimated to be around 1.6 million and 1.3 million respectively, which is almost the same as the number of people dying annually due to road traffic accidents.

According to the National Highway Traffic Safety Administration (NHTSA), almost a fifth of traffic accidents is caused by reckless, distracted drivers. Driver error assists dominantly in these vehicle crashes. In 2019, 4879 people were killed, and roughly 434,000 were injured in vehicle crashes concerning distracted drivers. Mobile phone usage is a major cause of these accidents.

Any activity that diverts attention from driving viz.

1.talking or texting on one’s phone

2. eating or drinking,

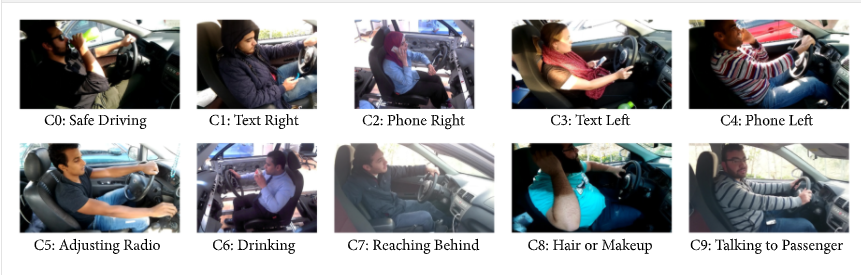
3. talking to passengers, or

4. fiddling with the stereo, entertainment, or navigation system

is distracted driving as determined by NHTSA. Furthermore, a broader view as provided by The Center for Disease Control and Prevention (CDC) states visual (i.e., taking one’s eyes off the road), manual (i.e., taking one’s hands off the driving wheel), and cognitive (i.e., taking one’s mind off driving) causes is also distracted driving.

Developing distraction mitigation systems that adapt IVIS functions in agreement with the driver’s state might evade the problem of distracted driving. The main focus of these systems would be to accurately recognize the driver’s distraction. Nonetheless, these detection systems can aid law enforcement in identifying such distractions on the highway and penalize accordingly. Detecting distractions can also be used to enable Advanced Driver Assistance Systems (ADAS) features like Collision Avoidance Systems (CAS) that have to plan elusive maneuvers.

The research done in the field of negligent driving determine manual, visual, or psychological sorts of distractions. Cognitive distraction simply deals with the task of chattering, listening or just hallucinating or fantasizing. This kind of distraction can harm the driver despite the fact that they are in protected driving stance just because they are “mentally” distracted. Visual interruptions regularly allude to circumstances where the driver takes their eyes off the street due to either "the presence of striking visual data away from the street causing impulsive eye glances and fleeting rotations of the head" or the utilization of entertainment devices such as a phone, tab, or radio. Visual distractions are instituted in the accompanying terms: "lethargy", "sleepiness", "exhaustion", and “distraction”. Furthermore, they for the most part rely upon facial landmarks detection and following. Manual distractions are for the most part worried about driver's exercises other than safe driving (i.e., spanning behind, fixing hair and makeup, or eating and drinking).In this sort of diversions, creators will in general rely vigorously upon hand tracking and driving posture assessment. In this paper, we center just around "manual" distractions/ negligence where a driver is diverted by messaging or using mobile phone, calling, eating or drinking, reaching behind, tinkering with the radio, fixing hair and makeup, or conversing with someone.



***Figure 1.1***

You must have heard about the self-driving cars in which the passenger can fully depend on the car for traveling. But to achieve level 5 autonomous, it is necessary for vehicles to understand and follow all traffic rules.

In the world of Artificial Intelligence and advancement in technologies, many researchers and big companies like Tesla, Uber, Google, Mercedes-Benz, Toyota, Ford, Audi, etc are working on autonomous vehicles and self-driving cars. So, for achieving accuracy in this technology, the vehicles should be able to interpret traffic signs and make decisions accordingly.

In this Python project example, we will build a deep neural network model that can classify traffic signs present in the image into different categories. With this model, we are able to read and understand traffic signs which are a very important task for all autonomous vehicles.

# PROBLEM FACED

1.The dataset we considered is suitable for the left-hand-drive that is driving on the right side of the road with the steering wheelon the left side of the car. India being a former British Colony inherited its right-hand driving system, hence while capturing the image we are flipping it and using the mirror image for processing.

2.Hardware installation- The dataset we used has the camera installed at a specific angle. While installing the camera in real-time we have to prop it at that particular angle for more effective image processing.

# 3. LITERATUREREVIEW

The work in the distracted driver detection field over the past seven years could be clustered into four groups: multiple independent cell phone usage detection publications, Laboratory of Intelligent and Safe Automobiles in University of California San Diego (UCSD) datasets and publications, Southeast University Distracted Driver dataset and affiliated publications, and recently StateFarm’s Distracted Driver Kaggle competition

**3.1. Cell Phone Usage Detection**

SVM-based model detects the use of mobile phone while driving (i.e., distracted driving). Their dataset consists of frontal image view of a driver’s face. However, their dataset is collected from transportation imaging cameras that are deployed in highways and traffic lights, which is, indeed, more competitive. Driver distraction detection and recognition using RGB-D sensor, AdaBoost classifier and Hidden Markov Models to classify a Kinect’s RGB-D data. Their solution depends on indoor-produced data. They sit on a chair and a mimic a certain distraction (i.e., talking on the phone). This setup misses two essential points: the lighting conditions and the distance between a Kinect and the driver. In real-life applications, a driver is exposed to a variety of lighting conditions (i.e., sunlight and shadow). Reference suggests using a Hidden Conditional Random Fields (HCRF) model to detect cell phone usage. Their model operates face, mouth, and hand features of images obtained from a camera mounted above the dashboard. Reference devised a Faster-RCNN model to detect driver’s cell phone usage and “hands on the wheel”. Their model is mainly geared towards face/hand segmentation. They train their Faster-RCNN on the dataset proposed in (that we also use in this paper). Their proposed solution runs at a 0.06 and 0.09 frames per second for cell phone usage and “hands on the wheel” detection. Reference tackles the problem of cell phone usage detection. Their approach does not hold any static assumptions though (i.e., in which region of the image a face is expected to be found). They use a Supervised Descent Method (SDM) to localize the face landmarks and, then, extract two bounding boxes to the left and the right side of the face. They train a classifier on each of the two regions to detect cell phone usage: right hand, left hand, or no usage. Using a histogram of gradients (HOG) and an AdaBoost classifier, they achieve a 93.9% classification accuracy and operate in a near real-time speed (7.5 frames per second).

**3.2. UCSD’s Laboratory of Intelligent and Safe Automobiles Work**

Reference presents a vision-based analysis framework that recognizes in-vehicle activities using two Kinect cameras that provide frontal and back views of the driver. Their approach provides “hands on the wheel” information (i.e., left hand only, both hands, no hands) and uses this information to detect three types of distractions: adjusting the radio, operating the gear, and adjusting the mirrors. Reference presents a fusion of classifiers where the image is to be segmented into three regions: wheel, gear, and instrument panel (i.e., radio). It proposes a classifier for each segment to detect existence of hands in those regions. The hand information (i.e., output of the classifiers) is passed to an “activity classifier” that infers the actual activity (i.e., adjusting the radio, operating the gear). Reference extends existing research to include eye cues to previously existing head and hands cues. However, it still considers three types of distractions: “wheel region interaction with two hands on the wheel, gear region activity, and instrument cluster region activity”. Reference presents a region-based classification approach. It detects hands presence in certain predefined regions in an image. A model is learned for each region separately. All regions are later joined using a second-stage classifier.

**3.3. Southeast University Distracted Driver Dataset**

Reference designs a more inclusive distracted driving dataset with a side view of the driver and more activities: grasping the steering wheel, operating the shift lever, eating a cake, and talking on a cellular phone. It introduces a contourlet transform for feature extraction and, then, evaluates the performance of different classifiers: Random Forests (RF), -Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). The random forests achieved the highest classification accuracy of 90.5%. Reference showed that using a multiwavelet transform improves the accuracy of Multilayer Perceptron classifier to 90.61%. Reference showed that using a Support Vector Machine (SVM) with an intersection kernel, followed by Radial Basis Function (RBF) kernel, achieved the highest accuracies of 92.81% and 94.25%, respectively. After testing against other classification methods, they concluded that an SVM with intersection kernel offers the best real-time quality (67 frames per second) and better classification performance. Reference improves the Multilayer Perceptron classifier using combined features of Pyramid Histogram of Oriented Gradients (PHOG) and spatial scale feature extractors. Their Multilayer Perceptron achieves a 94.75% classification accuracy.Utilizes motion history images (HMI) to make use of the data’s temporality. Pyramid Histogram of Gradients (PHOG) is applied to the motion history images. A Random Forrest trains on the extracted features and yields a 96.56% accuracy. Reference presents a convolutional neural network solution that achieves a 99.78% classification accuracy. They train their network in a 2-step process. First, they use pretrained sparse filters as the parameters of the first convolutional layer. Second, they fine-tune the network on the actual dataset. Their accuracy is measured against the 4 classes of the Southeast dataset.

**3.4. StateFarm’s Dataset**

StateFarm’s Distracted Driver Detection competition on Kaggle was the first publicly available dataset for posture classification. In the competition, StateFarm defined ten postures to be detected: safe driving, texting using right hand, talking on the phone using right hand, texting using left hand, talking on the phone using left hand, operating the radio, drinking, reaching behind, doing hair and makeup, and talking to passenger. Our work, in this paper, is mainly inspired by StateFarm’s Distracted Driver’s competition. While the usage of StateFarm’s dataset is limited to the purposes of the competition , we designed a similar dataset that follows the same postures.

**3.5. Traffic signs and signal recogonition**

The German Traffic Sign Benchmark is a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011. We cordially invite researchers from relevant fields to participate: The competition is designed to allow for participation without special domain knowledge. Our benchmark has the following properties:

* Single-image, multi-class classification problem
* More than 40 classes
* More than 50,000 images in total
* Large, lifelike database

**4. DATASET DESIGN**

Creating a new dataset was essential to the completion of this work. The available alternatives to our dataset areStateFarm and Southeast University (SEU) datasets. StateFarm’s dataset is to be used for their Kaggle competition purposes only (as per their regulations). As for Southeast University (SEU) dataset, it presents only four distraction postures. And, after multiple attempts to obtain it, we figured out that the authors do not make it publicly available. All the papers that benchmarked against the dataset are affiliated with either Southeast University, Xi’an Jiaotong-Liverpool University, or Liverpool University, and they have at least one shared author. With that being said, the collected “distracted driver” dataset is the first publicly available (obtainable after signing a license agreement) for driving posture estimation research. Our dataset is publicly available, subject to signing our agreement form from. The dataset introduced in this work is an extended and cleaned-up version of our dataset presented in.

**4.1. Camera**

Our dataset collection setup has a single camera with a fixed perspective, and the data collection was conducted on two phases. In each phase, a different camera is used. In one phase we use the rear camera, and in the other phase we used the LAPTOP camera. The latter camera provides depth information, but we only record the RGB images. Collecting data from different cameras adds an extra dimension of diversity to our dataset, and we demonstrate the feasibility of effective distraction detection by relying on RGB cameras which are widely available and of low cost.

The data was collected in a video format and then cut into individual images, 1080 x 1920 or 640 x 480 each. The cameras are fixed using an arm strap to the car roof handle on top of the front passenger’s seat. In our use case, this setup proved to be very flexible as we needed to collect data in different vehicles.

**4.2. Labeling**

*I*n order to label the collected videos, we designed a simple multiplatform action annotation tool using modern web technologies: Electron, AngularJS, and JavaScript.

##### 

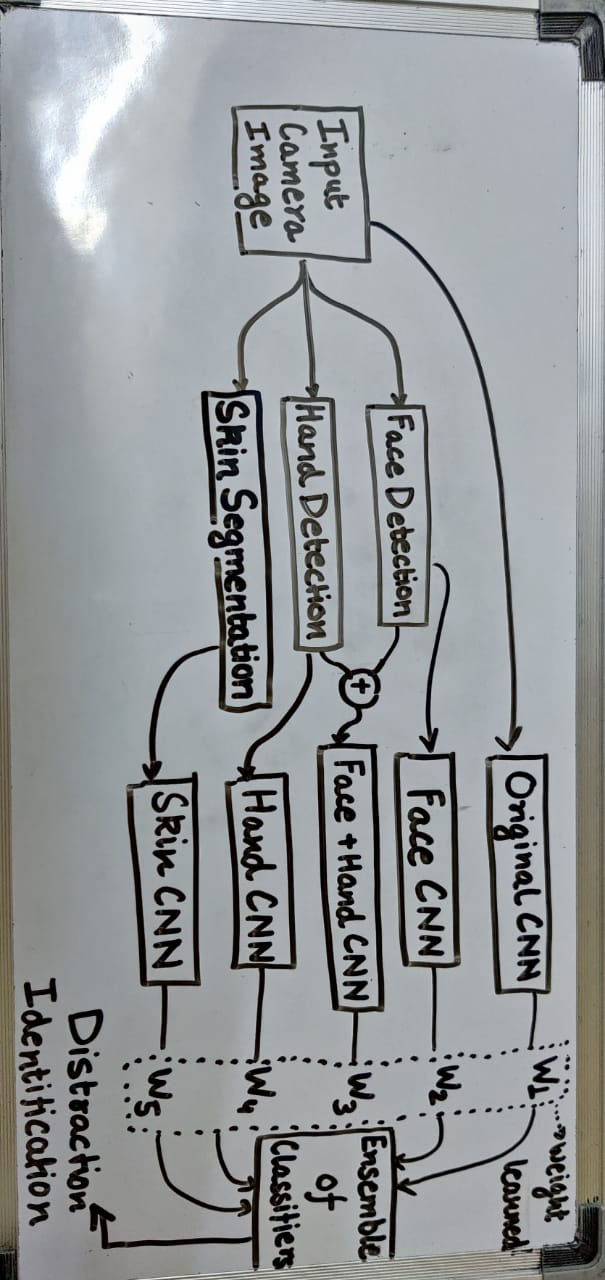
##### Figure 4.1

**4.3. Statistics**

We had 44 participants from 7 different countries: Egypt (37), Germany , USA , Canada , Uganda , Palestine , and Morocco . Out of all participants, 29 were males and 15 were females. Some drivers participated in more than one recording session with different time of day, driving conditions, and wearing different clothes. Videos were shot in 5 different cars: Proton Gen 2, Mitsubishi Lancer, Nissan Sunny, KIA Carens, and a prototyping car. We extracted 14,478 frames distributed over the following classes: safe driving (2,986), phone right (1,256), phone left (1,320), text right (1,718), text left (1,124), adjusting radio (1,123), drinking (1,076), hair or makeup (1,044), reaching behind (1,034), and talking to passenger (1,797). The sampling is done manually by inspecting the video files with eye and giving a distraction label for each frame. The transitional actions between each consecutive distraction types are manually removed. Figure [1](https://www.hindawi.com/journals/jat/2019/4125865/fig1/) shows samples for the ten classes in our dataset.

**5. Proposed Method**

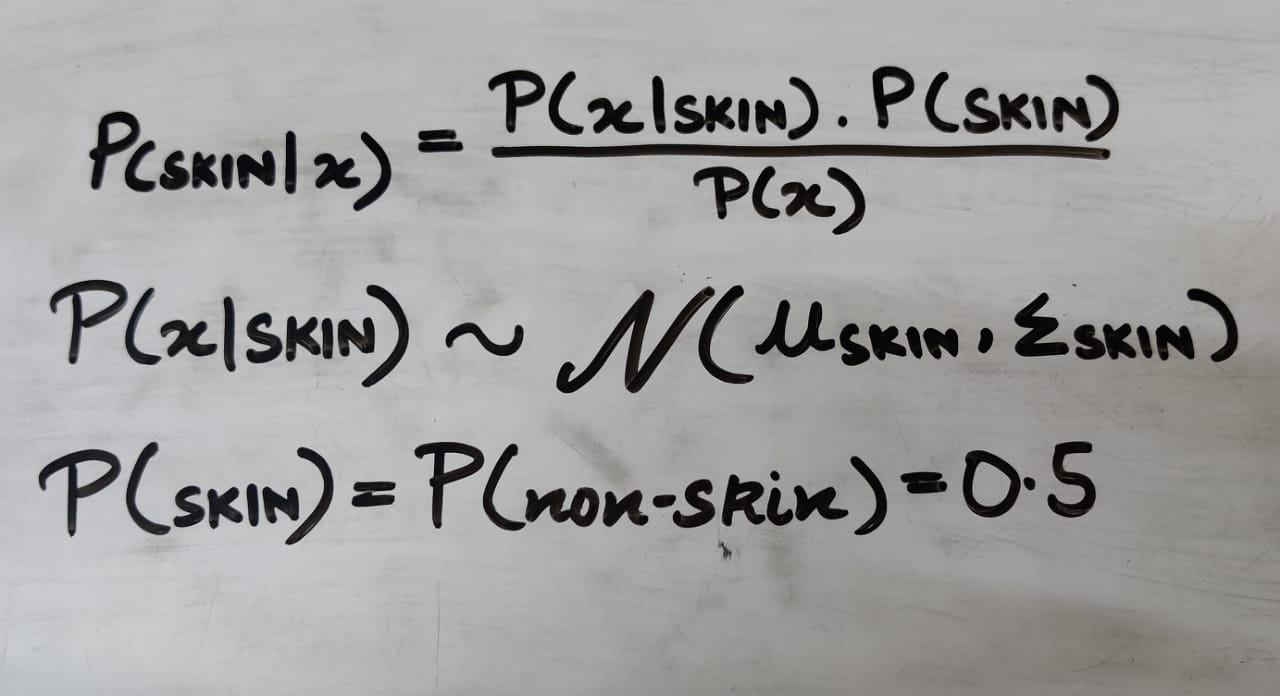
Our proposed solution consists of a genetically weighted ensemble of convolutional neural networks. The convolutional neural networks are trained on raw images, skin-segmented images, face images, hands images, and “face+hands” images. We fine-tune a pretrained ImageNet model (i.e., transfer learning) for these networks. Then, we evaluate a weighted sum of all networks’ outputs yielding the final class distribution using a genetic algorithm.



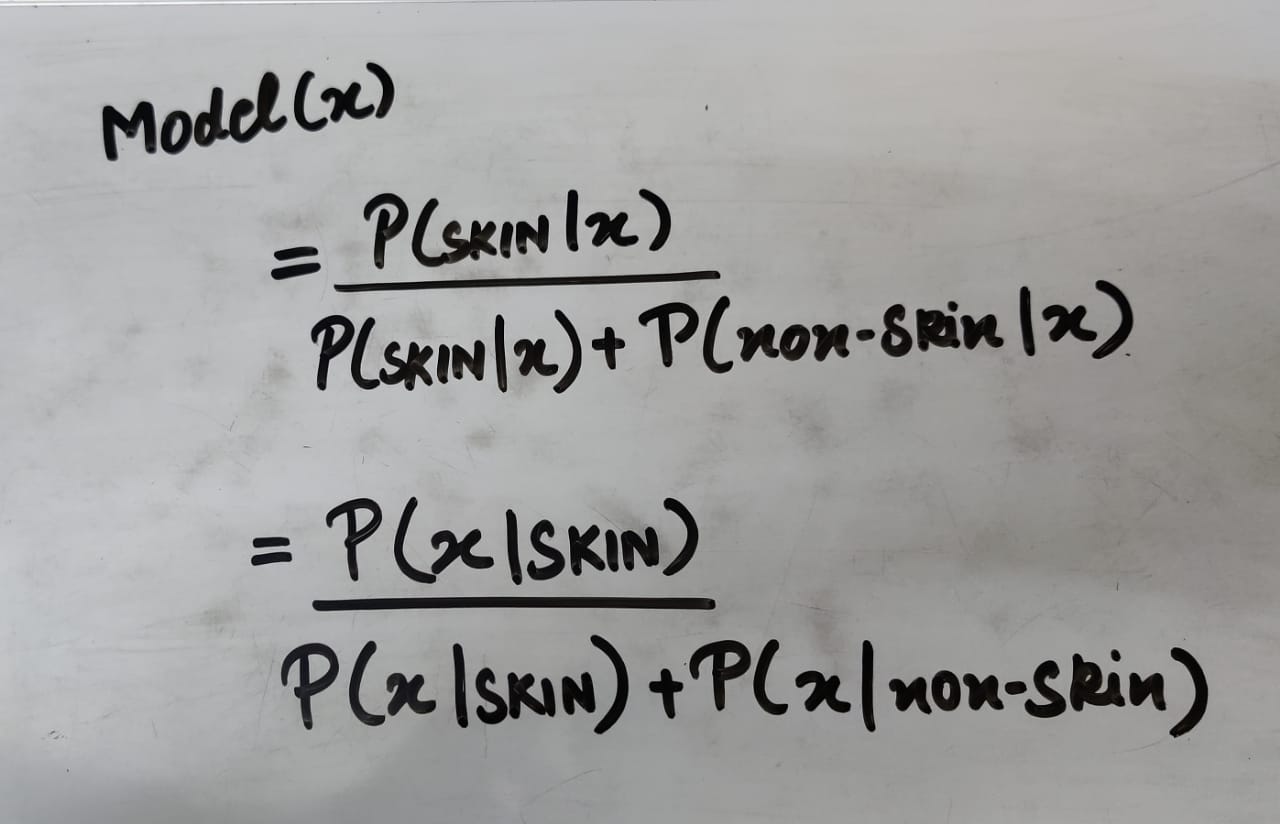
**Figure 5.1**

**5.1 Skin Segmentation**

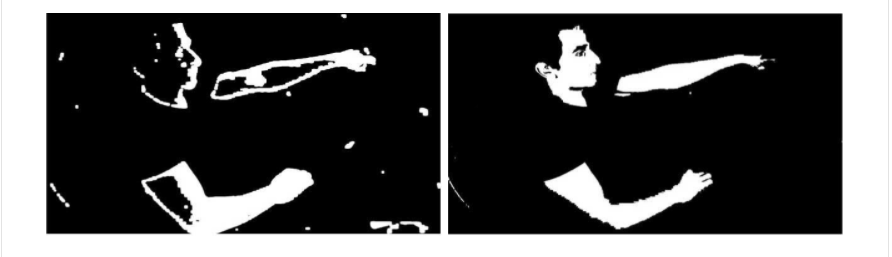
Skin segmentation is a challenging problem to solve, mainly due to the different lighting conditions happening during driving. We use a Multivariate Gaussian Naive Bayes classifier to develop a pixel-wise skin segmentation model. Our model is similar to except that we do not use a histogram as a Likelihood function. Instead, we fit the training data into Gaussian distributions to formulate the Likelihood functions. The posterior probability is evaluated as in.

****

We trained our model using the UCI Skin Segmentation dataset. The database contains RGB colors that are labeled for the skin and non-skin classes. It is generated using skin textures from face images of people from diverse ages, genders, and races. It contains a total of 245,057 color samples, out of which 50,859 are the skin samples and 194,198 are non-skin samples. Two Gaussian distributions (Likelihoods) are constructed for the skin and the non-skin classes by estimating   from the training data. For deployment phase, each pixel x in the input image is fed to the model as in . And then, a probability heatmap of skin in the image can be constructed. We classify a pixel to a “skin” if . Then, we cluster the skin pixels into objects and remove those with a small number of pixels, because neither faces nor hands skin blobs are expected to have small number of pixels.

****

One key disadvantage of such method is that it is very sensitive to image illumination conditions. Hence, incorporating pixel location can improve the skin classification accuracy. One way is to pass the pixel location to the input feature vector. However, to the best of our knowledge, there is no available dataset to train and evaluate such method. Besides, annotating a new dataset is costly. Therefore, we adopt an active learning-based approach to supervise the training. The above classifier (without pixel spatial information) is run against all training images to generate skin masks. Generated masks are manually inspected to cherry-pick samples with high skin segmentation accuracy. Those pixels are used as new training data of the proposed skin segmentation classifier, such that the feature vector includes pixels spatial information (X- and Y-coordinates within the images) in addition to the color information (Red, Green, and Blue color components). Figure 4.2 shows a sample skin-segmented image with (right) and without (left) pixel spatial information. We notice an accuracy improvement after considering the pixels spatial information.



**Figure 5.2**

**5.2. Face and Hands Detection**

We trained the model presented in on the Annotated Facial Landmarks in the Wild (AFLW) face dataset . It was sensitive to distance from the camera; faces that were close to the camera were not easily detected. We found that the pretrained model produced better results on our dataset. Given that we did not have any hand labeled face bounding boxes, we could not formally compare the two models. But it was obvious that gives a better detection accuracy based on inspecting the results manually. However, face misdetections are noticed in several examples, mainly because the detector is not trained to handle non-frontal faces.

As for hands detection, we used the pretrained model with modifications. Their trained model was a binary class AlexNet that classifies hands/non-hands for different proposal windows. We transferred the weights of the fully connected layers into convolutional layers such that each neuron in the fully connected layer was transferred into a feature map layer with a 1-pixel kernel size. Our proposed architecture, accepts variant-size inputs and produces variant-size outputs. The last convolutional layer has a depth of 2 (i.e., the binary classes), and for each pixel the summation of the two depths is one where W and H are the output’s width and height, respectively.

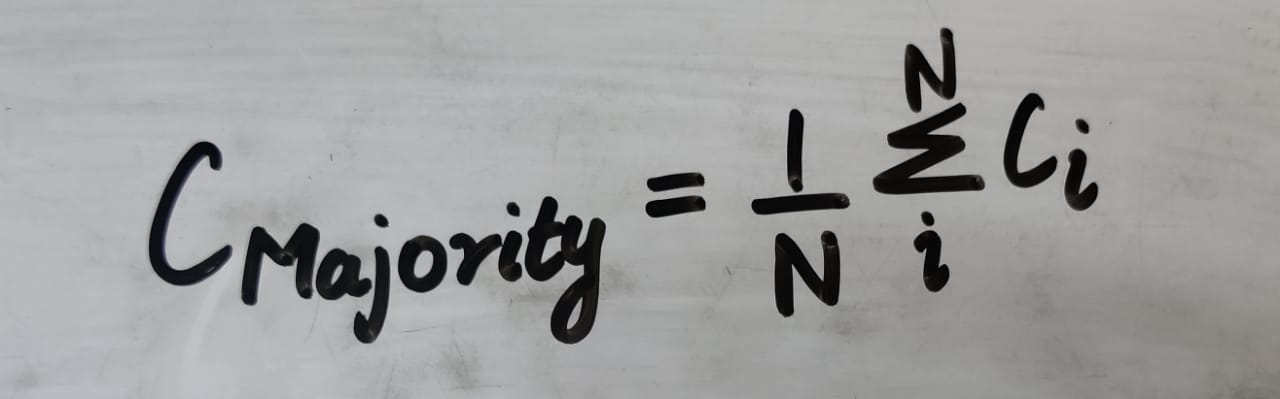
**5.3. Convolutional Neural Network**

For distracted driver posture classification, we trained and benchmarked different neural networks architectures: Each network is trained on 5 different image sources (i.e., raw, skin, face, hands, and face+hands images).

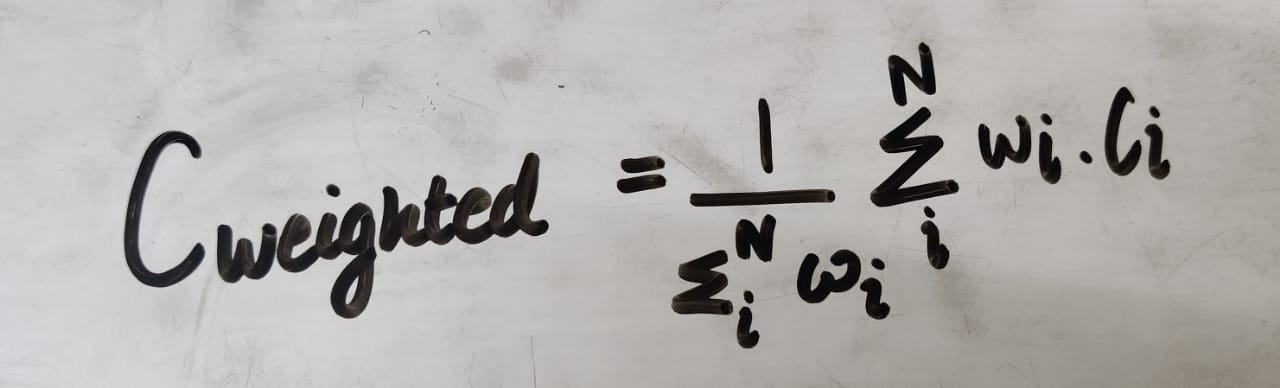
We trained our VGG-16 models from scratch. We did not use a pretrained model. As for InceptionV3, we performed transfer learning. We fine-tuned a pretrained model on the distraction postures. We removed the “logits” fully connected layer and replaced it with 10-neuron fully connected layer (i.e., corresponding to 10 driving postures). For all of our models, we used a gradient descent optimizer with an initial learning rate of 10-2. The learning rate decays linearly in each epoch with a step of 10-2 – 10-4. We trained the networks for 25 epochs. In each epoch, we divide the training dataset into minibatches of 40 images each.

**5.4. GA-Based Ensemble of Classifiers**

Each classifier produces a class probability vector (i.e., output of the “softmax” layer), C1…CN , such that Ci = R10is a vector having 10 probabilities (for 10 distraction classes) and  is the number of classifiers. it is assumed that all experts (i.e., classifiers) can equally contribute to a better decision by taking the unweighted sum of all classifier outputs.



However, that is not usually a valid assumption. In a weighted voting system we assume that classifiers do not contribute equally to the ensemble and that some classifiers might yield higher accuracy than others. Therefore, there is a need to estimate the weights of each classifier’s contribution to the ensemble. We opted to use a genetic algorithm (i.e., a search-based method).

****

In our genetic algorithm, a chromosome consists of N genes that correspond to the weights w1…wN . Our fitness function evaluates the Negative Log Likelihood (NLL) loss over a 50% random sample of the population. This helps prevent overfitting. Our population consists of 50 individuals. In each iteration, we retain the top 20% of the population and use them as parents. Then, we randomly select 10% of the remaining 80% of the population as parents. In other words, we have 30% of the population as parents. Now, we randomly mutate 5% of the selected parents. Finally, we cross-over random pairs of the parents to produce children until we have a full population (i.e., with 50 individuals). We ran the above procedure for only 5 iterations in order to avoid overfitting. We selected the chromosome with the highest fitness score (test against all data points, not 50%).

**5.3. Our Dataset**

For training and testing, GTSRB dataset contains 51839 images in 43 classes. We have selected 39,209 images for training and rest for testing. Images with deformation due to viewpoint variation, occlusion due to obstacles like trees, building etc.,natural degrading, weather condition are considered in this dataset. We have resized all input images to 128 × 128 using cubic interpolation method.

Our ‘train’ folder contains 43 folders each representing a different class. The range of the folder is from 0 to 42. With the help of the OS module, we iterate over all the classes and append images and their respective labels in the data and labels list.

**6. CONCLUSION**

Distracted driving is a major problem leading to a striking number of accidents worldwide. Its detection is an important system component in semiautonomous cars. In this paper, we presented a robust vision-based system that recognizes distracted driving postures. We collected a novel publicly available distracted driver dataset that we used to develop and test our system. Our best model utilizes a genetically weighted ensemble of convolutional neural networks to achieve a 90% classification accuracy. We aim to provide a baseline performance for future research to benchmark against. Face, hands, and skin detection proved to improve classification accuracy in our ensemble. However, in a real-time setting, their performance overhead is much higher than their contribution.

Moreover, With more data to train around the vehicles the driver can be more cautious of their surrounding and leading to more safety on the roads

In a future work, we need to devise a better face, hands, and skin detector. We would need to manually label hand and face proposals and use them to train a Fast-RCNN (or, any other object detector) to localize both faces and hands in one shot and evaluate it against our existing CNN-based localization method. Also to create a software that enables autonomous driving when the driver is distracted for a certain period of time.

# 7. RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of an approach for NEGLECTED DRIVER The proposed aim will be achieved by dividing the work into following bjectives:

1. recognize unsafe behavior
2. To make roads easier to travel with less percentage of accident happenings
3. As the interest and curiousity of people is increasing in self driving cars, a negligent driver would probably be more prone to accidents. So, this research would make these cars more safer.
4. send real time feedback to the driver using short sound alert

# 8. METHODOLOGY

Thefollowingmethodologywillbefollowedtoachievetheobjectivesdefinedforproposedresearchwork:

1. Detailedstudyof NEGLECTED DRIVER will bedone.
2. Installationandhandonexperienceonexistingapproachesof DISTRACTED DRIVER will bedone.Relativepros and cons will beidentified.
3. Variousparameterswillbeidentifiedtoevaluate the proposed system.
4. Comparison ofnew implemented approach with exitingapproacheswillbedone.

# 9.BIBLIOGRAPHY

# https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

# https://keras.io/

# https://docs.opencv.org/master/d0/de3/tutorial\_py\_intro.html

# https://analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

# https://www.tensorflow.org/tutorials/images/cnn

# https://likegeeks.com/kivy-tutorial/

# https://neurohive.io/en/popular-networks/vgg16/

# https://www.javatpoint.com/pytorch-testing-of-convolutional-neural-network-model

# https://www.kaggle.com/c/state-farm-distracted-driver-detection

# https://towardsdatascience.com/distracted-driver-detection-using-deep-learning-e893715e02a4

# https://www.sciencedirect.com/science/article/abs/pii/S0893608018300054