

A REPORT
ON
**‘DROWSINESS DETECTION IN
DRIVERS AND INDUSTRIAL
WORKERS’**

BY

Name of the Student

Neelabh Sinha

Manav Kaushik

ID Number

2016B5A80600P

2016B3A30472P

Prepared on completion of the
Laboratory Project Course No. EEE/INSTR F367

AT

**CSIR-Central Electronics Engineering Research Institute, Pilani
and**



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

July 2020

ACKNOWLEDGEMENT

First we would take the opportunity to thank the BITS Pilani administration and the highest order administration, Vice Chancellor Dr. Souvik Bhattacharya and Director Dr. S. K. Barai extended the opportunity to us to study at BITS Pilani, Pilani Campus. We also would thank the senate for introducing a course as a Laboratory Project through which incorporates practical learning exposure.

We would also like to thank the organization CEERI, Pilani, specifically our mentor Dr. Sanjay Singh and Research Scholar Mr. Sumeet Saurav, for giving us a chance to work in Cognitive Computing Lab, Cyber-Physical Systems. They helped us throughout to complete this project.

From the Electrical and Electronics Engineering Department, we thank our mentor, Dr. Karunesh Kumar Gupta. He always cooperated with us and supported us throughout the project. Whether it is personal or academic help, he always stood by our side.

Moving along, we would also thank our professors, Dr. Surekha Bhanot, Dr. Raj Kumar Gupta and Dr. Navneet Goyal, who guided us to understand the core concepts associated with the project and taught us useful subjects, which were Neural Network and Fuzzy Logic, Pattern Recognition, and Machine Learning respectively.

We would also like to thank the staff of CSIR-CEERI, Pilani and BITS Pilani who helped us at various places, and at last, we would like to thank each and every person who helped through the completion of this project. We extend our sincere thanks to all of them for their support and cooperation.

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,
PILANI (RAJASTHAN)**

DATE OF SUBMISSION: 03rd July, 2020

TITLE OF THE PROJECT: DROWSINESS DETECTION IN DRIVERS AND INDUSTRIAL WORKERS

ID NO.: 2016B5A30472P **NAME:** MANAV KAUSHIK **DISCIPLINE:** M.Sc. Economics + B.E. EEE

ID NO.: 2016B5A80600P **NAME:** NEELABH SINHA **DISCIPLINE:** M.Sc. Physics + B.E. E&I

Name of Mentor: Dr. Sanjay Singh **Designation:** Senior Scientist

Name of Mentor: Dr. K. K. Gupta **Designation:** Associate Professor

PROJECT AREAS: Deep Learning, Computer Vision, Video Processing, Action Recognition

ABSTRACT

Driver fatigue has become one of the key reasons for road accidents in modern days. Various surveys prove that if a driver is correctly identified as fatigued, and he, or she is timely alarmed regarding the same, the cases of accidents can be remarkably reduced. There have been various techniques adopted to identify a drowsy driver. Through this project, an in-depth study of various existing techniques of fatigue in a driver is studied, followed by developing a deep learning based model to accurately identify a driver's state using a novel technique of using spatiotemporal features of the face. It can be determined that the accuracy will remarkably increase when this technique is used.

KEY WORDS: driver fatigue, deep learning, spatiotemporal features

TABLE OF CONTENTS

Acknowledgement	1
ABSTRACT	2
INTRODUCTION	4
LITERATURE REVIEW	5
2.1. EXISTING METHODS OF DROWSINESS DETECTION	5
2.1.1. MATHEMATICAL MODELS	5
2.1.2. RULE BASED MODELS	6
2.1.3. MACHINE LEARNING BASED MODELS	6
2.1.3.1. SHALLOW MODELS	6
2.1.3.2. DEEP MODELS	7
2.2. FEATURES USED FOR DROWSINESS DETECTION	7
DATA	9
3.1 PRE-PROCESSING & DATA AUGMENTATION	10
4. MODELS	11
4.1. CONVOLUTIONAL NEURAL NETWORKS	11
4.1.1. Baseline Model:	13
4.1.2. Final Model (Fine-Tuned with VGG16 ImageNet):	16
4.1.2.1 PRE-TRAINING / TRANSFER LEARNING	17
4.1.2.2. VGG16 NETWORK	18
5. RESULTS	21
6. CONCLUSION	24
7. REFERENCES	24

1. INTRODUCTION

Driving involves the performance of a particular sequence of actions with situational awareness, as well as, quick and accurate decision making. Situational awareness is critical in driving, as direct attention is required to process the perceived cues. Monitoring attention status, therefore, is one of the most important parameters for safe driving .

Fatigue slows down human response time, which leads to inability in safe driving. A canadian survey in this topic reports 20% of fatal collisions to involve fatigue. For Pakistan, a similar survey reports 34% of road accidents due to fatigue. A US survey also quotes as large a number as 20% of fatal crashes due to a drowsy driver. For Europe, 20% of commercial transport crashes are considered to be involving a fatigued driver. All the statistics and numbers are alarming and raise a challenge for the research community to take attention and address the issue.

Due to these factors, research in the field of driver's drowsiness has been developing very rapidly, specially in the areas of driver workload estimation, followed by driver activity identification, secondary task identification and driving style recognition. Many techniques have been used in the past, and many have also been implemented by various multinational automotive companies to provide a secure driving experience.

Some of the symptoms of fatigue include: yawning, slow reaction time, eyelid closure, loose steering grip, and many more, and humans may exhibit multiple symptoms and in various combinations at various levels of fatigue, therefore a careful analysis of all those is very necessary.

This project presents a brief review of existing techniques used until now in the field of fatigue detection in drivers, and also based on developing a novel architecture using deep learning methods to detect drowsiness, using spatiotemporal features of a person's face. The technique is developed such that it is accurate as well as robust under various real-life scenarios.

2. LITERATURE REVIEW

2.1. EXISTING METHODS OF DROWSINESS DETECTION

Mathematical models, shallow models and deep models are few of the broad techniques used to driver's drowsiness detection.

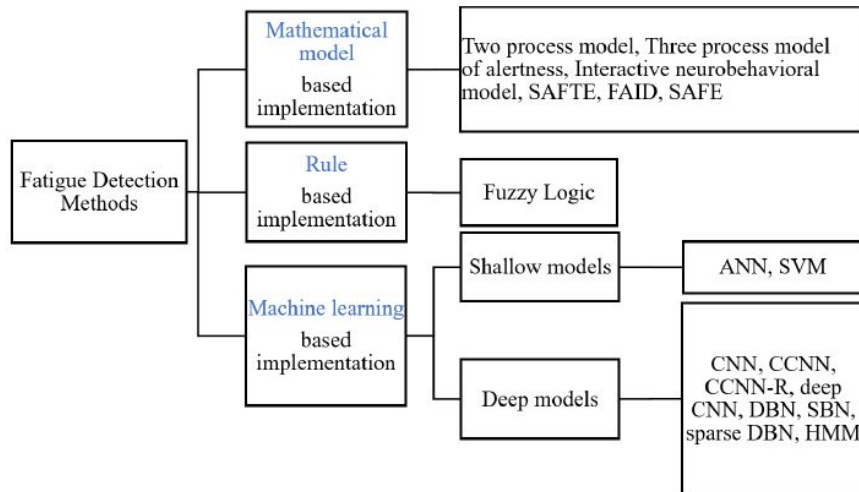


Figure. 1. Fatigue Detection Methods

2.1.1. MATHEMATICAL MODELS

Bio-mathematical work by implementing a quantitative based on sleep cycle to evaluate an individual's driving performance. Few common inputs include - circadian cycles, duration of sleep, duration of wakefulness and sleep history to predict a probabilistic value or risk, and/or a quantitative index of fatigue.

One of these models is the Two Process Model which takes two parameters, i.e., the circadian Process 'C' and the homeostatic process 'S' to predict levels of fatigue and performance. This is also further extended to a three process model of Alertness, which predicts fatigue risk and alertness based on parameters like duration of sleep and wakefulness. This computer based model considers both circadian and homeostatic components.

Various other models are developed in continuation of this which incorporate various combinations of possible inputs to predict fatigue.

2.1.2. RULE BASED MODELS

Rule based implementation is considered less challenging than mathematical systems as they are defined on definite rules. More often than not, fuzzy inference systems are used to build these complex models. FIS uses a fuzzy rule base to work in tandem with fuzzy membership functions and predicts decisions. FIS offers qualitative built-in knowledge of experts and maps inputs to outputs based on the defined set of rules. One such technique to detect drowsiness in drivers involved a fuzzy system, which Mouth and eye state as input to FIS, which was further used to predict the driver state into three classes, named as fit, fatigued and dangerous. For the same, The eyes state categories were taken as blinking, sleepy and slept, while mouth state was named as normal and yawning.

Advantages of using a FIS are:

- provides a high degree of flexibility
- offers data less training and further, provides parallel processing
- have the ability to learn by incorporating additional rules and knowledge base if desired at a later stage without much complexity

2.1.3. MACHINE LEARNING BASED MODELS

Machine learning based implementations majorly governed by the quality and amount of data, and the algorithms are trained on this extensive set of data, which are mainly acquired in laboratory and on the road testing environment. Further, the model can be described as shallow and/or deep models, based on parameters like the levels of representation and the technique used for feature extraction.

2.1.3.1. SHALLOW MODELS

Shallow models provide reasonable levels of prediction with minimal complexity. Shallow models consist of a few layers and require limited training data, but they require predefined discriminative features. Well known techniques in this domain include Artificial Neural Networks (ANN) with one hidden layer and Support Vector Machine (SVM).

In various techniques, ANNs and SVMs are trained on features like PERCLOS (% of eyelid closure), EEG and ECG signals of the driver, and other features, using in various permutations and combinations, to classify a driver as alert, drowsy or dangerous.

However, as mentioned earlier, these techniques use precomputed features. Due to inability to determine these factors accurately, deep models are used.

2.1.3.2. DEEP MODELS

Deep learning models, majorly are complex machine learning techniques which incorporate learning representation of complete dataset automatically, instead of task specific methods. Contrary to the earlier case, deep models undergo extensive training, and have the ability to extract the features on their own from the training data. Convolutional neural networks (CNN) have evolved as primary architectures to do this in the field of computer vision, and are applied extensively in Drivers' Drowsiness Detection.

Our project is based on this technique, to incorporate a deep learning based 3D convolutional neural networks to extract features from a video-based input of a driver's face to detect his/her state as active or drowsy.

2.2. FEATURES USED FOR DROWSINESS DETECTION

Various features incorporated in drowsiness detection are given in the following figure.

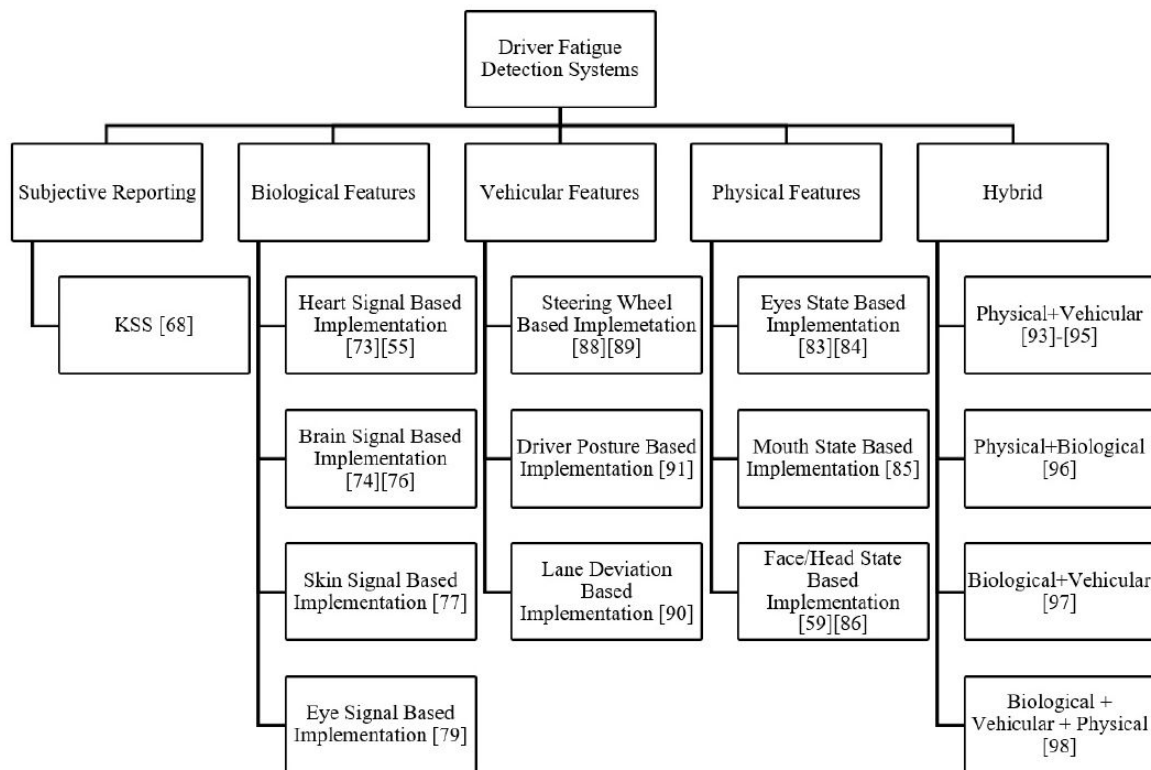


Figure. 2. Fatigue Detection Features

These features are used based on various requirements, and possess certain traits, which are, in brief, given in Table I.

On looking at various parameters associated with features to be used, we see that the biological features are intrusive in nature, due to which the driving experience and other factors of the driver will get hampered. Further, it does not have real time applicability, and the installation costs are high.

Simultaneously, vehicular features are not that accurate, because it will depend a lot on the surrounding, eg. A turning road will have a lot of movement of the steering wheel, and it can still detect this as a drowsy driver. Simultaneously, installation cost is also high in some of the techniques, and hampers real-time applicability.

Whereas, in physical features, these disadvantages are not present. There is some dependency on the brightness of the surrounding, but still, other advantages make this way better as compared to using other features. So, in this project, physical features are used to determine the state of the driver.

TABLE I: COMPARISON OF VARIOUS FEATURES

Category	Signal	Parameter	Contact	Cost	Real-time Applicability	Limitations
Biological Features	Brain	EEG	Yes	Low	No	Extremely Intrusive Prone to human movement
	Heart	ECG	Yes	High	No	
	Skin	sEMG	Yes	High	No	
Vehicular Features	Steering	SWA	Yes	High	Yes	Driver and Environment Dependency
	Lane	Lane Deviation	No	Low	Yes	
	Posture	Pressure	Yes	High	Yes	
Physical Features	Eyes	PERCLOS, Blink	No	Low	Yes	Illumination and Background Dependency
	Mouth	Yawn	No	Low	Yes	
	Face	Nod	No	Low	Yes	
	Nose	Structure	No	Low	Yes	

3. DATA

The dataset that we are using for our purpose is Driver Drowsiness Detection (DDD) dataset which was first introduced during the 2016 Asian Conference on Computer Vision. The videos in this dataset have been recorded at a 480 X640 resolution with a frame rate of 30 fps for day and 15 fps for night.

For each subject, videos were recorded in a controlled setting in five distinct conditions which are as follows:

1. without glasses
2. with glasses
3. with sunglasses
4. without glasses at night
5. with glasses at night

Different behaviours include yawning, nodding, closing eyes, regular driving, looking aside, talking and laughing, and each frame has been labelled as drowsy or non-drowsy. The dataset consists of training (18 persons) and evaluation (4 persons) sets.

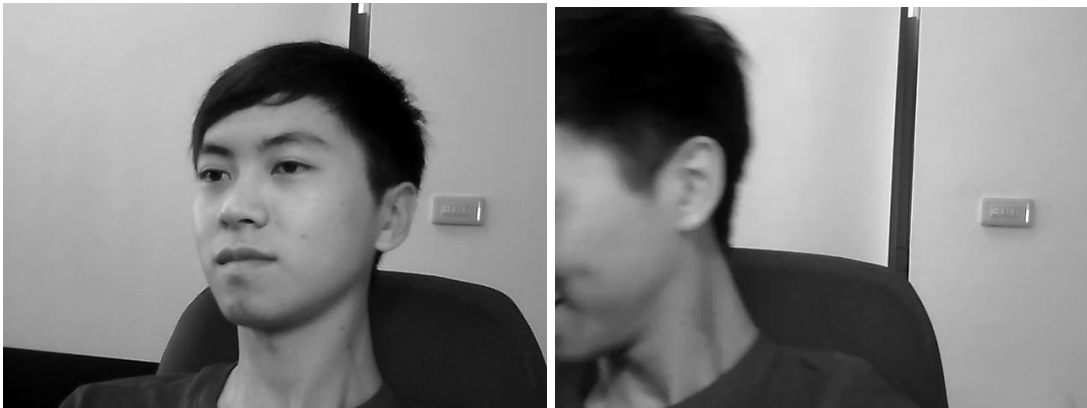




Figure 3. What our Data looks like (Drowsy and Non-drowsy case, respectively)

Our training data consists of around 8.5 hours of videos which we have used for calibrating our models. The evaluation dataset is used for testing or validation purposes (1.5 h of video). First we extract frames from all videos at 30 fps and rescale their size to 224 x 224. We have shown all the pre-processing steps in the following section.

3.1 PRE-PROCESSING & DATA AUGMENTATION

Our dataset has a good enough number of training frames which are used for calibration but these frames are highly prone to overfitting, thus we need to apply data augmentation techniques. Data augmentation solves two purpose for us:

- Reduces overfitting to a considerable extent.
- Enhances data size for better calibration.

Following preprocessing/ data augmentation steps were taken -

1. All the videos are converted into image frames at 30 frames per second
2. All the frames are labelled according the drowsiness state (i.e. drowsy or not drowsy)
3. Brightness of all the frames is normalized (by dividing from the largest value: 255).
4. Some of the frames are randomly rotated by 40 degrees.
5. Some of the frames are horizontally flipped.
6. Images are given zoom, shear, shift (height & width) of magnitude 0.2
7. The sample was rescaled to **224 x 224 x 1**. This approach achieves model invariance to translations (horizontal, vertical shifts), zooming, and face shapes.

Data after pre-processing:

Division in	No. of Frames	No. of Categories
Training Set	7,23,248	2
Test Set	1,73,299	2

4. MODELS

In this project, we have adopted the approach of using the model having the following specifications:

- Method: **Deep Model**
- Type: **Convolutional Neural Network**
- Pretrained on: **ImageNet VGG16**
- Category: **Physical Feature detection**

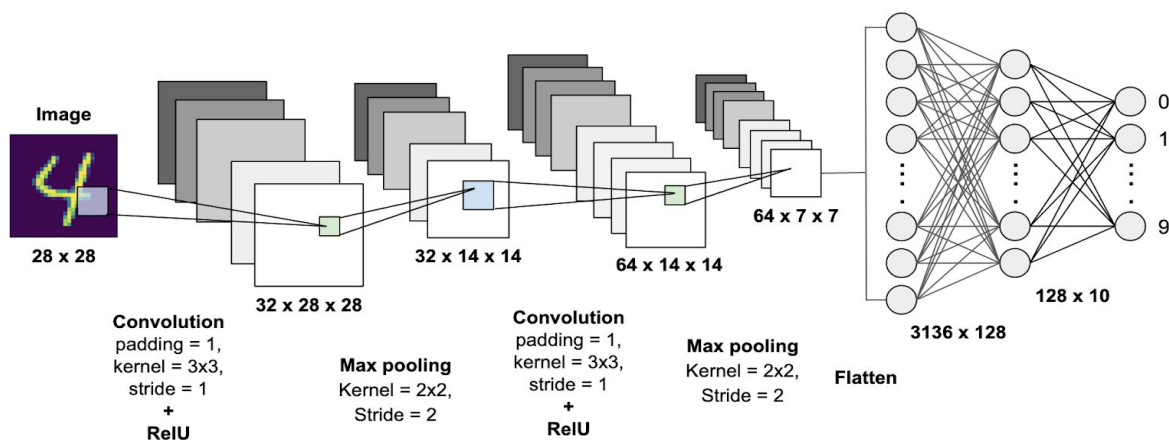


Figure 4. Model Architecture

In the following few sections, we shall explore the meaning and working of all these specifications.

4.1. CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network (CNN) is a Deep Learning technique which takes in an image as input, assigns dynamic/ learnable weights (and biases) to various regions in the image and then produces an output (classified category). The usual structure of a CNN looks as follow:

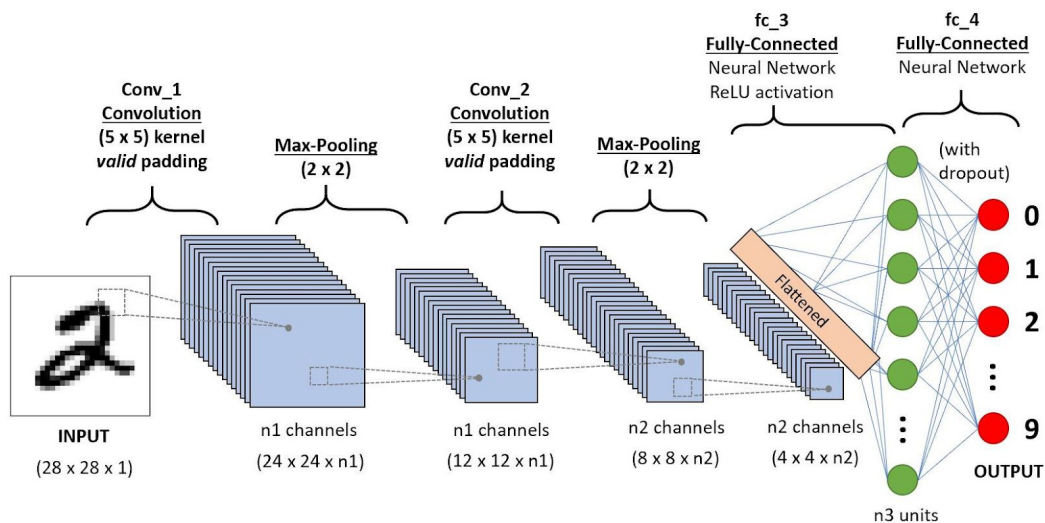
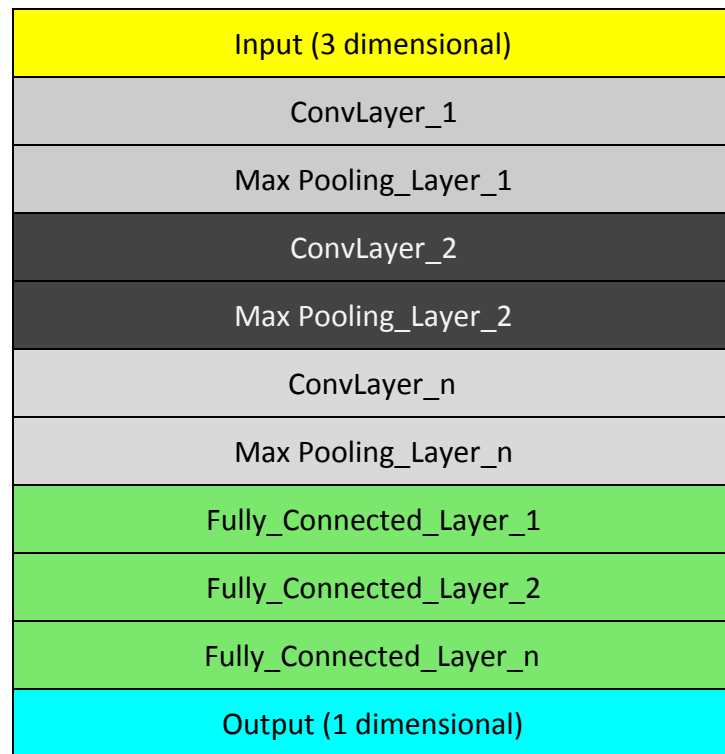


Figure 5. A CNN sequence to classify handwritten digits

All the subsequent layers are connected to each other by an activation function.

For our case we have used 2 activation functions:

- Relu (in ConvLayers and Fully Connected Layers)
- Sigmoid (for output)

We have prepared two CNN models for our purpose:

1. **A Baseline Model Trained from Scratch**
2. **A Fine tuned VGG16 Model pretrained on ImageNet dataset.**

4.1.1. BASELINE MODEL:

For the purpose of implementing a baseline model so as to obtain a model which has been trained from scratch (i.e. without any pretarining), we have prepared a CNN model with 3 Convolutional Layers and 'Relu' activation function. These layers are followed by Max-pooling layers after each Conv layer. At the top, two dense fully connected layers are attached which finally classifies a frame as drowsy or non-drowsy using a 'Sigmoid' activation function.

The structure of the Baseline Model has been shown below:

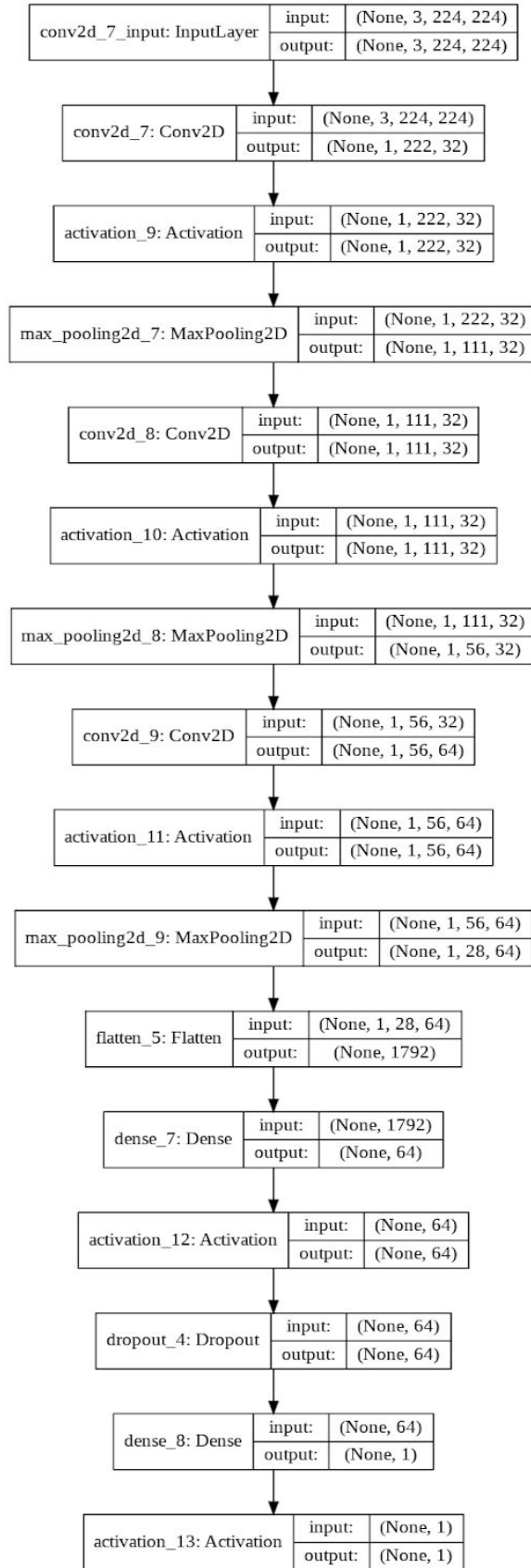


Figure 6. Baseline Model Structure

Summarizing the flow of data in this model, the following are some highlighting points:

- Input to the model: **(3 x 224 x 224)**
- Output of ConvLayers: **(1 x 56 x 64)**
- Input to Fully Connected Layers: **(1 x 1792)**
- Final Output: **0 or 1** (Not Drowsy or Drowsy)

We have trained our Baseline Model in **batches of 16** which means the total number of batches that are trained per epoch (iteration) = **45,203 batches**

Our major concern while training such networks is **Overfitting**.

To avoid this, we have predominantly used heavy Data Augmentation. But to really, tackle the issue, we also employ a Regularization technique called **L2 Regularization**. which forces the weights to change by smaller values and take small values so that no neuron of the network dominates over others, thus mitigating the risk of overfitting upto a great extent.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 1, 148, 32)	43232
activation_1 (Activation)	(None, 1, 148, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 1, 74, 32)	0
conv2d_2 (Conv2D)	(None, 1, 74, 32)	9248
activation_2 (Activation)	(None, 1, 74, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 1, 37, 32)	0
conv2d_3 (Conv2D)	(None, 1, 37, 64)	18496
activation_3 (Activation)	(None, 1, 37, 64)	0
max_pooling2d_3 (MaxPooling2D)	(None, 1, 19, 64)	0
flatten_1 (Flatten)	(None, 1216)	0
dense_1 (Dense)	(None, 64)	77888
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
activation_5 (Activation)	(None, 1)	0
Total params: 148,929		
Trainable params: 148,929		
Non-trainable params: 0		

Figure 7. Baseline Model Architecture

4.1.2. FINAL MODEL (FINE-TUNED WITH VGG16 IMAGENET):

We prepare our final model after fine tuning weights from a VGG16 model pre-trained on ImageNet dataset. The architecture of this model is as follows:

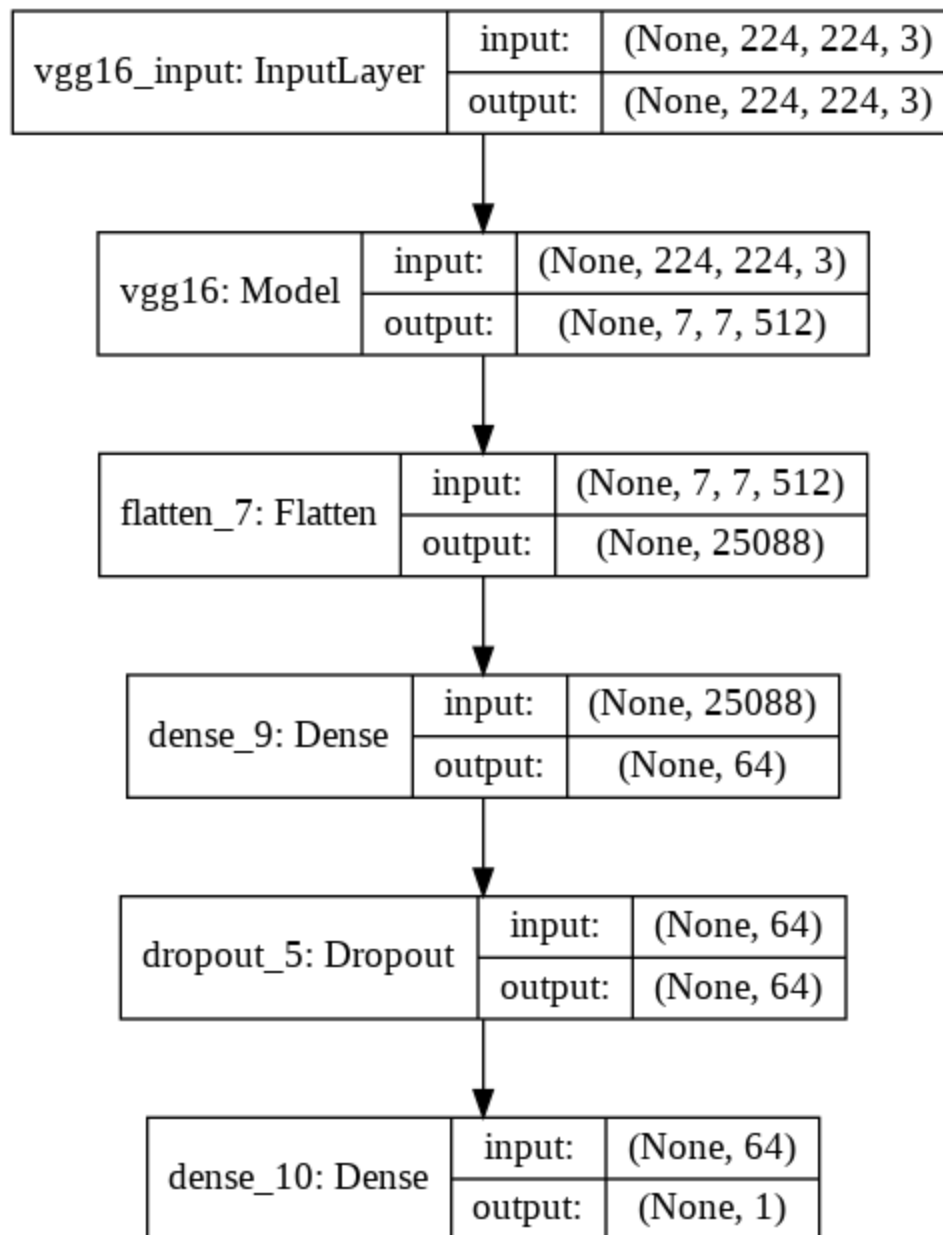


Figure 8. Final Model Structure

Now, to understand how this model works, we need to understand the concept of pre-training or the popular term: **Transfer Learning**.

4.1.2.1 PRE-TRAINING / TRANSFER LEARNING

One of the most important prerequisites, for achieving a reasonably good performance with a Deep Learning Model, is to have abundance of data which is rarely the case for most of the users, especially when it comes to visual data. That's where Transfer Learning comes to our rescue.

Transfer learning is a technique which transfers knowledge gained from the training of one model to another. Usually the former model is trained on abundant data and thus achieved a good amount of knowledge (in terms of lower level feature weights) and thus we can use the exact same weights and skip the training of ConvLayers of the later model which has scarce data.

For example, knowledge/ weights gained when learning to recognize apples can be transferred to learning how to recognize oranges.

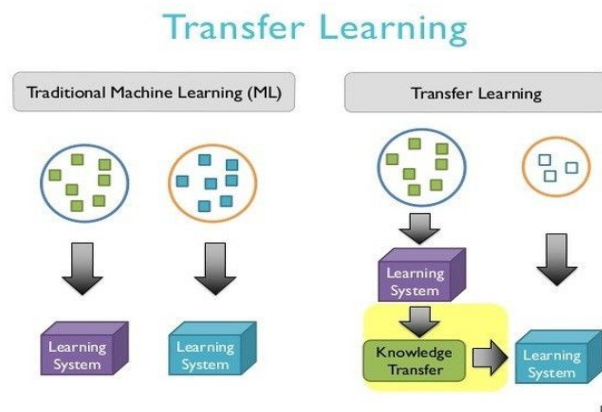


Figure 9. Transfer Learning in Practice

Usually in pre-trained models, core convolutional layers are trained on the larger dataset and their weights are fixed. After this, the final layers of the network (dense fully connected layers) are fine-tuned using our target dataset (smaller)

Due to scarcity of readily available data for our project, we are also using the application of Transfer Learning by pre-training our model on ImageNet Dataset

ImageNet Dataset: ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), which contains a huge set of images in various categories.

The total number of images in the dataset are over **14 million** distributed over more than **20,000 categories** (or labels). The dataset looks as follows:

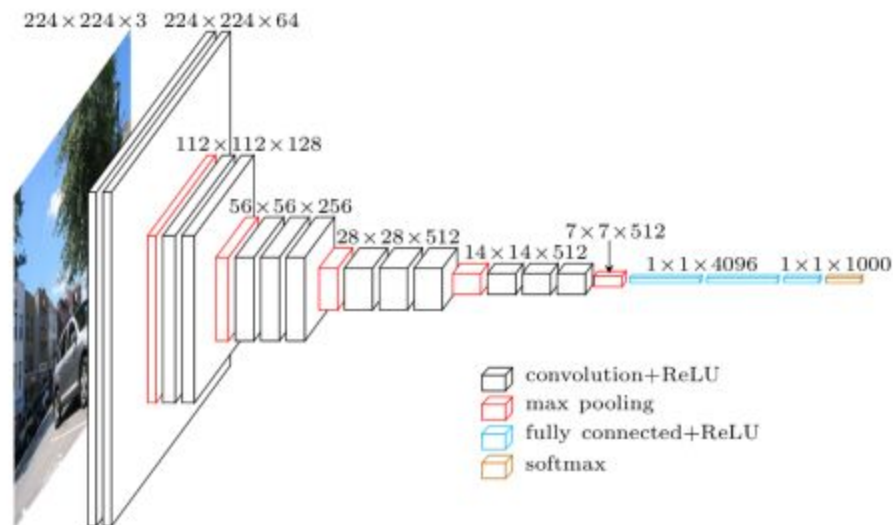


Figure 11. VGG16 Architecture

Now, we have seen that the ConvLayers are pre-trained and fixed with the weights of VGG16 networks obtained from ImageNet dataset. The final layers of the network are kept dynamic so that the model can take advantage of transfer learning while at the same time get fine tuned for the actual target (drowsiness detection) by training the fully connected dense layers.

Summarizing the flow of data in this model, the following are some highlighting points:

- Input to the model: **(3 x 224 x 224)**
- Output of ConvLayers: **(4 x 4 x 512)**
- Input to Fully Connected Layers: **(1 x 8192)**
- Final Output: **0 or 1** (Not Drowsy or Drowsy)

We have trained our Baseline Model in **batches of 16** which means the total number of batches that are trained per epoch (iteration) = **45,203 batches**

To avoid the problem of overfitting, we have predominantly used heavy Data Augmentation.

But to really, tackle the issue, we also employ a Regularization technique called **L2**

Regularization. which consists in forcing model weights to take smaller values thus mitigating the risk of overfitting upto a great extent.

Layer (type)	Output Shape	Param #
=====	=====	=====
vgg16 (Model)	(None, 4, 4, 512)	14714688
flatten_4 (Flatten)	(None, 8192)	0
dense_5 (Dense)	(None, 64)	524352
dropout_3 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 1)	65
=====	=====	=====
Total params: 15,239,105		
Trainable params: 524,417		
Non-trainable params: 14,714,688		
=====		

Figure 12. Final Model Architecture

Now, since we have seen the working and architecture of both our models, let's see how they perform on our problem dataset.

5. RESULTS

A performance comparison of the various architectures and approaches used in the literature is presented in the table below. Please note that all our experiments use the same random distortions for Data Augmentation during pre-processing. The best accuracies achieved by our respective models are as follows:

Scenario	Baseline Model	Final Model (Fine-Tuned)
No Glasses	69.33%	73.25%
Glasses	70.85%	75.64%
Sunglasses	71.35%	76.22%
Night_No glasses	67.67%	73.50%
Night_Glasses	62.42%	65.63%
ALL	68.56%	73.20%

When compared to other popular works in the literature, our approach showed true promise, which is evident from the following results:

Model / Approach	Accuracy
3D Convolution	
I3D, Carreira and Zisserman [19]	75.4%
ImageNet & Kinetics pre-trained, Wijnands et al. [20]	73.9%
2D Convolution	
InceptionV1, Szegedy et al. [18]	69.6%
MobileNetV2_1.4, Sandler et al. [16]	72.8%
Ours (Final Model)	73.20%

Moreover, we also visualize the training and testing accuracy as well as loss after each iteration. The concatenated graphs of the same have been shown below:

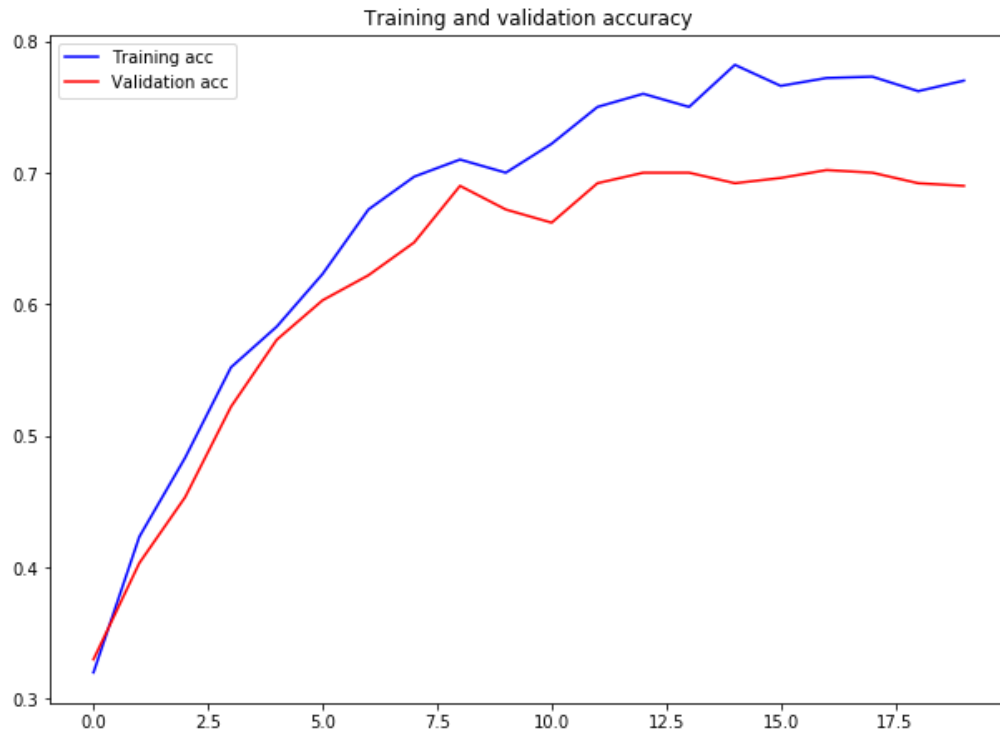


Figure 13. Baseline Model Accuracy vs #Epochs

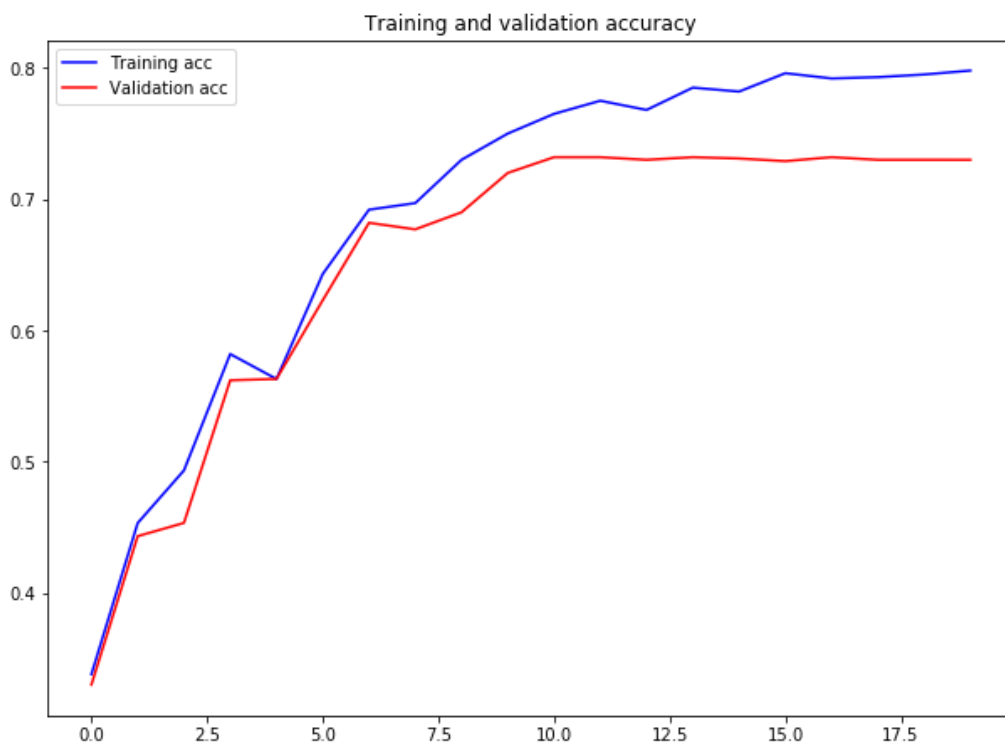


Figure 14. Final Model Accuracy vs #Epochs

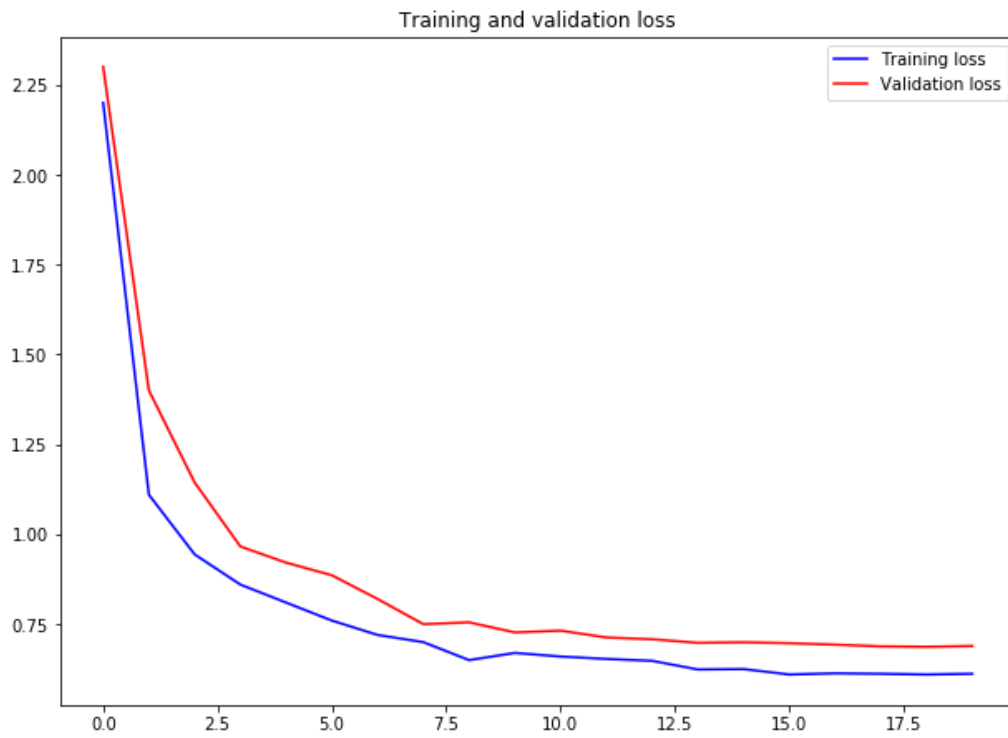


Figure 15. Baseline Model Loss vs #Epochs

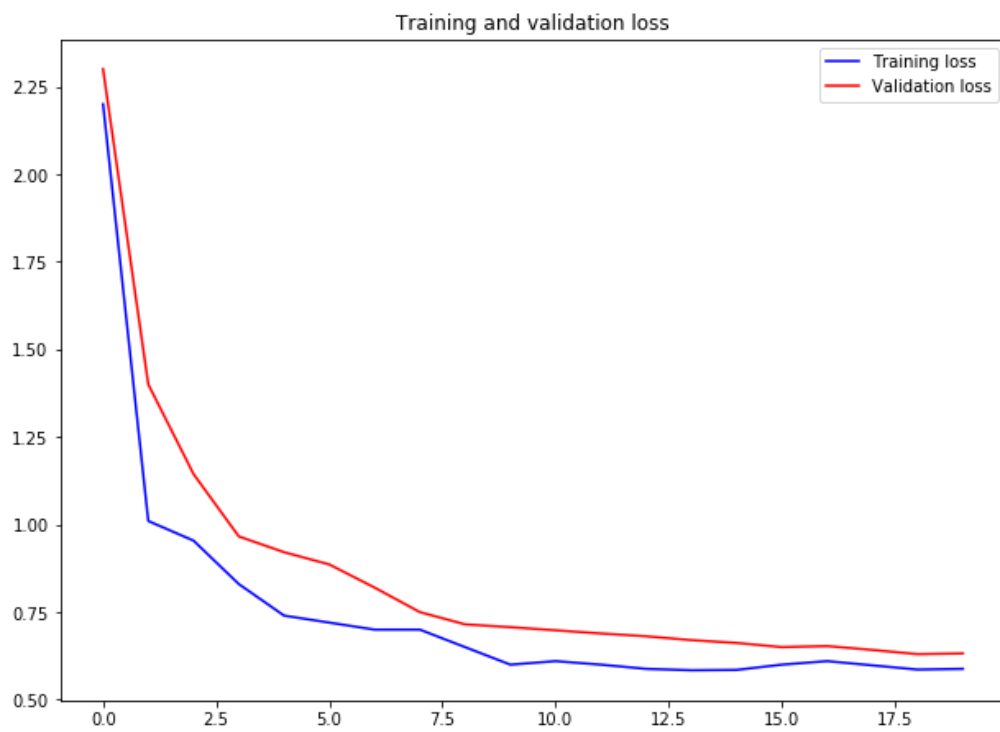


Figure 16. Final Model Loss vs #Epochs

6. CONCLUSION

- The process of drivers' drowsiness detection is very important for both individual and community safety
- The growing use of Artificial Intelligence can be immensely useful in predicting fatigue of a driver
- Our model automatically predicts a driver as drowsy or not by recognizing and key features from its face and predicting the output in real-time with an accuracy of 73.2%
- It has various advantages as the installation is simple - of a camera, and it does not hinder the driver's experience and comfort like in other techniques such as biological features or vehicle-based features
- Not prone to external disturbances, considers only the driver's face.

7. REFERENCES

- [1] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 37, no. 1, pp. 32–64, 1995.
- [2] Y. Xing, C. Lv, D. Cao, H. Wang, and Y. Zhao, "Driver workload estimation using a novel hybrid method of error reduction ratio causality and support vector machine," *Measurement*, vol. 114, pp. 390–397, Jan. 2018.
- [3] Y. Xing et al., "Identification and analysis of driver postures for in vehicle driving activities and secondary tasks recognition," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 1, pp. 95–108, Mar. 2018.
- [4] Y. Zhao et al., "An orientation sensor-based head tracking system for driver behaviour monitoring," *Sensors*, vol. 17, no. 11, p. 2692, 2017.
- [5] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 13, pp. 666–676, Mar. 2016.
- [6] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 596–614, Jun. 2011.
- [7] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver behavior analysis for safe driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3017–3032, Dec. 2015.
- [8] V. Saini and R. Saini, "Driver drowsiness detection system and techniques: A review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 4245–4249, 2014.

- [9] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: A review," *Sensors*, vol. 12, no. 12, pp. 16937–16953, 2012.
- [10] Q. Wang, J. Yang, M. Ren, and Y. Zheng, "Driver fatigue detection: A survey," in *Proc. IEEE 6th World Congr. Intell. Control Autom. (WCICA)*, Dalian, China, vol. 2, Jun. 2006, pp. 8587–8591.
- [11] (2011). Road Safety in Canada. Accessed: Mar. 24, 2017. [Online]. Available: <https://www.tc.gc.ca/>
- [12] K. Azam, A. Shakoor, R. A. Shah, A. Khan, S. A. Shah, and M. S. Khalil, "Comparison of fatigue related road traffic crashes on the national highways and motorways in Pakistan," *J. Eng. Appl. Sci.*, vol. 33, no. 2, pp. 47–54, 2014.
- [13] Schmidhuber J (2015) Deep learning in neural networks: an overview. *Neural Netw* 61:85–117. <https://doi.org/10.1016/j.neu.2014.09.003>
- [14] Silberman N, Guadarrama S (2016) TensorFlow-Slim image classification model library. <https://github.com/tensorflow/models/tree/master/research/slim>
- [15] Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds)
- [16] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: inverted residuals and linear bottlenecks. 2018 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Salt Lake City, UT, pp 4510–4520
- [17] Advances in neural information processing systems 27. Curran Associates Inc, Montreal, pp 568–576. <http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf> Soomro K, Zamir AR, Shah M (2012) UCF101: a dataset of 101 human actions classes from videos in the wild. Tech. Rep. CRCV-TR-12-01, Center for Research in Computer Vision, University of Central Florida, Orlando, FL
- [18] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, Boston, MA, pp 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [19] Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the Kinetics dataset. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, Honolulu, HI, pp 4724–4733. <https://doi.org/10.1109/CVPR.2017.502>
- [20] Wijnands, J.S., Thompson, J., Nice, K.A. et al. Real-time monitoring of driver drowsiness on mobile platforms using 3D neural networks. *Neural Comput & Applic* 32, 9731–9743 (2020). <https://doi.org/10.1007/s00521-019-04506-0>
- [21] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A

large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[22] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.