3-10-2021

Marijuana Intoxication Detection Using Convolutional Neural Network

Raj Gadhiya
University of Windsor

MARIJUANA INTOXICATION DETECTION USING CONVOLUTIONAL NEURAL NETWORK

By

Raj Gadhiya

A Thesis

Submitted to the Faculty of Graduate Studies
through the School of Computer Science in
Partial Fulfilment of the Requirements for
the Degree of Master of Science at
the University of Windsor

Windsor, Ontario, Canada

2021

MARIJUANA INTOXICATION DETECTION
USING CONVOLUTIONAL NEURAL NETWORK

By

Raj Gadhiya

APPROVED BY:

D. Xiao

Department of Physics

J. Lu

School of Computer Science

D. Wu, **Advisor**

School of Computer Science

January 25, 2021

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Machine learning is a broad study of computer science, widely used for data analysis and algorithms that has the ability to learn and improve by experience through training. Supervised learning, Unsupervised learning, Dimensionality Reduction, Deep Learning, etc are the methods offered by Machine learning. These techniques are applied in fields like medical, automotive finance, and many more. In this thesis, Convolutional neural network (CNN) which is a part of deep learning techniques is applied to identify if a person is under influence of Marijuana or sober, using facial feature changes like redness in eyes, watery eyes, and drowsiness caused after smoking Marijuana. CNN is a state-of-the-art method in tasks like image classification and pattern recognition. CNN's ability to learn from training the model using image dataset is a suitable method to be used in the problem of identifying a person's sobriety based on facial features. The proposed methodology is divided into three components. Which are dataset creation, face detection to extract input image from real-time video, and finally, tuning and training CNN model for making a prediction. The purpose of this thesis is to develop a CNN model that may be helpful if implemented in vehicles in the future to reduce impaired driving incidents. Impaired driving is a major criminal cause of vehicle accidents in Canada. Impaired driving is a serious problem that puts the lives of pedestrians on the road and drivers involved in impaired driving themselves in danger. This thesis presents how Machine Learning can be applied to predict driver's sobriety that may be helpful in reducing impaired driving incidents in the future by implementing in vehicles.

DEDICATION

To my family, friends, and everyone that supported me.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor, Dr. Dan Wu for providing me with the necessary guidance and support using his experience and expertise in the research field to complete my thesis. Without his suggestions, innovative ideas, and valuable comments this thesis would not be possible.

I would like to offer sincere gratitude toward my thesis committee members, Dr. Jianguo Lu for being my internal program reader and Dr. Dan Xiao for being my external program reader. Their valuable remarks helped me to improve my research.

I would also like to thank my all friends and family members who showed their support throughout my studies and thesis. I thank my parents for providing me with financial support to complete my studies successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

API	Application Programming Interface
CA	Condensation Algorithm
CNN	Convolutional Neural Network
Conv layer	Convolutional Layer
Conv net	Convolutional Network
EM	Expectation Maximization Algorithm
FP	False Positive
FPR	False Positive Rate
GMM	Gaussian Mixture Model
IDE	Integrated Development Environment
LSTM	Long Short-term Memory
MLP	Multilayer Perceptron
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TP	True Positive
TPR	True Positive Rate
TPR	True Positive Rate

CHAPTER 1
INTRODUCTION

1.1 Overview

We see a lot of incidents of impaired driving in our daily life which put not only the driver's life but pedestrians' lives in danger too. Vehicle crashes involving alcohol or drugs are the leading criminal cause of death in Canada. Every day, on average, 4 Canadians are killed and 175 are injured because of impaired driving related crashes [20]. To help prevent vehicle related crashes, most modern vehicles are equipped with safety features like drowsiness detection, lane assist, brake assist, and countless other safety measurements that make driving safer by keeping the driver focused behind the wheel. Moreover, we will review research papers in the next chapter on approaches to detect drunkenness, drowsiness and driver distraction detection using Machine Learning techniques in an effort of reducing vehicle crash incidents. However, there are no research papers available on detecting Marijuana intoxication. Therefore, we are proposing this approach to detect Marijuana intoxication using Convolutional Neural Network. Implementing this approach in vehicles may be able to help in preventing a person from operating a vehicle while being intoxicated.

The use of Marijuana is legal in many states and has become legal not very long ago in Canada as well. People may be careless and drive under influence of Marijuana since it is available easily now. It is dangerous to drive after smoking Marijuana because it makes your response very slow and makes you lose concentration. Fast response and focus while being behind the wheel are the most important factors to consider for safer driving.

1.2 Statistical analysis of driving incidents

Every day, on average, up to 4 Canadians are killed and many more are injured in alcohol and/or drug-related motor vehicle crashes on public roads involving at one "principal highway vehicle" (i.e., passenger cars, vans, trucks, and motorcycle) [19]. Every day, 29 people in the United States die in motor vehicle crashes that involve an alcohol-impaired driver. This is one death every 50 minutes [20].

In 2014, approximately 2,297 people died because of road crashes, and out of those, around 1,273 (55.4%) of deaths were because of driving under influence of alcohol and/or drugs [19]. 299 (13%) people died in crashes occurred involving individuals who were positive for alcohol, 618 (26.9%) of deaths in crashes occurred involving individuals who were positive for drugs and 356 deaths, or 15.5%, occurred in crashes involving individuals who were positive for both alcohol and drugs [19].

In 2015, police reported 72,039 impaired driving incidents [21]. That represents a rate of 201 incidents per 100,000 population. Yukon and Saskatchewan had the highest reported impaired driving cases while Ontario, Quebec, and Manitoba had the lowest reported cases.

- In 2016, 10,497 people died in alcohol-impaired driving crashes, accounting for 28% of all traffic-related deaths in the United States [22].
- Out of 1,233 traffic deaths among children ages 0 to 14 years in 2016, 214(17%) involved an alcohol-impaired driver [22].
- In 2016, more than 1 million drivers were arrested for driving under the influence of alcohol or narcotics [22].
- Drugs other than alcohol (legal and illegal) are involved in about 16% of motor vehicle crashes [22].
- Marijuana use is increasing and 13% of night-time, weekend drivers have marijuana in their system [22].
- Marijuana users were about 25% more likely to be involved in a crash than drivers with no evidence of marijuana use, however other factors–such as age and gender–may account for the increased crash risk among marijuana users [22].

Many vehicle manufacturers are conducting research and applying machine learning and artificial intelligence in their vehicles for improving the safety. One of the most popular examples would be Tesla Motors. It is an innovative car manufacturer known for its electric cars and its autopilot technology. Their camera networks analyse raw images to perform semantic segmentation, object detection, and monocular depth estimation. Their birds-eye-view networks take videos of road layout, static infrastructure, and 3D objects [33]. Their neural network learns from information captured from camera networks.

Therefore, taking inspiration from current methods we are going to use the machine learning technique, Convolutional Neural Networks (CNN) for our approach. CNN is a state-of-the-art technique for image classification tasks. It can classify images as good as humans if not better.

The use of CNNs is becoming more and more popular in fields like medical, finance, automobiles, social media, etc. CNNs are widely used in computer vision technics. Though they are not widely explored for detecting marijuana intoxication. We will go through a literature review to explore some papers representing approaches using deep learning technics to deal with drunk person classification problems. We are using similar experiments to train a CNN model to detect a person who is intoxicated using marijuana.

There are methods available to detect intoxication with 100% accuracy such as Breathalyzer for drunkenness detection or THC Breath Analyser. A company called Cannabix Technologies [49] is working with the Yost Research Group at the university of Florida to develop this breath analyser. But this device may be more suitable for traffic enforcement authorities and a user may not be in full conscious when intoxicated to use it to check if he or she is sober or not before driving. Therefore, something like video surveillance would be much more helpful that can automatically detect driver's sobriety from facial features.

1.3 Thesis motivation

Even with countless safety features in modern vehicles, there are a high number of accidents resulting in serious injuries and deaths. If we explore in the past when vehicles were discovered, there were many flows and lack of safety features or barely any features at all. With time, safety became the number one priority, and a tremendous amount of safety improvements can be seen in modern vehicles. We will also see research papers in the next chapter Literature Review, that demonstrates Machine Learning techniques that can be used to make vehicles safer by detecting impaired driving. A high rate of accidents shown in following statistical analysis of driving incidents, adding to impaired driving detection is a useful idea to consider.

1.4 Thesis objective

The main objective behind this thesis is to build and train a Convolutional Neural Network to predict marijuana intoxicated faces. In the future, it may be helpful in implementing in vehicles so that it can warn or stop a person from operating the vehicle when he or she is intoxicated. For making this possible we will go through research papers published for detecting a drunk person or other similar papers like drowsiness detection using neural networks to learn how a neural network can be trained to deal with the problem statement. In addition to that, we will need to create a dataset as well which contains pictures of faces when sober and when under influence of marijuana because there are no known datasets available currently on this topic. In the later chapter, we will

be comparing our results with other research papers and some baseline classification algorithms trained on the same dataset to evaluate the performance of our model.

1.5 Introduction to CNN

Convolutional Neural Network is a class of deep learning neural networks. CNN represents a huge breakthrough in image recognition. They are most commonly used to analyse visual imagery and are frequently working behind the scenes in image classification. From Facebook tagging to autonomous vehicles, CNNs are present for machine learning related tasks. Image classification is the process of taking an input and outputting a class or a probability that the input belongs to.

Let us take a simple example from [34] to understand neural networks. Assume that we want to detect a swan from a picture. For that, we need to consider features of a swan like oval-shaped white blob (body), round, elongated oval with orange protuberance, long white rectangular shape (neck) as can be seen in Figure-1 below.

Figure 1: Certain Features of a Swan [34]

Now, let us say if the detector algorithm for determining these features was to be created by a human, the detector would be either too general or too over-engineered. That would make the detector either too simple or hard to generalize. We can get an idea about what difficulties we may face by seeing the following picture in Figure-2 that shows various other possible scenarios of a swan that wood makes it complicated to detect using a simple detector.

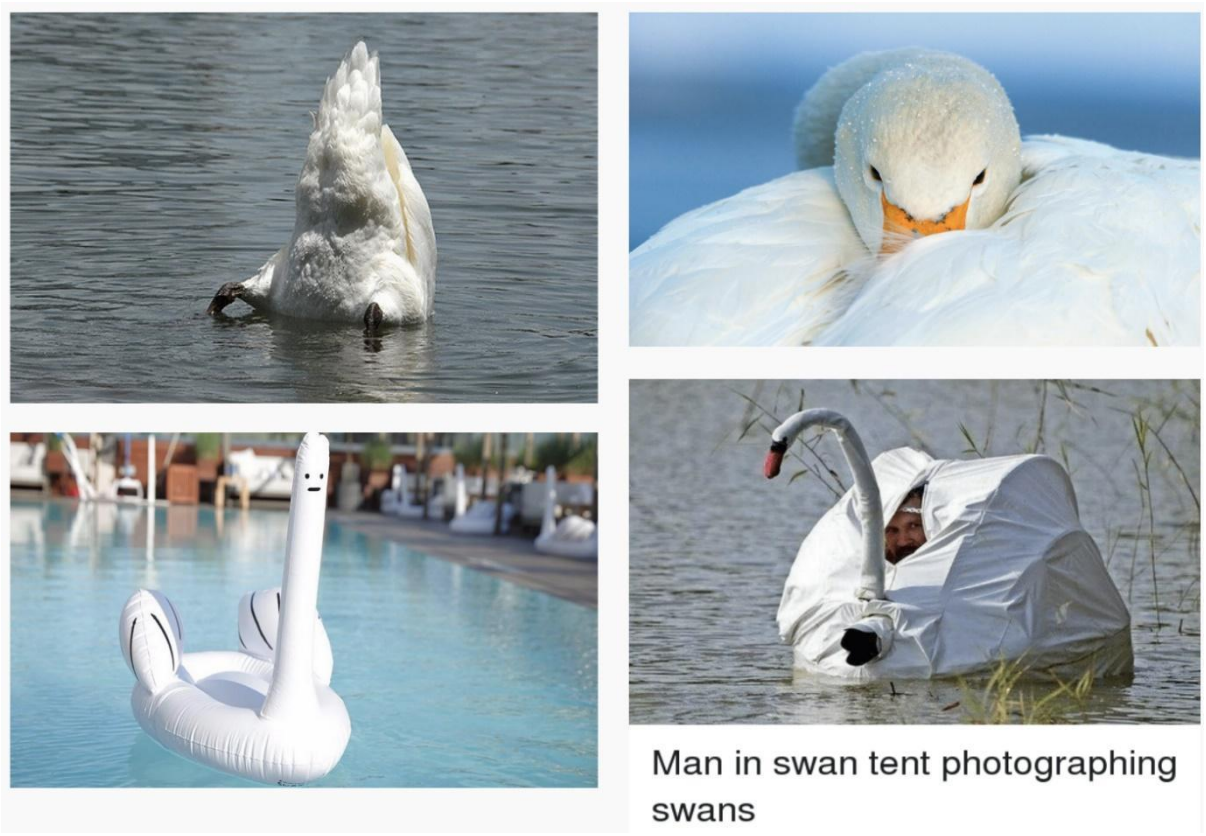

Figure 2: Possible Scenarios of a Swan [34]

What if we could learn features to detect? We need to use representation learning (or Feature Learning). Representation Learning is a technique that allows a system to automatically find relevant features [34]. Techniques for such tasks fall under two categories, Unsupervised and Supervised Learning. CNN is a supervised learning technique. Neural networks have the capability to learn from experience like neurons in human brains hence called neural networks. CNN detects features by relating nearby pixels of an input image by computing feature matrix using something called filter or kernel. A more detailed CNN's operation is explained in Chapter 3. First, let us see some applications to understand in what kind of tasks Convolutional Neural Networks can be used.

1.6 Applications of CNN

Convolutional Neural Networks are popular in a vast variety of fields. Machine Learning and Deep Learning techniques are often found behind the applications where we need to compute a large amount of complex data and tasks like decision making. Neural Networks are designed to work like a human brain and can learn over experience. Some of the popular but not limited applications are listed below.

- Facial Recognition
- Advertising
- Analysing Documents
- Analysing Weather Conditions
- Image Segmentation
- Medical
- Automotive

1.6.1 Facial Recognition

In October 2016 Yin Fan et al [1] presented a video-based emotion recognition system. The core module of this system is a hybrid network that combines recurrent neural network (RNN) and 3D convolutional networks (C3D) in a late-fusion fashion. Specifically, RNN takes appearance features extracted by convolutional neural network (CNN) over individual video frames as input and encodes motion.

Following image in Figure-3 is just a visual representation of features considered to detect a face. For example, algorithm will detect landmarks on a face like two eyes, nose, and mouth to identify a face.

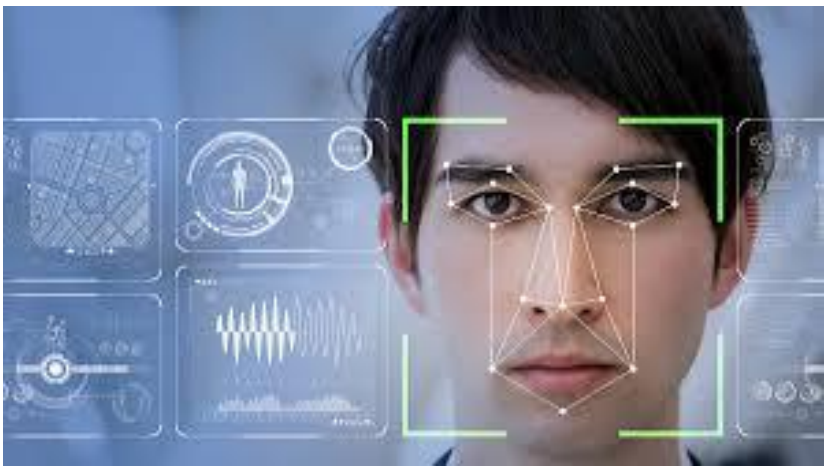

Figure 3: Facial Recognition [28]

1.6.2 Advertising

When browsing online websites, we encounter lots of ads. The ad publishers like Google and Yahoo sell ad zones on different web pages to advertisers who want to show their ads to users. And then Publishers get paid by advertisers every time the ad display leads to some desired action

such as clicking the ad and redirecting to their website [35]. Deep CTR Prediction in Display Advertising by Junxuan Chen et al. is a good example that shows use of neural network in predicting click through rate (CTR). Because, in most online advertising predicting the number of clicks is the core task of ads allocation.

1.6.3 Analysing Documents

One of the examples of analysing documents is, Fast CNN-based document layout analysis [1] proposed by Dario Augusto Borges Oliveira from IBM research Brazil. In this paper they proposed a technique in which a trained CNN model will detect the layout of a document from the image of a document. They pre-process a document input image and segment it into its blocks of content, use their vertical and horizontal projection to train a CNN model for multi-class classification considering text, image, and table classes.

1.6.4 Analysing Weather Conditions

In 2016, Ziqi Zhu et al [2] proposed a method to detect extreme weather recognition using convolutional neural networks. The weather is affected by many factors, features that can accurately represent various weather characteristics are difficult to be extracted. Therefore, in this paper, they applied convolutional neural networks to deal with this problem. The CNN model was trained using ILSVRC-2012 dataset.

1.6.5 Image Segmentation

Let us see an example of an autonomous vehicle where vehicle will need to understand each moving or stationary object on the road to take certain actions in order to avoid crashes. Neural networks can learn patterns from input image in order to predict object classes. Therefore, neural networks are used for segmentation. The main deep learning architecture used for image processing is a Convolutional Neural Network, or specific CNN frameworks like AlexNet, VGG, Inception, and ResNet [35].

Image segmentation is demonstrated in Figure-4 where objects from an image are separated by using bounding boxes.

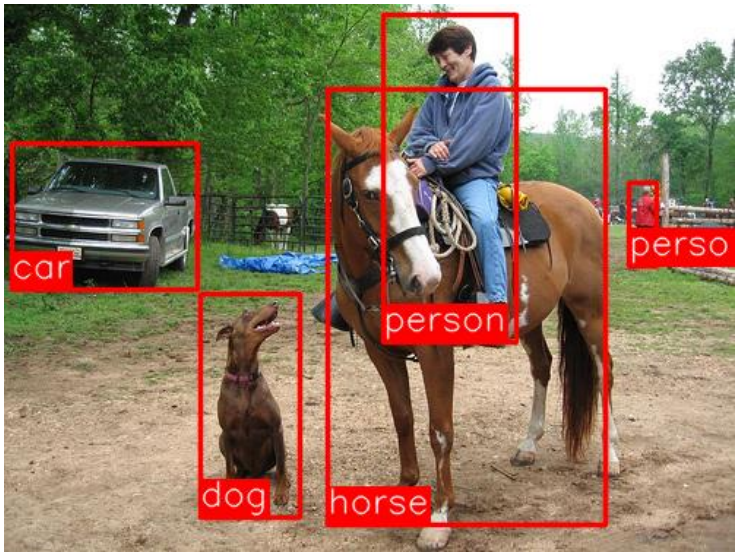

Figure 4: Image Segmentation [28]

1.6.6 Medical

CNN can also be used to diagnose various medical conditions. There are lots of research papers available that demonstrates use of CNN to predict cancer cells, tumour cells, determine health of retina and so many other examples of healthcare using machine learning.

Let us see an example that uses CNN to predict Diabetic retinopathy. In July 2016, the United States Food and Drug Administration approved the use of medical device using a form of artificial intelligence (CNN) to detect diabetic retinopathy in diabetic adults. The diagnosis of diabetic retinopathy (DR) through color fundus images requires experienced clinicians to identify the presence and significance of many small features which, along with a complex grading system, makes this a difficult and time-consuming task [3]. In this paper they propose a CNN approach to diagnose DR from digital fundus images and accurately classifying its severity.

1.6.7 Automotive

Autonomous vehicles are becoming a reality and lots of investments are being made to make it possible in future by countless automotive manufacturers including Tesla, BMW, Audi, BlackBerry QNX and many more. CNN can make a huge breakthrough in autonomous vehicles.

Figure 5: Autonomous Vehicles [29]

Image in Figure-5 is just a visual representation of how vehicles are sensing the objects on the road by sending signals.

In June 2018, Junekyo Jhung et al proposed an end-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles that improves driving performance compared to traditional CNN-based approaches. They used DAVE-2SKY and it was able to learn to inference steering wheel angles for the lateral control of self-driving vehicles [4].

CHAPTER 2

LITERATURE REVIEW

In this chapter, we will review other machine learning approaches to prevent impaired or distracted driving. Most of the modern cars have features to stop driver from being distracted. Use of machine learning is new trend and being adapted by many vehicle's manufacturers. There are many research papers available that uses CNN for various purposes like drunk detection, drowsiness detection and driver distraction detection from which we will discuss some of the approaches in brief. Although, there is not enough research done that uses CNN to detect if a person has smoked marijuana by using facial features. We will be comparing our thesis approach with some papers discussed below like drunkenness detection and drowsiness detection which shares somewhat similar features when person is high on marijuana.

2.1 Driver Behaviour Detection using CNN

In July 2020, proposed by Mohammad Shahverdy et al, a driver behaviour detection and classification using deep convolutional neural networks [5]. They proposed a novel yet efficient approach to monitor driver behaviour to reduce vehicle accidents by addressing possible privacy violation by previous methods which are rely on computer vision techniques. Instead of using driver's face, they are using signals generated from vehicle movement patterns. They are classifying the driving behaviour into five types, normal, aggressive, distracted, drowsy and drunk driving. In order to do that, they gathered vehicle data, including acceleration, gravity, RPM, speed, and throttle (the amount of accelerator pedal is pushed) [5]. The acceleration and gravity are gathered in three axes from a smartphone. RPM, speed, and throttle are measured by using the On-Board Diagnostic (OBDII) tool. They apply a time window on nine collected signals to distinguish the different driver behaviours. Then all windowed data are converted to the images by the recurrence plot technique [5]. They experimented with a different number of convolutional layers and filter sizes to train the model. Their paper mentions they were able to achieve an accuracy of 88% average.

For our experiments, we are also altering the number of parameters of our model to find suitable parameters. For different problems, a CNN model cam act differently depending on the input images. So, there is no concrete way of figuring out the exact configuration of a model for a

specific problem. That is why experimenting with different numbers of layers, kernel sizes, altering parameters, etc can be helpful in achieving the best performance of a classification model.

2.2 Drowsiness Detection

In March 2020, Maryam Hashemi et al proposed CNN-based Driver Drowsiness Detection [6]. According to reports from World Health Organization (WHO), vehicle related accidents are one of the top 10 causes that lead to death in the world [36]. Therefore, they are proposing this approach to detect drowsiness detection in drivers to reduce such accidents. This paper presents a novel approach for driver drowsiness detection. The proposed methodology uses Convolutional Neural Networks for monitoring diver's eye closure by focusing on two goals of real-time accuracy and speed. They mentioned that there is lack of dataset for such problem just like our approach, so they are also creating their own dataset of eye closure pictures. The first step in the process is to detect eye. In order to locate eyes from a frame, first need to estimate the headbox. To detect the headbox Viola and Jones algorithm [37] are used. Facial landmark detection used in this approach is a regression tree machine learning method and it was trained on the iBUG 300-W face landmark dataset [38]. When the landmark point of eye is reached, frame of eyes is cropped because face is asymmetry shape and not required or drowsiness detection, only eye is adequate to be considered. After cropping the eye, finally, the eye status is classified by computing distances from the most right and the most left point of the right and left eyes [6]. In the training process, eye is classified into two categories, open and closed. They introduced three potential neural networks. In which, one of them is a fully designed neural network (FD-NN), and others use transfer learning with VGG16 and VGG19 with extra designed layers (TL-VGG) [6]. For the dataset, they extended ZJU dataset to 4,157 images (2100 open and 2057 closed) [6]. Their TL-VGG19 method achieved accuracy of 95%, TL-VGG16 method achieved 95.45% accuracy and FD-NN method was able to achieve accuracy of 98.15%. to compute the results, they used a system with Core i7-6700K CPU@ 4.00GHz with 16GB RAM and NVIDIA GeForce GTX 1070Ti GPU.

Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques [17] proposed by Jabbar, R. at al. and published in 2018 is an attempt to reduce road crashes and related accidents using Multilayer Perceptron (MLP). They took inspiration to carry out this research because of the number of vehicle related accidents happening due to drowsiness from a study conducted by the AAA Foundation for Traffic Safety. The report showed that 23.5% of all automobile crashes recoded ion the United States in 2015 were because of drowsiness. For this research, they are using National Tsing Hua University (NTHU) Driver Drowsiness Detection

Dataset [19]. This dataset consists of images of 22 subjects of various ethnicities and these images of 22 subjects are divided into testing and training dataset. This dataset was created by recording various driving scenarios which simulates yawing, slow blink rate, conscious laughter, and dizzy dozing. They are using 18 subjects for training the model and 4 subjects for testing. First, videos from NTHU database us extracted followed by extracting images from video frames. Videos in this dataset are captured with 30 frames per second. They are extracting every frame as images. Next step is to detect facial landmark coordinates from the image. This task is done by using Dlib library's 68-point facial landmark detector. In our approach of detecting person intoxicated by marijuana, we are using 5-point facial landmark detector which is faster than 68-point facial landmark detector. After using the landmark detector to extract coordinates, the extracted coordinates are then used to be given as input to Multiplayer Perceptron based Classifier with three hidden layers to predict if person is feeling drowsiness or not. The first step in the process is, extracting videos from NTHU drowsy driver detection dataset then images are extracted from video frames. Once the input image is obtained, landmark coordinates from images are extracted. These coordinates of facial features are used as input to the algorithm based on multilayer perceptron classifier. Finally, the algorithm can detect of person's eye is closed or open in order to detect drowsiness. They were able to achieve overall accuracy of 80.92% which includes experiments with glasses, night without glasses, night with glasses, without glasses, and with sunglasses.

In June 2018, Macro Flores, Jose Armingol and Arturo Escalera proposed Real-time drowsiness detection system for an intelligent vehicle [18]. To help in reducing road accidents, in this paper a new advanced driver assistance system (ADAS) is presented. In their study, they observed that an estimate of between 10% and 20% of traffic accidents are cause by drowsiness [46]. That motivated them to carry out research to design and to build systems able to monitor drivers and their level of attention while driving. They considered visual features on face when feeling drowsy like yawn frequency, head movement, facial expression, eye-blinking frequency, and eye-gaze movement [18]. Like our approach, the first step in their method is face detection. To detect face, their system uses VJ object detector which is useful machine learning technique for object detection. Three important aspects. Integral image, AdaBoost technique and cascade classifier [47] makes the VJ object detector an efficient object detector. Drawback of using VJ object detector is that it does not work appropriately when the face is not in front of the camera. To overcome this problem, they implemented Condensation Algorithm (CA) proposed by Isard and blake for tracking active contours using a stochastic approach. To solve the problem of face alignment, they

also implemented a tracker using a neural network in conjunction with CA that can predict the exact position of face in front of camera. Once the face is located, both eyes are tracked using expectation maximization algorithm (EM) and marked using ellipses. Final step in the approach is to determine the state of eyes, closed or open using Support Vector Machine (SVM). A training set consists of open eyes and closed eyes was built in order to train SVM. After several training experiments, they decided to use RBF kernel and they were able to reach the accuracy of 94%.

2.3 Drunkenness Detection

DIF- dataset of intoxicated faces fir drunk person identification proposed by Vineed Mehta et al [7] in May 2018 is a very similar approach to our proposed method for identifying persons intoxicated by use of marijuana. In this paper they are proposing an automatic bimodal non-invasive intoxication detection. They are using Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) for training to compute the video and audio, respectively. A new dataset is proposed as well to train the model called DIF (Dataset of Perceived Intoxicated Faces for Drunk Person Identification). To generate the dataset, they browsed through social media and videos of drunk people available online. They observed that there are channels on social media websites like YouTube, where people record themselves in intoxicated states. They found videos of interviews, reviews, reaction videos, video blogs and short films. In order to search these videos from websites, they used keywords like drunk reactions', 'drunk review', 'drunk challenge' etc. For our approach we are using the same method to search videos where people record themselves while smoking marijuana to build our dataset. Their dataset was built from total 78 videos in the sober category and 91 videos in drunk category. The sober category consists of 78 subjects and drunk category consists of 88 subjects. To process these videos, they are using pyannote-video library [19] and face detection is performed from video frames. After extracting frames, face alignment is done using OpenFace toolkit [1]. Next step is to use extracted images for prediction. Two pre-trained models on face dataset are used. VGG-face [9] and AlexNet based network trained on RAF-DB [7] dataset. They are using AlexNet model to extract features is because it is trained on the RAF-DB which is a database of facial expressions. Therefore, the feature extracted from the pre-trained network will be more aware of the facial deformations generated due to facial expressions. During the experiments, they observed that VGG-face base representation gave better results than RAF-DB based. They are also training a Lon Short Term Memory LSTM network using the audio features. They were able to achieve the accuracy of 78.12% for CNN model and 78.06% accuracy

for audio model. By combining both methods they were able to predict the drunkenness with the accuracy of 88.39%.

One more paper is Drunk selfie: Intoxication Detection from Smartphone Facial Images proposed by Colin Willoughby et al [10] that detects person's drunkenness based on photo taken from a smartphone. they observed statistics of impaired driving incidents from a website. According to the report from website, over 10,000 people died because of drunk driving. taking inspiration from the statistics they carried out this research. They also developed a prototype android app and implemented this method. This app can take pictures of person's face and classify them into sober or drunk using their machine learning intoxication classifier. To train the model they used "3 glass later wine project" dataset created by Brazilian photographer Marco Alberti [17]. Photographer created this dataset in curiosity to see how people act after each glass of wine. So, he captured photos of different individuals when they were completely sober, after one glass, after two glass and after three glass of wine. For this experiment, 52 individuals were involved including males and females both. Dataset of 2,332 images was built using 212 photographed combining with internet photos and using augmentation. For augmentation, the imgaug python library was used. Image rotation, brightening, burning, changing perspective, changing contrast, and adding tint were the transformation operations used for expanding the image dataset. The first step is image pre-processing. In this part faces from the image is being detected using Histogram of Oriented Gradients (HOG) method. They used Dlib implementation for facial landmark detection. In our approach, we are using the same method to detect face from an image. Instead of using 68-point facial landmark detection, we are using faster 5-point facial landmark detection. After the face is detected, landmarks such as eyes, mouth and nose are located because they are likely to change in response to alcohol [12]. These landmarks are detected using the Facial Landmarking algorithm [36]. Once the landmarks are detected, they are used for aligning the face. Features from the images are now extracted like forehead redness because according to the research done by them, when alcohol is consumed, the face gets red especially cheeks and forehead. They compared various classifier types including the Linear Support Vector Classifier, Polynomial SVC, Random Forest, and Decision Tree Classifier. Out of these, Random Forest performed the best. They were able to achieve the accuracy of 81%.

Drunk person identification using thermal infrared images proposed by G. Koukiou, G. Panagopoulos, and V. Anastassopoulos [17] published in July 2019 uses a similar approach of identifying drunk person using thermal images instead of using direct images from a normal camera. The infrared images used in this work were acquired by means of the Thermo Vision

Micron/A10 Model infrared camera of FLIR company [17]. The main reason behind using thermal images instead of normal images is that the physiological properties of the face depend mainly on its temperature, which in turn is strictly related to the distribution of the blood vessel network on it. [37].

To obtain the data, they acquired pictures when persons had not consumed any alcohol. Then they gave 330ml of beer to each person three times for every 30 min and obtained images after 25 minutes of each beer. In order to monitor the temperature change from the consumption of alcohol, 20 different points of the face were selected of each person. From these points, a 20-dimensional feature vector was created which corresponds to a point in the 20-dimensional space. Thus, formatting four clusters when sober and after three beers in an interval of 30 minutes. In order to examine the separability of the four clusters in the feature space, they compare the scatter of each cluster from the centre of the cluster. The distance between the first and last cluster was calculated and compared with the summation of the clusters maximum scatter. The result showed that the distance between the first and the fourth cluster was much larger than the summation of the clusters maximum scatter. Therefore, two clusters were totally separable.

The conclusion is, they presented an approach to the problem of identifying drunk people using thermal images using the method based on the representation of a specific person into feature space and observed that as the person consumes beers, cluster of features vector moves further. For improving the performance and accuracy, they will be conducting more experiments in future.

One more research paper that uses images other than direct images from normal camera is Face Recognition and Drunk Classification Using Infrared Face images proposed by G. Hermosilla, J. Luis, G. Farias, E. Vera, F. Pizarro, and M. Machuca [16] was published in Jan 2018 takes on the approach to identify person's sobriety by examining changes on face after consuming alcohol using infrared images instead of thermal images used in the previous paper. A process called thermoregulation shows that a biological organism modifies its internal temperature within certain limits and is commanded by hypothalamus [16]. In their research studies, they found that some studies have shown that the thermoregulatory system can be altered depending on mood or the consumption of certain food [38]. Alcohol causes motor disturbances and disturbances in the psychic systems, resulting in abnormal behaviour on a biological level, such as dilation of blood vessels [39-40] and increased blood pressure. In case of human face, the temperature increases around nose, forehead, and eyes in response to alcohol consumption.

According to their research, there are large number of applications in machine learning, such as face recognition, facial expressions, and personal identification, computer systems, etc. but the problem of drunk classification is not widely studied. The problem of marijuana intoxication classification is not widely explored either. Therefore, experimenting and researching for this problem may provide useful insights and results that can be used for future studies. Only significant work they found was done by the University of Patras, Greece [41-42]. The aim of their study is to demonstrate that it is possible to classify drunk and sober person using the intensities of pixels located in different part of a face like forehead, nose, and mouth. Using this, a space separable features can be generated [16]. Only problem with this study was they had only small number of data created with limited subjects (8 individuals) available therefore it was not possible to ensure generalisation of the classifier. In order to improve accuracy and better generalisation they are using larger dataset called Pontificia Universidad Católica de Valparaíso-Drunk Thermal Face database (PUCV-DTF). In the construction of this database, 46 individuals, 40 men and 6 women were selected with the average age of 24 years. This test excluded people who consumed alcohol daily in order to to get better visual changes on face area. Procedure was carried out after all subjects were rested for 30 minutes to stabilise the metabolism to the temperature conditions of the test laboratory. Then they were given 355ml can of beer and waited for 30 minutes and then again given the same amount of beer and it was repeated until four beers were consumed. For thermal imaging, FLIR TAU 2 [43] with a resolution of 640x480 pixels, a frame rate of 30 frames per second, thermal sensitivity of 50mK, and a spectrum range between 7.5 and 13.5 μm was used [16]. Built dataset contains 250 images per subject and 50 images per subset (sober, 1 beer, 2 beers, 3 beers and 4 beers). For feature extraction, dimensionality reduction methods were used on local regions of a thermal face image. They observed in [44], they were using features extracted from a grid of 20 points which did not have biological details on the location of the feature points. That is why this paper is using a different grid of 22 points which was inspired by [45]. In this method, points are selected at positions where there are capillaries and veins that cross the face. After feature extraction is completed, a feature vector of 22 dimensions is generated. Hence there are 50 feature vectors for each of the five classes. Since there are 22 dimensions, separating them by a hyperplane is too complex that is why they are using Fisher linear discriminant analysis to reduce the dimensionality. Once the dimensionality reduction is completed using FLD, the Gaussian mixture model (GMM) was used to perform the classification. At the end of experiments, they were able to achieve the accuracy of 87% when a person has consumed at least one beer.

CHAPTER 3

METHODOLOGY

3.1 Flow of work

In this chapter we will see methods and techniques used in the implementation of this thesis which consists of dataset creation, face detection using 5-point landmark face detection and training the convolutional neural network. Furthermore, experiments and results will be discussed in Chapter 4 including comparison of results with other popular machine learning classifiers like SVM, Decision Trees and Random Forest.

Our focus of this thesis is to build a CNN model that can predict images of sober faces and marijuana intoxicated faces. If we recall from Chapter 1, there are plethora of applications and use of CNN in variety of fields. For such different tasks, different configurations and parameters of a classification algorithm are required depending on the classification task. There is no certain way to tell which configuration will work the best for a given classification problem. So, to find the suitable parameters we need to carry out number of experiments with different parameters. Proposed methodology can be divided into three main parts, dataset creation, face detection and training the neural network (CNN model) as shown in Figure-6 below.

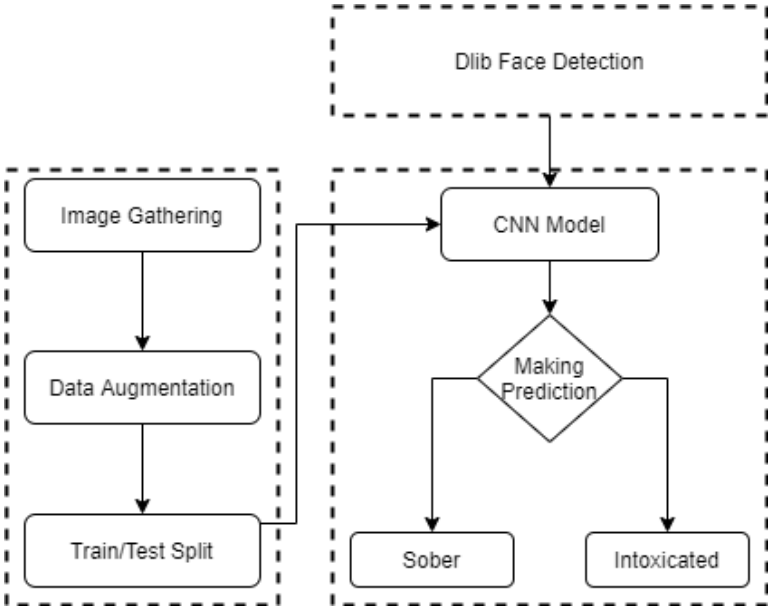

Figure 6: Proposed Methodology

Dataset: Process of dataset creation and dataset augmentation.

Face Detection: Implementation of Dlib 5-point landmark face detection.

CNN Model: Architecture of CNN model, training and testing of the model.

3.1.1 Image Gathering

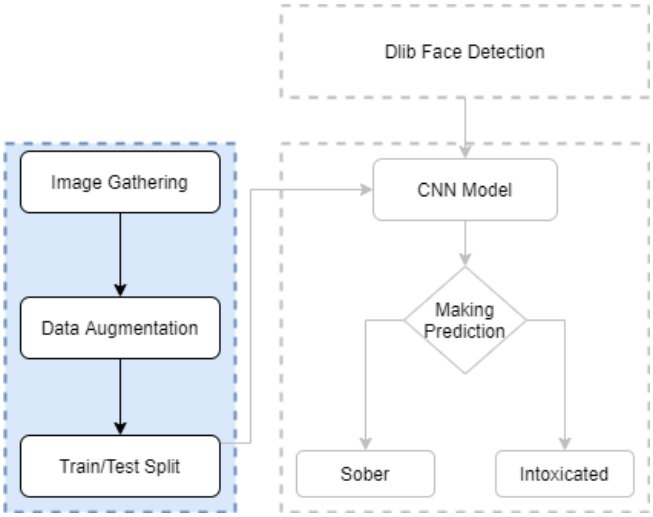

Figure 7: Dataset Creation

Creating a dataset was the most challenging and time-consuming task. Figure-7 is highlighting the part we will be covering in this section. Most research papers we saw in literature review Chapter-2, created their datasets in lab by photographing real people's faces after experimenting with different amount of alcohol consumed by participants. But couple of research papers created their dataset by collecting images from internet which seems less resource consuming since it does not require a bunch of participants, alcoholic drinks (wine or beer) needed for consumption by participants or camera. Because of limited resources, we created our dataset by using similar method of acquiring images from social media platform like YouTube and google image search. Dataset needed to be created because there was no known dataset available which consists of pictures of faces intoxicated by marijuana. Approach to create dataset was inspired by method Yadav, D. P., and Dhall [8] used in their drunk person detection paper.

Pictures of intoxicated people can be gathered from social media platforms like YouTube and Google Image Search. There are tons of youtubers who posts videos of them smoking marijuana in front of camera. We can take screenshots and crop images from such videos when they are intoxicated. From some videos we can extract picture of the same individual when they are sober and when intoxicated but it is not always possible to get "before" and "after" picture of the same

person. So, the dataset will consist of random images of faces when they are intoxicated and when they are sober. Figure-8 below shows a sample of what dataset looks like displaying intoxicated faces on top and sober faces at the bottom.

Figure 8: Dataset Sample

We went through google searches to find people's face by using key words like marijuana, smoking pot, marijuana red eye etc. From YouTube, we watched several videos of people smoking marijuana in front of camera and captured multiple screenshots when they were sober and when they were intoxicated. A detailed demonstration of how images were gathered is explained later in Chapter-4. It is very difficult to gather thousands of images so using image augmentation was in order.

3.1.2 Data Augmentation

It may not be always possible to have access to a huge amount of data. In that case Data augmentation can be very useful. Limited number of images in a dataset can be a major obstacle in training a Convolutional Neural Network because training a classification model requires thousands of images. To deal with this problem, we can use a common method used in machine learning called Data Augmentation. Data augmentation is very useful and widely used technique to expand dataset whenever available dataset is not large enough.

We can increase the size of dataset just by using simple image transformation techniques like rotation, random crop, flipping, altering image contrast etc. Figure-9 displayed below is an example of output after performing transformation techniques. We can see the original image at the left side of the figure and six augmented images.

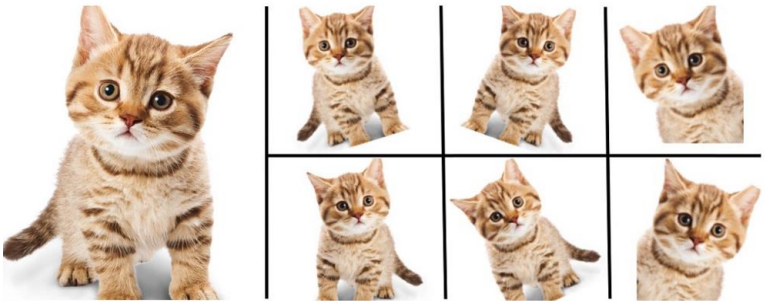

Figure 9: Data Augmentation Sample [30]

Following Image in Figure-10 shows a sample of augmentation performed on one of the images from our dataset. Where left image is the original image and six images on right side are generated after performing image transformations.

Figure 10: Augmented Dataset Sample

3.1.3 Face Detection

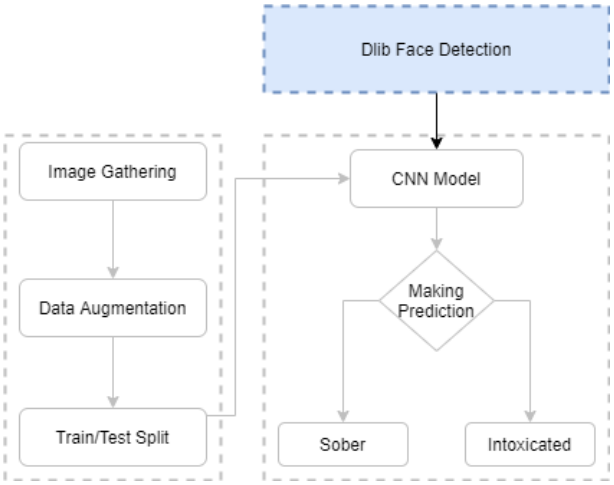

Figure 11: Face Detection

Figure-11 is highlighting the part we will discuss in this section. In order to classify an image, first we need to extract image from video and locate the face from the image so it can be used as input of our CNN model. For face detection, we are using Dlib 5-point facial landmark detector. It is faster and effective compared to 68-point facial landmark detector by 10% in speed and significantly smaller (only 9.2MB compared to 68-point facial landmark detector's 99.7MB). This algorithm is suitable for tasks like face alignment. Figure-12 shows an example of face being detected from a picture using 5-point facial landmark detector.

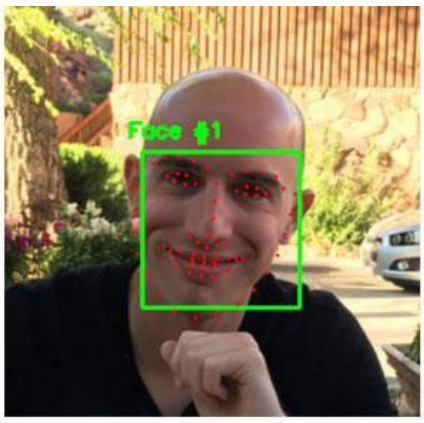

Figure 12: Face Recognition in action [16]

Davis King [12] released a public repository which contains trained model on 3 million faces. This model is a ResNet network. It is essentially a version of the ResNet-34 network from the paper Deep Residual Learning for Image Recognition by He, Zhang, Ren, and Sun [13] with a few layers

removed and number of filters per layer reduced by half. Adrian Rosebrock explains how to use and implement faster facial landmark detection in a tutorial [14].

Following image in Figure-13 is a screenshot of implementation of Dlib and video used is one of the examples of YouTube video used to extract images for dataset. Face is detected from the video and marked by using a green rectangle. Once the face is detected, we can simply crop the image inside the green rectangle and use the image as input of our classification model.

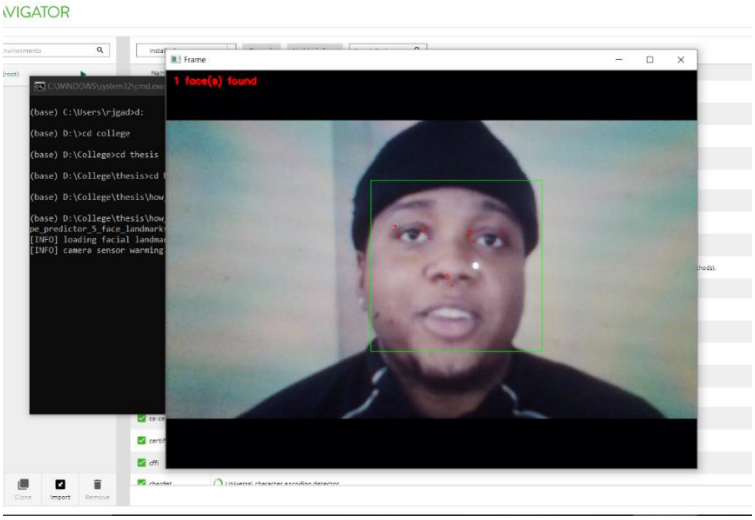

Figure 13: Face detection implementation

In DrunkSelfie paper proposed by Willoughby et al [11] uses Dlib with 68-point detector instead of 5-point detector [16] used in this thesis. 5-point detector works by locating 5 points, two eyes, nose and mouth on face in order to locate a face. Similarly, in the paper Real-time driver drowsiness detection [18] uses the same approach as Willoughby's paper of using 68-point detector for face detection.

3.1.4 CNN Architecture

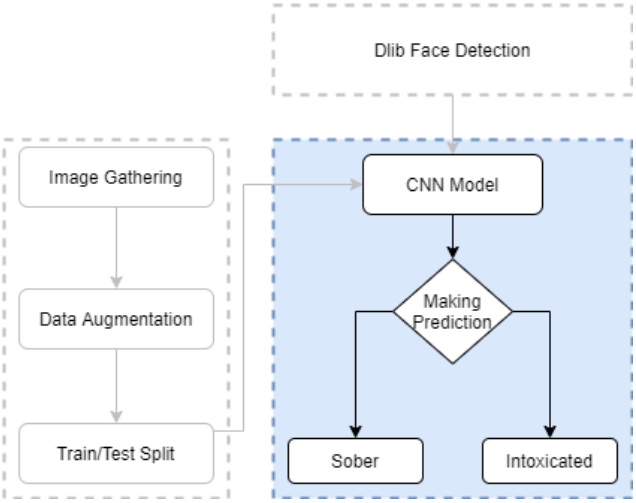

Figure 14: CNN Model

Figure-14 is highlighting the part we will be covering in this section. The Convolutional Neural Network (CNN) is a class of deep learning neural networks. CNNs represent a huge breakthrough in image recognition and classification tasks. They are commonly used to analyse visual imagery and are frequently working behind the scenes in image classification. Image classification is the process of taking an image as input and predict the class or a probability of the image as output. A typical Conv network consists of following layers.

- Convolutional layers
- Activation layers
- Pooling layers
- Fully connected layer

Number of layers and types of layers in a CNN depends on the purpose of the model. For example, a complex model built to classify images of cat and dog may require different configuration lie size of filter, number of layers, etc, than a model for classifying species of flowers. Figure-15 below is showing the architecture of the model we are using for Marijuana intoxication detection problem. It shows the main components like number of layers, size of input, number of nodes, and size of kernel/filter. We will go through each component in detail in the following section.

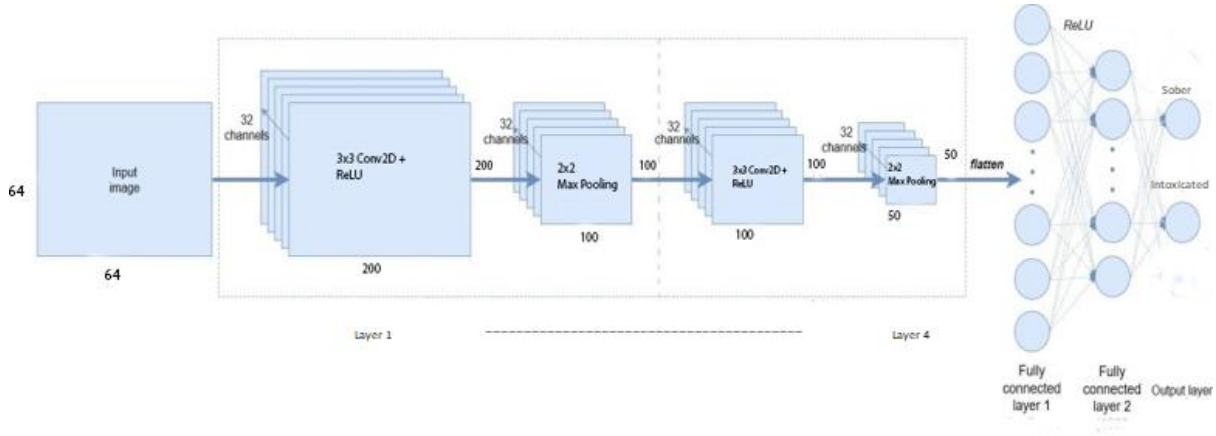

Figure 15: CNN Architecture

Implementation of the model was done using Keras API. Keras is a high-level neural networks API written in Python language and capable of running on top of TensorFlow, CNTK, or Theano [23]. In our implementation, it is running on top of TensorFlow. Keras allows for easy and fast prototyping which makes it easier to build a classification model. It is packed with all the necessary libraries required to build a CNN model.

Our classification problem is binary (possible output is sober or intoxicated) so we are using sequential model. Sequential model allows us to build a model layer by layer. As shown in the architecture in Figure-15, our model is made with a linear stack of layers which will process the input through layers successively and finally predict the class of the input image through output layer. First layer is Conv2D with 32 filters of size 3x3 followed by three more Conv2D layers. In order to activate these layers, we need to introduce activation function in each Conv layer. *ReLU*, which stands for Rectifier Linear Unit is used as activation function in our model. Immediately after each convolution layer there us a pooling layer present. Pooling layer is used to perform pooling operation on the feature map calculated by Conv layer. Pooling is performed to reduce the size of image to prepare it for next convolution layer. A matrix is formed after the pooling operation is done in the last Conv layer. This matrix needs to be converted into continuous vector. Which can be done using Flattening operation. As the name suggests it flattens the input i.e., 2x2 matrix output from previous layer will be converted to single dimensional vector. After the flattening, comes the fully connected layer. This layer connects all the nodes from previous layer and the output layer. This layer will contain the number of nodes always between the number of input nodes and output nodes. Finally, last one is the output layer which will contain only one node because we are dealing with binary classification problem, there is only one possible output, sober or intoxicated. Let us see each layer in detail in the following section.

3.1.5 Input Image

Figure-16 below is representing an RBG image which has been separated by its three colour channels, Red, Green, and Blue. There are different types of images available in different colour spaces such as HSV, CYMK, Greyscale etc. It would be computationally intensive if images reach higher dimensions i.e., 8K (7680x4320). The role of a ConvNet is to reduce the images into a form which is easier to process, without losing features which are important for better classification [24].

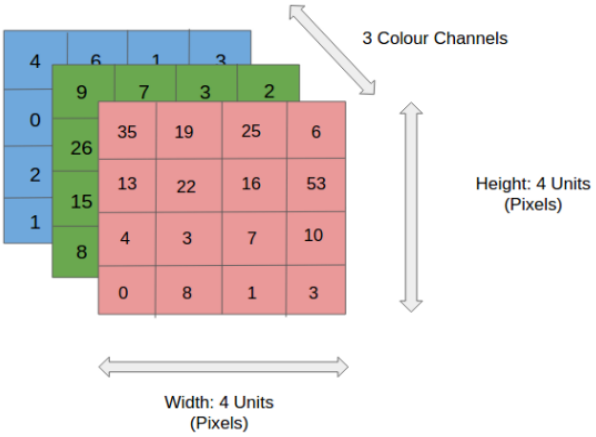

Figure 16: Input Image [24]

3.1.6 Convolutional Layer and Kernel

Convolutional layer is the first layer in a CNN model. The role of a Conv layer is to convolute through whole image pixel by pixel to generate feature map that will be used by next layers. Let us take an example of a simple 2D image of 5x5 size shown in Figure-17 below along with the kernel/filter of size 3x3. The element involved in carrying out the convolutional operation in a convolutional Layer is called the Kernel or Filter.

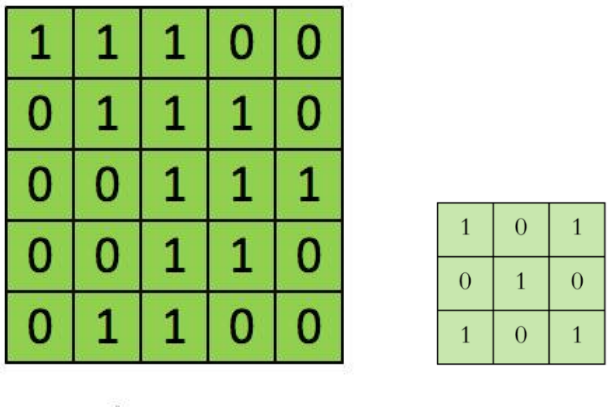

Image Dimensions = 5 (Height) x 5 (Width) x 1 (Number of channels), Kernel size = 3 (Height) x 3 (Width)

Figure 17: Image Matrix and Filter Matrix [24]

In the above demonstration, the green section resembles our 5x5x1 input image. From this grid, kernel or filter is used to carry out the convolution operation. In the image above, we have taken 3x3 size filter which is represented in the right side of the image. This filter will convolve through all the image pixels from left to right until every pixel is covered. In the end we will get convolved matrix. Kernel convolution is not only used in CNNs but is also a key element of many other Computer Vision algorithms. Convolved matrix is calculated according to the following formula.

$$G[m, n] = (f \times h)[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - k \quad (3.1)$$

Where,
f = Input image
h = kernel / filter
m and **n** = indexes of rows and columns of the result matrix

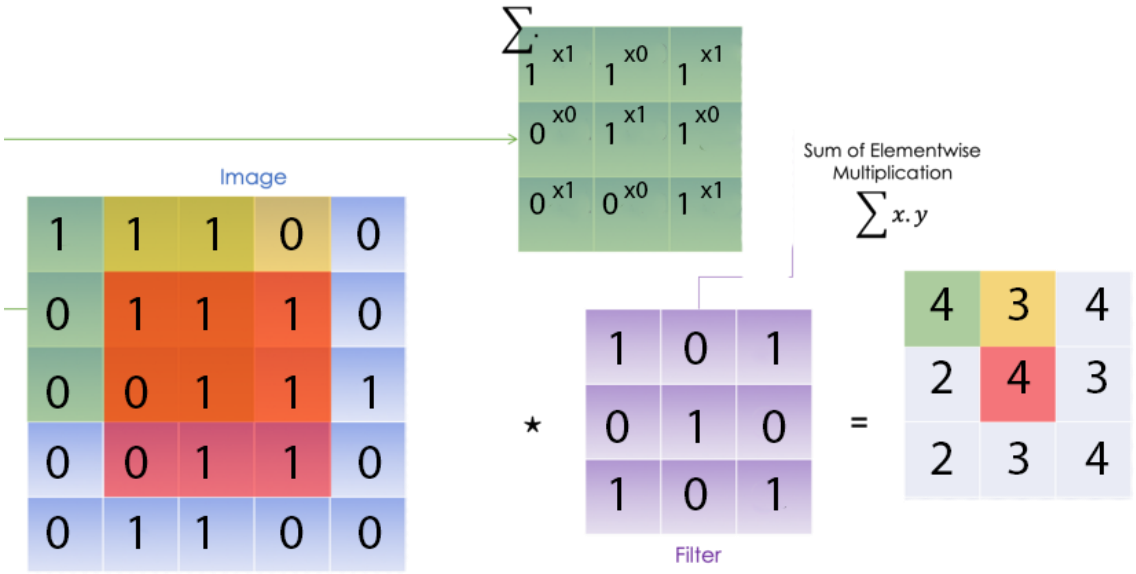

Figure** 18**: Output

Figure-18 above is a visual representation of operation of a convolutional layer. As shown in the figure, we get an output matrix that after using filter or kernel on the input image. This matrix represents learned features that will be fed to pooling layer to further reduce the dimensionality.

Following images in Figure-19 shows few steps of formula (3.1) in action.

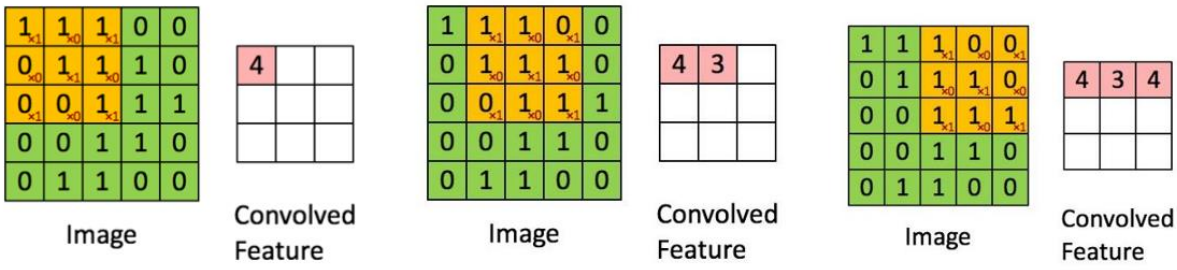

Figure 19: Convolutional layer in action [24]

As shown three steps above, kernel will keep sliding through whole image and we will get feature map matrix at the end of the process. Final output of first convolution is displayed in the following image in Figure-20.

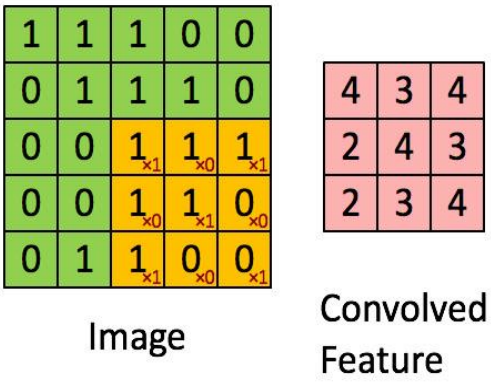

Figure 20: Output Feature Map [24]

For the simplicity, we used an image with one channel. If there are multiple channels like RGB image, then we will need to use the filter with same number of channels. So, for an RGB image which has 3 channels we need to use a filter with 3 channels.

In case of multiple channels (e.g., RGB), the kernel has the same depth as that of the input image. Matrix Multiplication is performed between *Kn* and in stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output [24].

3.1.6 Pooling Layer

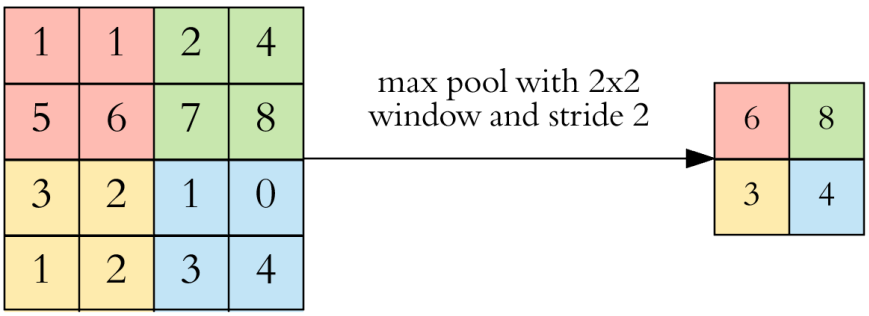

Figure 21: Pooling Operation [24]

Pooling layer works like convolutional layer, it is responsible for reducing the spatial size of the convolved feature matrix. Pooling layer down samples each activation map. Let us consider a pooling layer of size 2x2. It will slide through feature map just like a kernel would slide through input image. In the above Figure-21, we can see it is selecting maximum numbers from each 2x2 window. There are two types of Pooling. Max Pooling and Average Pooling as shown in the figure below.

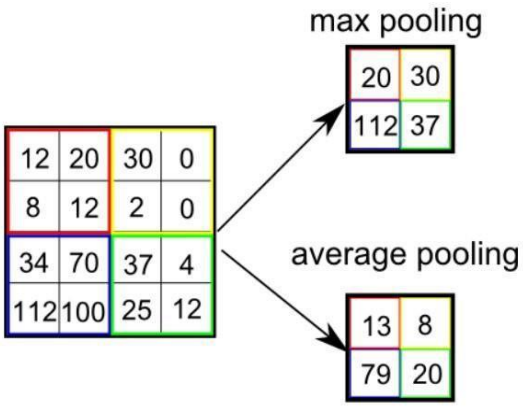

Figure 22: Max Pooling and Average Pooling [24]

Figure-22 is showing two available pooing methods. As the name implies, max pooling selects the maximum number from the feature map while average pooling will store the average value of the feature map. We are using max pooling in our model because it works a little faster them average pooling without affecting the accuracy very much.

3.1.7 ReLU

ReLU is the abbreviation of Rectified Linear Unit. This layer applies the non-saturating activation function. This is the most commonly used activation function in deep learning models. Mathematically it can be represented using following function.

$$f(x) = max\ (0, x) \tag{3.2}$$

ReLU effectively removes negative values from an activation map by setting them to zero and returns the value back if *x* is positive. It can be represented graphically as shown in following image in Figure-23.

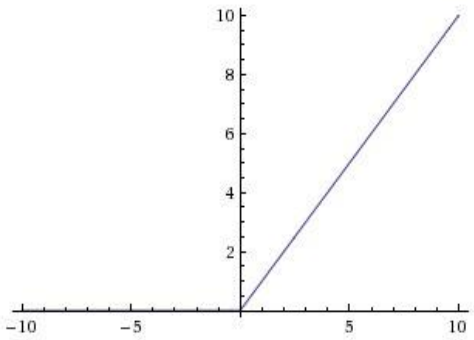

Figure 23: ReLU [24]

It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. There are other functions as well that can be used to increase nonlinearity, But *ReLU* is often preferred to other function because it trains the neural network several times faster without a significant penalty to generalization accuracy. Other activation functions are *Sigmoid, Softmax, Softplus, Softsign, Tanh, Selu, Elu*, and *Exponential.*

3.1.8 Fully Connected Layer

This layer is located between output layer and all the nodes from previous layer and can have number of nodes in the range of number of nodes available in the previous layer and output layer. Figure-24 is showing an example of fully connected layer connecting nodes from input and output.

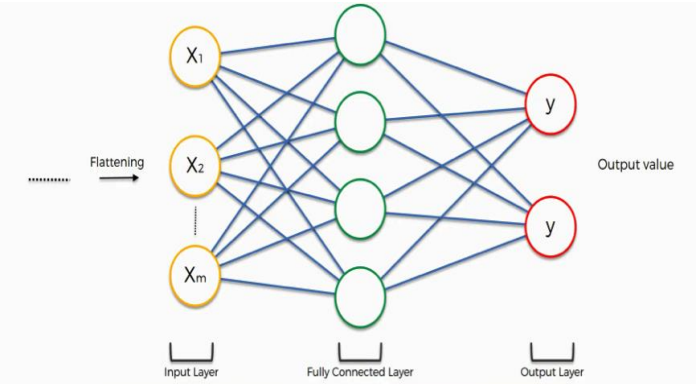

Figure 24: Fully Connected Layer [31]

After all the convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers. Neurons in a fully connected layer have connections to all activations in the previous layer. Fully connected layer is to learn non-linear combinations of the high-level features as represented by the output of the convolutional layer. The fully connected layer is learning a possibly non-linear function in that space [24].

3.1.9 Dropout Layer

To reduce overfitting, we are using **dropout layer and image augmentation.** During training, some number of layer outputs are randomly ignored or dropped out. In effect, each update to a layer during training is performed with a different "view" of the configured layer. We will try to use different setting for this layer like **dropout (0.2)** and **dropout (0.5)** means dropping connection probability is 20% or 50% respectively.

In the paper, Realtime drowsiness detection by Jabbar, R. et al also mentions using dropout to prevent overfitting. Introducing dropout layer using Keras is very easy, we can specify the frequency to drop the connection.

3.2 Hyper Parameters

Hyperparameters are the variables which determines classification model's structure (e.g., Number of Hidden Units) and variables which determine how the network is trained (e.g., Learning). Hyperparameters are set before training the model [26]. Following are some common parameters to tune for training the model.

Kernel/Filter:
A kernel or filter is a matrix which is used for carrying out convolution operation in Conv layer. We discussed kernel in the chapter 3 in detailed. In our case we are using a 3x3 kernel for training our model. Smaller filters will collect as much local information as possible, while bigger filters will collect more global, high-level and representative information [27].

Stride:
It is the number of pixels you wish to skip while traversing the input horizontally and vertically during convolution after each element-wise multiplication of the input weights with those in the filter. It is used to decrease the input image size considerably as after the convolution operation the size shrinks to ceil((n+f-1)/s) where 'n' is input dimensions 'f' is filter size and 's' is stride length. ceil rounds off the decimal to the closet higher integer [27].

Number of Channels:
It is the equal to the number of colour channels for the input but in later stages is equal to the number of filters we use for the convolution operation. The more the number of channels, more the number of filters used, more are the features learnt, and more is the chances to over-fit and vice-versa [27].

Pooling-layer Parameters:

Pooling layer has similar parameters to convolutional layer. Just like the kernel, it will convolute through feature map generated after convolutional operation. In our model 2x2 size max pooling operation is used to reduce the dimensionality of the image.

3.3 Cross Validation

We are using cross fold validation in order to reduce overfitting. K-fold cross validation is one of the popular methods used to evaluate machine learning models when the available data is limited just like in our case. Drawback to this technique is that it can be very resource intensive and time consuming if higher number of folds are used. In k-fold cross validation, dataset is divided into "k" equal parts and model is trained by switching these splits for training and testing. For example, in our case, we are using 3-fold cross validation. As shown in Figure-25 and Figure-26 below, dataset is divided into three equal splits. (In our experiments we divided data into three equal folds of 625 images in each fold)

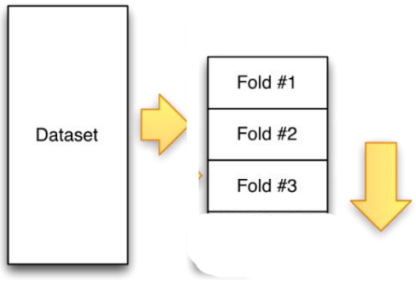

Figure** 25**: 3-fold cross validation

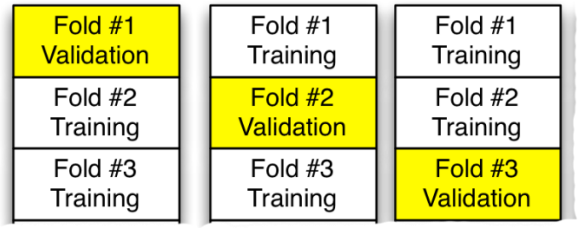

Figure** 26**: 3-fold split

When training the model one of the folds (let us say fold k) is kept for validation and other remaining folds are used for training the model. once the training is completed another fold (k-1) is kept for validation and all other folds are used for training and likewise it is repeated until every fold is used. At the end of 3-fold cross validation, we will get three different models. We can compare the results of each fold to verify if we get similar results or not.

Advantage of using k-fold cross validation is that performance of model can be evaluated correctly because each time model will see differ data when being trained. Hence, the model will not be biased towards dataset splits.

CHAPTER 4

RESULT AND EXPERIMENTS

In this chapter, we will discuss about experiments and methods carried out to build and train our CNN model discussed in Chapter 3 and finally evaluate the performance of the model by comparing the results with other approaches like drunk detection and drowsiness detection that uses neural networks to detects a person's sobriety. Since there are no known papers available on detecting intoxication caused by use of marijuana yet, we are comparing our results with research papers like drunk and drowsiness detection. These papers show use of neural network to classify images based on changes in facial features after consuming alcohol or when a person is feeling drowsy which resembles with facial features change after smoking marijuana (drowsy, watery and red eyes). Since there are no research papers available for reference, detecting intoxication caused by marijuana is going to be an initial and primitive approach.

Experiments in this chapter consist of method used for gathering images to create the dataset used to train our model and finding suitable model parameters for optimal performance and accuracy. Gathering images and building our own dataset were the most challenging and extensively time-consuming tasks. Since there are no public dataset available to use yet, we had to create our own dataset. We saw earlier in the literature review in Chapter 2, that most of the research papers on drunk and drowsiness detection developed their own dataset by photographing group of people [8,19,17] and couple of them built the dataset from gathering images from social media platforms [9,12]. The dataset used in this thesis was built using same technique of collecting images from social media which is discussed in the following section in detail.

4.1 Dataset creation

Dataset is the most important part for training a neural network with optimal performance. To create our dataset, two types of images were required which are sober and intoxicated. Such images can be acquired from google image searches and social media platforms like YouTube where people may upload their videos when smoking marijuana for entertainment purposes such as what happens if you smoke given amount of marijuana at once, to see the reaction of a person when they smoke marijuana for the first time, etc. Especially the images of sober faces can be found easily from the internet like google image search, just by simply searching with keywords like

faces, human face, face closeup, etc. However, collecting images that can show facial feature changes like watery red eyes after smoking marijuana can be tricky and time consuming. Collecting images from social media is inspired by research paper [7]. If we search in the YouTube with queries like smoking marijuana, smoking weed, getting high, etc, we can find many videos of people smoking marijuana in front of camera just for fun. From these videos, we can easily take screenshots and crop images when person is sober and after the same person has smoked marijuana. Using this method, we can gather approximately 15 to 20 images belonging to both classes, depending on the video. In total, 150 images (75 sober and 75 intoxicated) were gathered from videos.

Let us take an example, from this link [36], we can keep pausing the video at appropriate time and take a screenshot. The following image in Figure-27 shows a screenshot of one of the videos where a person can be seen experimenting with what happens before and after smoking marijuana. In the starting of the video (top image), person can be seen sober and can be seen intoxicated short after (bottom image).

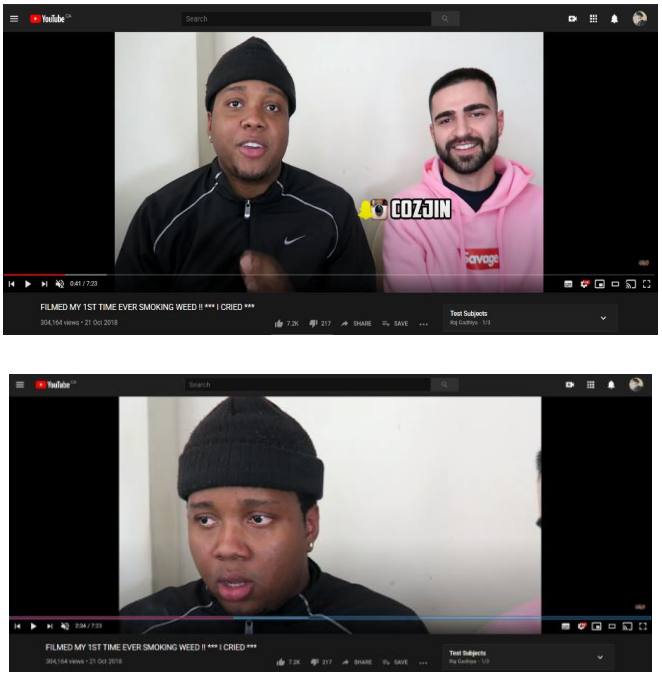

Figure 27: YouTube Video Sample [36]

In this case we were able to get sober state and intoxicated state of the same person, but it is not always possible. So to increase the size of dataset, random sober and intoxicated images were added as well by searching queries like “after smoking marijuana”, “smoking pot”, “marijuana red eyes”, etc in Google search and gathered around 450 images (225 sober and 225 intoxicated) resulting in a total 600 images (300 sober and 300 intoxicated). This dataset is very small and not

sufficient to train a neural network well enough. To overcome this problem, a common dataset expansion technique called data augmentation was used.

4.2 Dataset Augmentation

Data augmentation is the most popular method used when available dataset is too small. To train a neural network, depending on the type of classification problem usually large amount of data is required to train the model well. It may not be always possible to have enough data such as in our case. Hence, data augmentation can be very useful technique to expand the dataset. Most common and easy way to increase dataset size is by using image transformation techniques. We used some common image transformation techniques like rotation, flipping and random cropping on our 600 original images. At the end of the process, we were able to generate 2,750 images (1,375 sober and 1,375 intoxicated). Out of these 2,750 images, we kept 300 images (150 sober and 150 intoxicated) separate so we can test our model against unseen images after training is done and used remaining 2,450 images (1,225 sober and 1,225 intoxicated) for training and validation of the model. Generally, a CNN model would require approximately 80% training data and 20% validation data to be trained well. Therefore, we are using 1,825 images for training and 625 images for validation. Table-1 shown below summarizes breakdown of dataset in detail.

Dataset summary	
Number of original sober images	300
Number of original intoxicated images	300
Number of original images	**600**
Number of augmented sober images	1,375
Number of augmented intoxicated images	1,375
Number of total augmented images	**2,750**
Number of images for training	1,825
Number of images for validation	625
Number of total training images	**2,450**
Number of reserved images for testing	**300**

Table 1: Dataset Summary

4.3 Confusion Matrix

Confusion matrix can be used to measure the effectiveness of a classification model or a classifier. It is used to calculate performance measurements for machine learning classification tasks. Building a confusion matrix is very easy. Confusion matrix can be built by using correctly predicted values and incorrectly predicted values by a classification model, placed inside a matrix as shown in Figure-28 below. To build a confusion matrix, first, we need to calculate True Positive, False Positive, True Negative and False Negative values which are explained below.

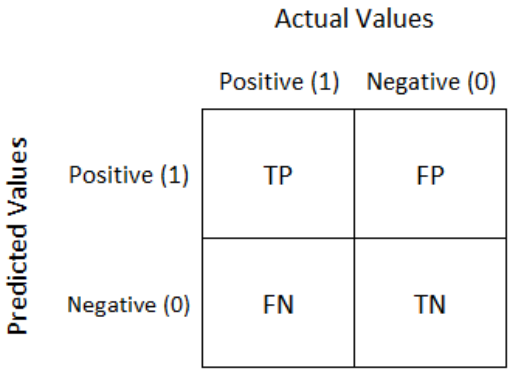

Figure 28: Confusion Matrix

True Positive (TP) is the number of correctly predicted value of positive class i.e., correctly predicting intoxicated class.

False Positive (FP) is the number of incorrectly predicted value of positive class**.** For example, if the model predicts the input image as intoxicated, but the input image belongs to sober class. It is also known as "Type I error".

True Negative (TN) is the number of correctly predicted value of negative class i.e., correctly predicting sober class.

False Negative (FN) is the number of incorrectly predicted value of negative class. For example, if the model predicts the input image as sober, but the input image belongs to intoxicated class. It is also known as "Type II error".

After obtaining the values of TP, FP, FP and FN through experiment, we can calculate performance measures such as Accuracy, Error Rate, False Positive Rate, False Negative Rate, True Positive Rate, True Negative Rate, Precision, Recall and F-Score using following formulas.

Where: $t_p = TP, t_n = TN, f_p = FP, f_n = FN$

Accuracy:

Accuracy is how often the model predicted the correct value. When we are building a classification model, we want it to be most accurate in predicting our inputs. After building the model, to make the decision of whether the model is good enough or not, accuracy alone is typically not enough. For example, if we are building a classification model that can classify terrorists and passengers boarding a plane. Let us say we have 100 people boarding the plane (95 passengers and 5 terrorists) and the model predicts 90 passengers correctly (TN), 5 passengers incorrectly (FP), all 5 terrorists incorrectly (FN) and 0 terrorist correctly (TP). We will still get the accuracy of 90%. But in reality, the model is horrible and let all 5 terrorists board the plane! Accuracy can be calculated using following formula (4.1) and by multiplying it with 100, we get the accuracy in percentage.

$$\boldsymbol{Accuracy} = \frac{t_p + t_n}{Total} \qquad \textbf{(4.1)}$$

Error Rate:

It is calculated as the number of all incorrect predictions divided by the total number of images available. The best error rate is 0.0, whereas the worst error rate is 1.0. We want our model to make as less mistakes as possible and want it to be closest to 0. We saw that only accuracy alone is not enough to decide if a model is good, only error rate alone is not enough either. If we calculate the error rate using passenger and terrorist classification example from above, we will get an error rate of 0.05 which looks too good, but we know that the model was horrible and let all 5 terrorists board the plane. Error rate can be calculated using following formula (4.2).

$$\boldsymbol{Error\ rate} = \frac{f_p + f_n}{Total} \qquad \textbf{(4.2)}$$

False Positive Rate:

How often a model incorrectly classifies class A (Positive class) as class B (Negative class). For example, if our model predicts that an input image belongs to intoxicated class when the input image actually belongs to sober class. Using False Positive Rate, we can measure how often our model is making mistake predicting intoxicated class. Lower the false positive rate, lesser the model is predicting the intoxicated class wrong. A good False Positive Rate should be as close as possible to 0. It can be calculated using following formula (4.3).

$$\boldsymbol{False\ positive\ rate} = \frac{f_p}{f_p + t_n} \quad \textbf{(4.3)}$$

False Negative Rate:

Just like False Positive Rate, False Negative Rate shows that how often a model predicts an input belongs to class B (Negative class) whereas it actually belongs to class A (Positive class). For example, if our model predicts that an input image belongs to sober class when the input image actually belongs to intoxicated class. Using false negative rate, we can measure how often our model is making mistakes predicting sober class. It can be calculated using following formula (4.4).

$$\boldsymbol{False\ negative\ rate} = \frac{f_n}{f_n + t_p} \quad \textbf{(4.4)}$$

True Positive Rate:

How often a model can predict correctly the images belonging to positive class. For example, in our case, how often our model can predict the correct class for images belonging to intoxicated class. This measure can help us to understand the performance of our model for intoxicated class. It is also known as “Sensitivity” or “Recall”.

Now, let us recall our example of passenger and terrorist classification boarding a plane. We show that only accuracy and error rate alone were not much helpful deciding if the model was good or not. For that we can use something called precision and recall. For our passenger and terrorist classification problem, our Recall value will be 0. That means our model cannot tell at all if a passenger boarding the plane is a terrorist or not which completely defeats the purpose of the model despite having 95% accuracy!

Recall can be calculated by using following formula (4.5).

$$\boldsymbol{Recall} = \frac{t_p}{t_p + f_n} \quad \textbf{(4.5)}$$

True Negative Rate:

How often a model can predict correctly the images belonging to negative class. For example, in our case, how often our model can predict the correct class for images belonging to sober class. This measure can help us to understand the performance of our model for sober class. It is also known as "Specificity". It can be calculated by using following formula (4.6).

$$\boldsymbol{Specificity} = \frac{t_n}{t_n + f_p} \quad \textbf{(4.6)}$$

Precision:

Precision can be defined as the ability of a classification model to identify the proportion of data belonging to positive class correctly (how many are actually positive from all the positive predicted). i.e., our model's ability to identify the proportion of intoxicated images correctly.

Let us use the same example of passenger and terrorist classification and we will get precision of 0 because we do not have any true positives (model could not identify any terrorists).

It can be calculated using following formula (4.7).

$$\boldsymbol{Precision} = \frac{t_p}{t_p + f_p} \quad \textbf{(4.7)}$$

Precision and Recall are frequently used as performance measures in machine learning models. In some situations, we can accept higher precision and lower recall or higher recall and lower precision depending on the classification problem.

For example, in case of passenger and terrorist classification, we really want to stop all terrorists from boarding the plane (high recall). Unfortunately, we cannot increase both: increasing precision reduces recall, and vice versa and it is called precision/recall tradeoff.

Let us see another example from [51], if we are training a classifier that can filter safe videos for kids, we would prefer a classifier that may reject a few good videos (low recall) but keeps only safe videos (high precision).

However, in some cases, we want to find the optimal balance between precision and recall when good performance in predicting both the classes is required. For example, in our case, we want to our model to classify both intoxicated and sober classes correctly. To find the balance between precision and recall, we can use something called F-Score.

F-Score:

F-score is the harmonic mean of Precision and Recall. It is also known as "F1-score" or **"**F-measure". It can be calculated using precision and recall value by following formula (4.8). If we want to create a classifier that is optimal for both classes, we want the F-score to be maximized.

$$\boldsymbol{F_1} = \frac{2}{recall^{-1} + precision^{-1}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \qquad \textbf{(4.8)}$$

In the next section, Experiments, we will use these performance measures obtaining from corresponding confusion matrices to analyse results of experiments. Let us see the experiments performed to train our CNN model and then compare the results with other popular classification algorithms like SVM, Decision Trees and Random Forest trained as baseline classification models using the same dataset.

4.4 Experiments

In this section, experiments carried out to train our CNN model discussed in Chapter 3 are summarized. For every experiment, same dataset for training (80%) and validation (20%) were used with the image size of 64×64×3 (Height, Width, No. of channels). For training the model, 1,825 images (925 sober and 925 intoxicated) were used and for validation 625 images (312 sober and 313 intoxicated) were used. To check the performance of our model against unseen images, we kept 300 images (150 sober and 150 intoxicated) separate from training and validation to check how the final model will perform against unseen data.

Experiment 1

Experiment 1 was conducted using 3 convolutional layers with 32 kernels in each convolutional layer and the kernel or filter size of 3×3. These convolutional layers need to be activated in order to retrieve feature vector from the input image. To activate these convolutional layers, *Relu* activation function was used. There are different activation functions available to choose from i.e., *softmax, softplus, tanth, selu,* etc but the reason behind using *Relu* is that it works significantly better than most other activation functions and is often used as default activation function according to tutorial explained in [50].

After each convolutional layer, a Pooling layer of 2×2 size was used. We can choose from two different pooing functions, Max Pooling and Average Pooling. We used Max Pooling because it works faster with almost no penalty in performance. Max Pooling will return maximum value from the matrix while Average Pooling will calculate the average of all values. Pooling layer must be smaller (i.e., 2×2) than the convolutional layer's filter (i.e., 3×3), because we need to reduce the dimensionality of the input image. For example, if the input image is of size 5×5, the convolutional layer of 3×3 will give us an output of 3×3 matrix then the pooling operation done by 2×2 layer will further reduce the size of matrix to 2×2.

After the convolution and pooling operations are done, we need to create a fully connected layer to connect the set of nodes we get after flattening and set of nodes in the output layer. Fully connected layer is located between the last pooling layer and the output layer, it is also called hidden layer. In this experiment, the fully connected layer used 128 units or nodes. The number of nodes in this layer should be always between the number of output nodes and the number of input nodes we get after convolution and pooling layers. For the activation of the fully connected layer, *Relu* was used as well.

The Last layer is the output layer with only one node in it, because we are dealing with a binary classification problem, so the output is going to be either sober or intoxicated hence it will have only one node. *Sigmoid* function was used for activation of output layer because we need either 1 or 0 output (binary classification problem).

For experiment 1, *adam* optimizer was used. There are several other optimizers are available to use like *SGD, RMSprop, Adadelta, Adagrad, Adamax, Nadam* and *Ftrl*. To compute the loss, *binary_crossentropy* loss function was used which is suitable for binary classification problems. There were no dropout layers used in experiment 1. Table 6 shown below summarizes parameters used for experiment 1.

Parameters for experiment 1	
Input image size	64×64×3 (height, width, no. of channels)
Number of Convolutional layers	3
Kernel or filter size	3×3
Number of filters	32
Activation (Convolutional layers)	*Relu*
Number of Pooling layers	3
Pooling size	2×2
Pooling method	Max pooling
No. of nodes in Fully connected layer	128
Activation (Fully connected layer)	*Relu*
No. of nodes in Output layer	1
Activation (Output layer)	*Sigmoid*
Number of Dropout layers	0
Dropout rate	0.5
Optimizer	*rmsprop*
Loss function	*binary_crossentropy*
Number of Epochs	22

Table 2: Parameters for experiment 1

Graphs are often useful to understand statistics or data in visual form. In Machine Learning, we have something called learning curves which are basically plots that shows model fitting progress during the training. Learning curves are plotted using model accuracy and model loss throughout the training process. Learning curves can be useful to diagnose under-fitting, over-fitting or well-fitting of the model. **Accuracy** is calculated in the form of percentage and shows us how accurate our model's prediction is for each epoch. **Loss** is calculated by loss function, *binary_crossentropy*. Model loss is nothing, but prediction errors made by our model for each epoch. Both of these measures are plotted on a graph as shown in Figure 30. These plotted are representing the training

process of Experiment-1. Blue plot represents training accuracy and loss while orange plot represents validation accuracy and loss.

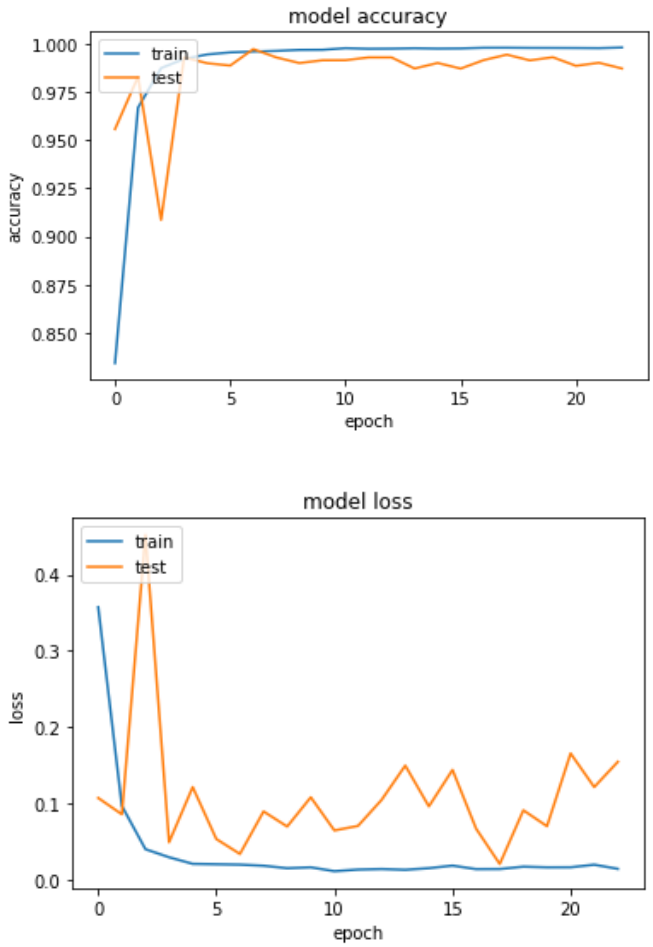

Figure 29: Accuracy and Loss for Exp1

In Figure-29, we can see that validation accuracy is constantly below training accuracy meaning the model is predicting training data with higher accuracy while predicting validation data with lower accuracy. Also, validation loss is going higher than training loss which is a sign of overfitting. So, we will try to alter some parameters to see if we get better results in next experiment.

After conducting Experiment-1 using parameters shown in Table-2, we get the following Confusion Matrix shown in Figure-31 and Table-3 below shows the values calculated from this Confusion Matrix.

n = 625	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 236	**FP** = 91	
Actual: Intoxicated	**FN** = 76	**TP** = 222	
	312	**313**	

Figure 30: Confusion Matrix for Exp1

Performance Measures for Experiment 1	
Accuracy	73%
Error Rate	0.26
False Positive Rate	0.27
False Negative Rate	0.25
Precision	0.70
Recall	0.74
F-Score	0.71

Table 3: Performance Measures for Exp1

By looking at the Table-3, we can see that False Positive Rate and False Negative rates are a little higher. While our Precision and Recall are lower. We can try to improve them in next experiment.

Experiment 2

We saw in the previous experiment that the model may be overfitting means it was learning more features than needed. So, let us try reducing the number of nodes from 128 to 64 and adding a dropout layer. Dropout layer can help the model to reduce overfitting by eliminating connections to random nodes. This time instead of *adam*, *RMSprop* optimizer was used but they both work the same for similar classification problems. All other parameters are same as the previous Experiment 1. Table-4 shown below, summarizes parameters used in Experiment-2.

Parameters for experiment 2	
Input image size	64×64×3 (height, width, no. of channels)
Number of Convolutional layers	3
Kernel or filter size	3×3
Number of filters	32
Activation (Convolutional layers)	Relu
Number of Pooling layers	3
Pooling size	2×2
Pooling method	Max pooling
No. of nodes in Fully connected layer	64
Activation (Fully connected layer)	Relu
No. of nodes in Output layer	1
Activation (Output layer)	Sigmoid
Number of Dropout layers	1
Dropout rate	0.5
Optimizer	Rmsprop
Loss function	binary_crossentropy
Number of Epochs	20

Table 4: Parameters for Experiment 2

By the look of the learning curves below in Figure-31, it can be seen that the model is fitting better than the previous experiments. Training and validation accuracy are coinciding with each other. That means the model is predicting both accuracy and validation data with similar performance. But it seems like validation accuracy is still below training accuracy in the epochs between 8 and 11 which may be a sign of overfitting. Also, the validation loss is getting higher than the training loss at the end of epochs.

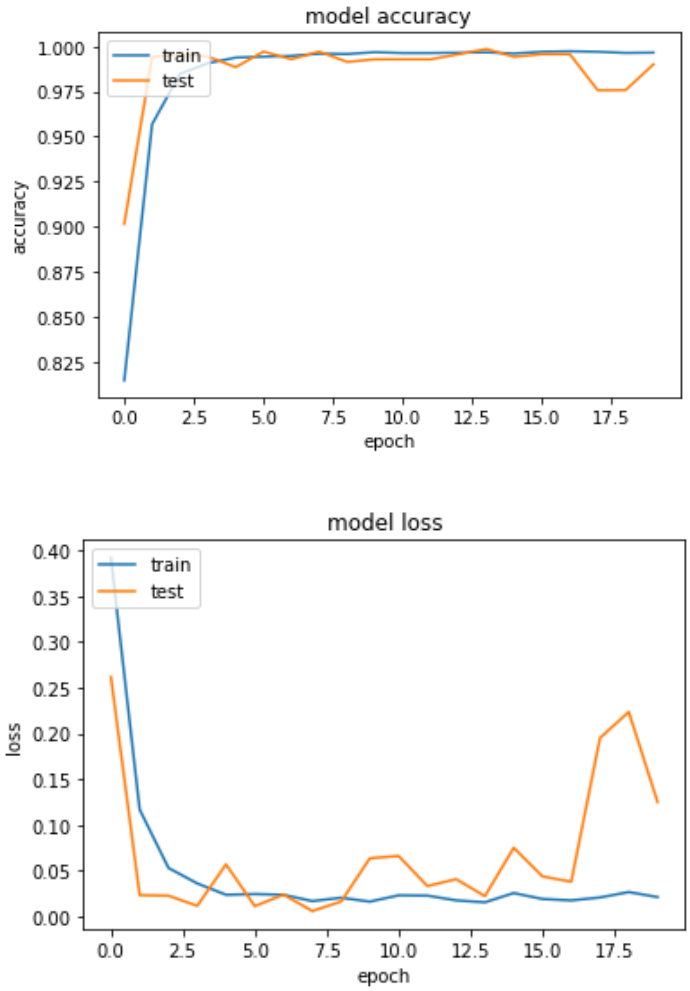

Figure 31: Model Accuracy and Model Loss for exp2

Figure-32 shows Confusion Matrix obtained from Experiment 2 and the Table-5 below shows the performance measures calculated from this Confusion Matrix.

n = 625	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 267	**FP** = 101	
Actual: Intoxicated	**FN** = 45	**TP** = 212	
	312	**313**	

Figure 32: Confusion Matrix for Exp2

Performance Measures for Experiment 2	
Accuracy	76%
Error Rate	0.23
False Positive Rate	0.27
False Negative Rate	0.17
Precision	0.67
Recall	0.82
F-Score	0.73

Table 5: Performance Measures for Exp2

We can see from the Table-5, Accuracy has improved slightly while False Positive Rate stayed exact same. On other hand, False Negative Rate dropped from 0.25 to 0.17 that means the model is making less mistakes identifying sober images. Precision is reduced and Recall has increased that means model is identifying some of the sober images as intoxicated. Let us try to improve results in Experiment-3.

Experiment 3

In this experiment, one more convolutional layer was added than the previous experiment. The number of epochs were increased as well. By increasing the number of convolutional layers we can give the model a chance to learn features better. All other parameters are same as Experiment 2. Table-6 below summarizes the parameters used for this experiment.

Parameters for experiment 3	
Input image size	64×64×3 (height, width, no. of channels)
Number of Convolutional layers	4
Kernel or filter size	3×3
Number if Filters in Convolutional layer	32
Activation (Convolutional layers)	Relu
Number of Pooling layers	4
Pooling size	2×2
Pooling method	Max pooling
Pooling size	2x2
No. of nodes in Fully connected layer	64
Activation (Fully connected layer)	Relu
No. of nodes in Output layer	1
Activation (Output layer)	Sigmoid
Number of Dropout layers	1
Dropout rate	0.5
Optimizer	Rmsprop
Loss function	binary_crossentropy
Number of Epochs	23

Table 6: Parameters for Experiment 3

We can see from the learning curves shown in Figure-33 below that the model is fitting better then previous experiments. We can see that the training accuracy and validation accuracy as well as training loss and validation loss are staying close to each other throughout the entire training process indicating modle is performing same on training data as well as on validation data. Since

everything looks good in Experiment 3, we still need to be sure. So we can use cross validation for this experiment.

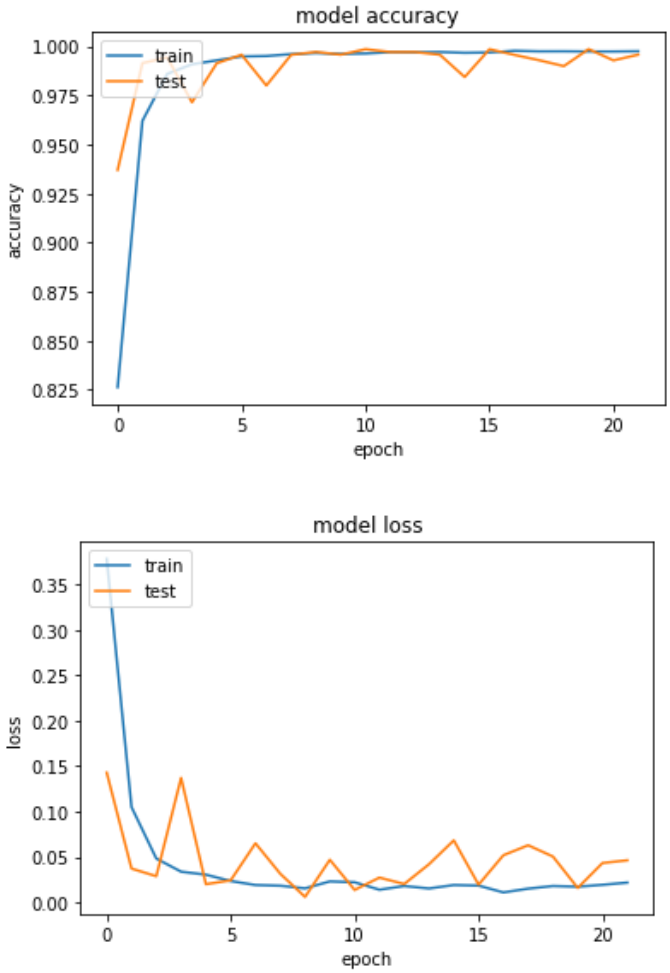

Figure 33: Accuracy and Loss for Exp3

Unlike Experiment-1 and 2, we will generate three different Confusion Matrices because we are using 3-fold validation and compute performance measures for each matrix to see if we get similar results in each fold to make sure the model is well fitted. Following Figures-34, 35 and 36 shows Confusion Matrices for all three folds. During the training of each fold, three different validation sets (swapping out new 625 images from total data for each fold) were used.

n = 625	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 247	**FP** = 77	
Actual: Intoxicated	**FN** = 65	**TP** = 262	
	312	**313**	

Figure 34: Confusion Matrix for Fold1

n = 625	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 268	**FP** = 50	
Actual: Intoxicated	**FN** = 44	**TP** = 263	
	312	**313**	

Figure 35: Confusion Matrix for Fold2

n = 625	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 265	**FP** = 54	
Actual: Intoxicated	**FN** = 47	**TP** = 259	
	312	**313**	

Figure 36: Confusion Matrix for Fold3

Performance Measures for Experiment 3			
	Fold 1	**Fold 2**	**Fold 3**
Accuracy	81%	84%	83%
Error Rate	0.22	0.15	0.16
False Positive Rate	0.23	0.18	0.16
False Negative Rate	0.19	0.16	0.15
Precision	0.77	0.80	0.82
Recall	0.80	0.83	0.84
F-Score	0.78	0.81	0.82

Table 7: Performance Measures for Exp3

As we can see from the Table-7 that we got almost same results for all three folds which indicates lower chance of overfitting. For comparison and discussion, we will use the average of all three folds. We achieved good accuracy of 83% and a lower error rate of 0.16. We also have low False Positive Rate as well as False Negative Rate that means model is not incorrectly classifying any of the classes more frequently. Higher Precision and Recall indicate are indication of good performance in predicting both classes as well as a higher F-Score shows good balance between Precision and Recall.

Now that we can consider Experiment-3's model a good model, we can finally test it against our reserved 300 images to see how it performs on unseen data. Confusion Matrix below in Figure-37 was calculated by making predictions on reserved images.

n = 300	**Predicted:** Sober	**Predicted:** Intoxicated	
Actual: Sober	**TN** = 126	**FP** = 29	
Actual: Intoxicated	**FN** = 24	**TP** = 121	
	150	**150**	

Figure 37: Confusion Matrix for Reserved Images

Performance Measures for Reserved Dataset	
Accuracy	82%
Error Rate	0.17
False Positive Rate	0.18
False Negative Rate	0.16
Precision	0.80
Recall	0.83
F-Score	0.81

Table 8: Performance Measures for Reserved Images

By looking at the results in Table-8, we can say that the model performs reasonable on unsee data as well with the accuracy of 82%, lower False Positive Rate and False Negative rate and higher Precision, Recall and F-Score. Let us compare these results with our baseline classifiers in next section.

4.5 Comparison

For comparison, we trained different classifiers like Decision Tree, SVM and Random Forest. These classifiers were trained using the same dataset to make the comparison fair. These classifiers were implemented using *Scikit-learn* machine learning library written in Python language.

Unlike Convolutional Neural Network, which can take direct images as input, *Scikit-learn* classifiers cannot take direct image as input and requires two-dimensional array as input. So, we needed to convert each image into two-dimensional array.

4.5.1 SVM

Support Vector Machine abbreviated as SVM is a classification algorithm that can be used for regression and classification tasks. Objective of SVM is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. We can see from the image in Figure-38 below that it finds an optimal hyperplane (displayed right side of the image) that can separate the data by using number of possible hyperplanes (displayed left side of the image) and choosing the one with maximum margin. Which is maximum distance from datapoints of both classes.

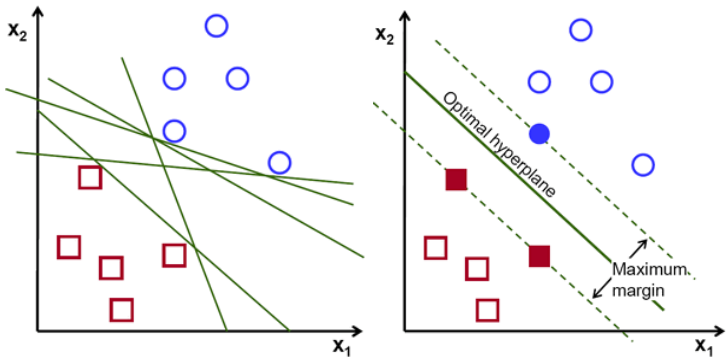

Figure 38: Hyperplanes [54]

After training the SVM using our dataset, we could achieve the accuracy of 61% which is the lowest among all classifier used for experiments. 3-fold cross validation was also used for SVM. It gave the precision of 0.68 and the recall of 0.61 whereas the f1-score of 0.56 which is closest to worst f1-score (worst f1-score is 0.5).

4.5.2 Decision Trees

Decision Trees are another Machine Learning algorithms that can be used for tasks like classification and regression. They are fundamental components of Random Forest Classifier. They can be used to classify complex data. Let us take an example from [52] on Iris dataset. Following image in Figure-39 represents a Decision Tree structure and values represented in flow chart are from Iris dataset.

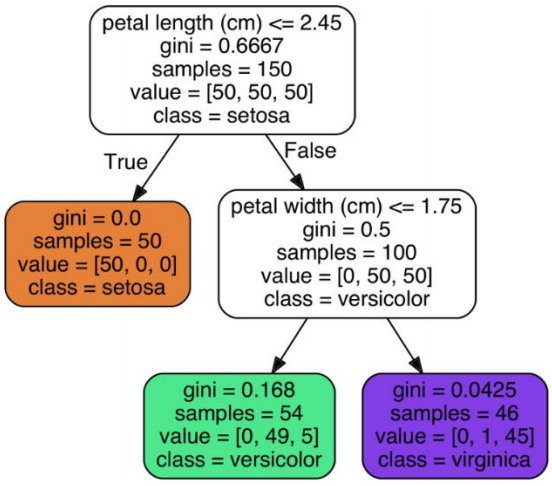

Figure 39: Decision Tree [53]

Suppose we want to classify an iris flower. The classifier starts from root node (depth 0) checking the flower's petal width, if it is smaller than 2.45 cm. If it is, then it will move down to the root's

left child node (depth 1, left) which does not have any child. So, it will not go further and predict that the image belongs to class *setosa*. If the flower's petal length is greater than 2.45 then it will move down to root's right node (depth 1, right). Since this node is not a leaf node, it will check further if petal width is smaller than 1.75 cm. If it is, then the flower will be predicted as *versicolor*. If not, then it will be predicted as *virginica.*

Decision tree was able to achieve the accuracy of 72% a little less than CNN which was able to achieve the accuracy of 82%. Accuracy was calculated using 3 cross fold validation to ensure there is no overfitting. Decision tree gave us the precision of 0.76 and recall of 0.72. If we compare it to CNN, we can see that CNN achieved better precision and recall of 0.80 and 0.82, respectively. The f1-score of Decision tree is 0.71 while CNN achieved f1-score of 0.80.

4.5.3 Random Forest

Random Forest is an ensemble of Decision Trees. Ensemble means, using multiple learning algorithms to obtain better performance. As the name implies, Random Forest consists of large number of decision trees. Each individual tree will predict and output the class of an input. After each individual tree has performed prediction, final output will be the highest number of predicted class. Let us assume that we have nine decision trees as shown in Figure-40. We can see that out of nine individual trees, six of them predicted that the class of an input is "1". So final prediction will become "1".

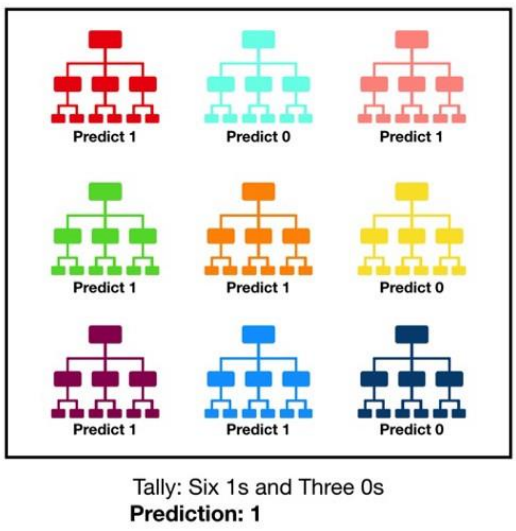

Figure 40: Random Forest [55]

Random Forest was able to achieve the accuracy of 77% which is the highest accuracy out of all three base line classifiers we used and closest to CNN which achieved 83% of accuracy. Random

Forest gave us the precision and the recall of 0.79 and f1 score of 0.78. Following Table-7 is summarizing performance measures of all classification algorithms used for experiments.

Classifier	Precision	Recall	f1-score	Accuracy
SVM	0.63	0.56	0.49	56%
Decision Tree	0.66	0.65	0.65	65%
Random Forest	0.73	0.72	0.71	72%
CNN	**0.80**	**0.83**	**0.81**	**82%**

Table 9: Comparing of Classifiers

It can be seen from the Table-9 that the CNN performed best with 82% accuracy among all classifiers and the SVM performed the worst with only 61% accuracy. Performance measures for SVM, Decision Tree, Random Forest and CNN displayed in Table-9 were calculated using reserved dataset split (300 unseen images).

As we discussed earlier, there are no research papers available currently on detecting intoxication caused by use of marijuana. However, there are many research papers available on drunk detection and drowsiness detection which are somewhat related because they also work based on the facial feature changes. So, we decided to compare accuracy with those research papers as well.

Table-10 below summarizes name of research papers used for comparison, list of methods used, brief description of tasks and accuracies from different approaches and compares with our approach.

Definition	Method	Task	Accuracy
Intoxication Identification Using Thermal Imaging [16]	Uses CNN, Infrared Images to train	Classify drunk and sober	~80%
Drunk Classification Using Infrared Face Images [15]	Fisher linear discriminant	Classify drunk and sober	~87%
Drunk Selfie Detection [12]	Android app that uses deep learning model	Classify drunk and sober	~81%
Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques [17]	Uses (MLP)	Detect drowsiness	~79%
Marijuana Intoxication Detection	**Uses CNN**	**Classify intoxicated by marijuana and sober**	**~83%**

Table 10: Comparison with Literature Review

In the paper Intoxication Identification Using Thermal Imaging [16], a dataset of 4100 images was created by photographing 41 people. In their approach they experimented with two different neural networks by training different regions of face and observed high convergence success mainly on the area of the forehead. Accuracy rate of 80% was achieved by testing the trained models on five different individuals.

In Drunk Classification Using Infrared Face Images [17], a dataset created with the help of 46 individuals was used consists of 250 images per subject and 50 images per subset (five subsets: “Sober”, “1 Beer”, “2 Beer”, “3 Beer” and “4 Beers”). Accuracy of 87% was obtained by reducing

dimensionality of data by using the Fisher Linear Discriminant Analysis and then using the Gaussian mixture model to perform classification.

Drunk Selfie Detection [12] shares more similarities with our approach of marijuana intoxication detection because it uses normal images unlike infrared and thermal images and also has a very small dataset. In this approach, they used a dataset of 212 images created by photographing 53 different people when they are sober and after they consumed one, two and three glasses of wine. Such a small dataset is not sufficient to train a model, so they augmented existing images by blurring, changing lighting, saturation and using tint to match with images they observed from searching images of bar and parties from the internet. After augmentation, they generated 2,332 images. They were able to classify images with the accuracy of 81% using Random Forest Classifier.

In Real-time driver drowsiness detection [17], they are using National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset. It contains recorded videos of subjects recorded in a variety of simulated driving scenarios like yawning, slow blink rate, conscious laughter, and dizzy dozing. From this dataset, they took 22 subjects for training and testing. The classification model used in this approach is based on Multilayer Perceptron Classifier with three hidden layers. The input layer has 136 nodes, and the number of neurons is 100. All three hidden layers have 10 neurons each as well as dropout layer with 20% rate is used between each layer to reduce overfitting. The last layer has 2 neurons (for “Drowsy” and “Non drowsy”) with *softmax* function which predicts the class of input image.

4.6 Tools

A brief description of tools like IDE for writing and implementing code, API to implement machine learning libraries and programming language used in the development of this thesis are listed in Table-11 shown below.

Tools	
IDE	Spyder 3.3.5 from Anaconda Navigator
API	Keras: The Python Deep Learning Library
API	Scikit Learn
Programming Language	Python 3.7
Other Software	ImageBatch, Adobe Photoshop

Table 11: Tools

4.6.1 IDE (Integrated Development Environment)

There are countless IDEs available for coding and implementation for different programming languages. An IDE enables programmers to write programs easily with the help of inbuild common libraries and autocomplete functions. For the implementation of this thesis, Spyder 3.3.5 available in Anaconda Navigator was used.

4.6.2 API (Application Programming Interface)

Keras machine learning library written in Python language was used to implement libraries required to create our convolutional neural network. Keras makes it easy to implement deep neural networks with the help of wide collection of machine learning libraries. It can run on top of TensorFlow, Microsoft Cognitive Toolkit, R, or Theano in our case, it is running on top of TensorFlow in the implementation of this thesis.

For implementing baseline classifiers (SVM, Decision Trees and Random Forest) Scikit Learn library also written in Python language.

4.6.3 Other Software

For image augmentation and pre-processing images like cropping, converting image formats and resizing, a software called ImageBatch was used which is based on LibGD and EasyBMP graphic libraries [48]. Adobe Photoshop was used to create and edit some of the figures.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

We have proposed a method to detect if a vehicle driver is under influence of Marijuana or not using convolutional neural network to help reducing vehicle accidents happening because of impaired driving. We took the inspiration from research papers published on drunk detection and drowsiness detection using deep neural networks and built a Convolutional Neural Network (CNN) by experimenting with different number of layers, filter sizes and tuning various hyperparameters. We also created a dataset by gathering imagers of sober faces as well as faces intoxicated by marijuana to train our model. We used image augmentation techniques like image transformation (rotating, flipping, cropping, etc.) to expand the dataset. At the end of experiments, we were able to achieve the accuracy of 82% when testing on unseen images kept aside for testing other than images used in training and validation.

Following are some highlights of contribution of this thesis.

- Demonstrated how to locate faces from video using Dlib implementation.
- Demonstrated versatility of neural networks.
- Built and trained a CNN model that can predict marijuana intoxicated face.
- Built Dataset of Marijuana intoxicated faces and sober faces' images.

5.2 Future Work

This is an initiate approach to detect marijuana intoxicated face using convolutional neural network. Hence there was limited amount of information available and no public dataset was available to train the model. Because of the lack of dataset, model is generalized on current dataset which has limited numbers of images. For this model to be implemented in practice, we will need to carryout experiments with real human faces such as by photographing participating individuals when they are sober and photographing them after they have given marijuana to smoke just like some of the research papers [8, 19, 17] discussed in literature review who created their own dataset.

Model may not perform correctly if the subject is wearing sunglasses, because it is based on redness of eye caused by smoking marijuana. To overcome this problem, we may need to consider other features like driver head movement, driver's response time, driving style, etc. In the paper, "Driver behaviour detection and classification" [5], use of different driving style is considered to detect driver's sobriety. This similar approach can be implemented in future along with our model to improve the accuracy.

Privacy is a big concern these days since everything is being online and personal information such as names, photos, addresses, etc are shared with countless third-party service providers. Since we are extracting images from driver's video, we may need to consider encrypting these videos to protect privacy of drivers.

In the future, using thermal or infrared pictures can be used to carryout experiments to see if it increases the accuracy. Use of thermal images to detect drunk person can be seen in the paper proposed by Koukiou, G. et al [16]. And use of infrared images is present in the paper proposed by Hermosilla, G. et al [17].

In one of the papers, we saw in literature review, face recognition and drunk classification using infrared face images [16] shows that the thermoregulatory system can be altered depending on moods and consumption of certain foods. Similarly, monitoring the change in temperature after smoking marijuana can be useful and can be considered to improve the performance of our model.

REFERENCES

[1] Augusto Borges Oliveira, D., & Palhares Viana, M. (2017). Fast CNN-based document layout analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 1173-1180).

[2] Ziqi Zhu, Li Zhuo, Panling Qu, Kailong Zhou, & Jing Zhang. (2016). Extreme Weather Recognition Using Convolutional Neural Networks. 2016 IEEE International Symposium on Multimedia (ISM), 621–625. IEEE.

[3] Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional neural networks for diabetic retinopathy. *Procedia Computer Science*, *90*, 200-205.

[4] Jhung, J., Bae, I., Moon, J., Kim, T., Kim, J., & Kim, S. (2018, June). End-to-end steering controller with cnn-based closed-loop feedback for autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 617-622). IEEE.

[5] Shahverdy, M., Fathy, M., Berangi, R., & Sabokrou, M. (2020). Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications*, *149*, 113240.

[6] Hashemi, M., Mirrashid, A., & Shirazi, A. B. (2020). Deep learning based Driver Distraction and Drowsiness Detection. *arXiv preprint arXiv:2001.05137*.

[7] Mehta, V., Katta, S. S., Yadav, D. P., & Dhall, A. (2019, October). DIF: Dataset of Perceived Intoxicated Faces for Drunk Person Identification. In *2019 International Conference on Multimodal Interaction* (pp. 367-374).

[8] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition.

[9] Shan Li, Weihong Deng, and JunPing Du. 2017. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2584–2593.

[10] Willoughby, C., Banatoski, I., Roberts, P., & Agu, E. (2019, July). DrunkSelfie: Intoxication Detection from Smartphone Facial Images. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 496-501). IEEE.

[11] "Davisking. (n.d.). Davisking/dlib. Retrieved January 12, 2021, from https://github.com/davisking/dlib

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[13] D'Seas, G., Rosebrock, A., JBeale, Ben, Nasir, M., Majid, . . . Anji. (2020, April 18). (Faster) Facial landmark detector with dlib. Retrieved January 12, 2021, from https://www.pyimagesearch.com/2018/04/02/faster-facial-landmark-detector-with-dlib/

[14] WINE PROJECT. (n.d.). Retrieved January 12, 2021, from https://www.masmorrastudio.com/wine-project

[15] Koukiou, G., Panagopoulos, G., & Anastassopoulos, V. (2009). Drunk person identification using thermal infrared images. 2009 16th International Conference on Digital Signal Processing, 1–4. https://doi.org/10.1109/ICDSP.2009.5201249

[16] Hermosilla, G., Verdugo, J. L., Farias, G., Vera, E., Pizarro, F., & Machuca, M. (2018). Face recognition and drunk classification using infrared face images. *Journal of Sensors, 2018*.

[17] abbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., & Jiang, S. (2018). Real-time driver drowsiness detection for android application using deep neural networks techniques. Procedia computer science, 130, 400-407.

[18] Flores, M. J., Armingol, J. M., & de la Escalera, A. (2008, June). Real-time drowsiness detection system for an intelligent vehicle. In 2008 IEEE Intelligent Vehicles Symposium (pp. 637-642). IEEE.

[19] Impaired Driving Statistics. (2020, December 22). Retrieved April 10, 2020, from https://maddchapters.ca/parkland/about-us/impaired-driving-statistics/

[20] Government of Canada, S. (2016, December 14). Impaired driving in Canada, 2015 Impaired driving in Canada, 2015. Retrieved January 12, 2021, from https://www150.statcan.gc.ca/n1/pub/85-002-x/2016001/article/14679-eng.htm

[21] Impaired Driving Incidents. Retrieved January 12, 2021, from https://www.cdc.gov/motorvehiclesafety/impaired_driving/impaired-drv_factsheet.html

[22] Team, K. Keras API. Retrieved January 12, 2021, from https://keras.io/

[23] Saha, S. (2018, December 17). A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. Retrieved January 12, 2021, from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[24] Dansbecker. (2018, May 07). Rectified Linear Units (ReLU) in Deep Learning. Retrieved January 12, 2021, from https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning

[25] Radhakrishnan, P. (2017, October 18). What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network? Retrieved January 12, 2021, from https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

[26] Ramesh, S. (2020, October 07). A guide to an efficient way to build neural network architectures- Part II: Hyper-parameter... Retrieved January 12, 2021, from https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7

[27] Omoyiwola, D. (2018, November 06). Machine Learning on Facial Recognition. Retrieved January 12, 2021, from https://medium.com/datadriveninvestor/machine-learning-on-facial-recognition-b3dfba5625a7

[28] Forson, E. (2019, June 09). Understanding SSD MultiBox - Real-Time Object Detection In Deep Learning. Retrieved January 12, 2021, from https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

[29] Autonomous Vehicles. Retrieved May 8, from https://pixabay.com/illustrations/car-automobile-3d-self-driving-4343635/

[30] Image Augmentation Sample. Retrieved May 8, from https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

[31] Fully Connected Layer. Retrieved May 8, from https://medium.com/@rohithramesh1991/convolutional-neural-network-3818bb487ce8

[32] YouTube Video Sample. Retrieved May 8, from https://www.youtube.com/watch?v=3qthyE49PKo&list=PLCJT7U1vW2dg3d-6Q4CAxKDZrimgBpnuO&index=2&t=154s

[33] Tesla Motors using deep neural networks. Retrieved May 8, from https://www.tesla.com/en_CA/autopilotAI

[34] Certain Features of a Swan. Retrieved May 8, from https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

[35] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep CTR Prediction in Display Advertising. In Proceedings of the 24th ACM international conference on Multimedia (MM '16). Association for Computing Machinery, New York, NY, USA, 811–820. DOI:https://doi.org/10.1145/2964284.2964325

[36] World Health Organization The top ten causes of death. Retrieved May 8, from https://www.who.int/news-room/fact-sheets/detail/thetop-10-causes-of-death

[37] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," Proc IEEE CVPR 2014, pp. 1867-1874.

[38] B. Falk, R. Burstein, J. Rosenblum, Y. Shapiro, E. Zylber-Katz, and N. Bashan, "Effects of caffeine ingestion on body fluid balance and thermoregulation during exercise," Canadian Journal of Physiology and Pharmacology, vol. 68, no. 7, pp. 889–892, 1990.

[39] H. Kalant and A. Le, "Effects of ethanol on thermoregulation," Pharmacology & Therapeutics, vol. 23, no. 3, pp. 313– 364, 1983.

[40] P. Buddharaju, I. Pavlidis, and I. Kakadiaris, "Pose-invariant physiological face recognition in the thermal infrared spectrum," in 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 53–60, New York, USA, 2006

[41] G. Koukiou and V. Anasassopoulos, “Face locations suitable drunk persons identification,” in 2013 International Workshop on Biometrics and Forensics (IWBF), pp. 1–4, Lisbon, Portugal, 2013.

[42] G. Koukiou and V. Anastassopoulos, “Neural networks for identifying drunk persons using thermal infrared imagery,” Forensic Science International, vol. 252, pp. 69–76, 2015.

[43] FLIR, “Tau 2 product specification,” 2014, http://cvs.flir.com/ tau2-product-spec.

[44] G. Koukiou and V. Anastassopoulos, “Drunk person identification using thermal infrared images,” International Journal of Electronic Security and Digital Forensics, vol. 4, no. 4, pp. 229–243, 2012.

[45] J. Rustemeyer, J. Radtke, and A. Bremerich, “Thermography and thermoregulation of the face,” Head & Face Medicine, vol. 3, no. 1, p. 17, 2007.

[46] Tian Z. and Qin. H.: Real-time Driver’s Eye State Detection. IEEE International Conference on Vehicular Electronics and Safety, pp. 285-289 (2005).

[47] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[48] ImageBatch batch image converter. Retrieved January 2020, from http://www.imagebatch.org/

[49] THC Breathalyzer. Retrieved January 2020, from http://www.cannabixtechnologies.com/

[50] Image Classification Using Convolutional Neural Network. Retrieved March 2020, from https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8

[51] Precision and Recall example. Retrieved March 2020, from https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

[52] Géron Aurélien, “Training and Visualizing a Decision Tree,” in *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Beijing: O'Reilly, 2019, pp. 168–169.

[53] Géron Aurélien, “Figure 6-1. Iris Decision Tree,” in *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Beijing: O'Reilly, 2019, p. 168.

[54] SVM Figure. Retrieved December 2020, from https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c

[55] Yiu, T. (2019, August 14). Understanding Random Forest. Retrieved January 12, 2021, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2

VITA AUCTORIS

NAME:	Raj Gadhiya
PLACE OF BIRTH:	Gujarat, India
YEAR OF BIRTH:	1993
EDUCATION:	Bachelor of Engineering Computer Science, Noble Group of Institutions, Junagadh, 2015
	University of Windsor, Windsor, ON, Canada, 2021