Enhancing Road Safety: A Comprehensive Approach to Detecting and Preventing Fatigue and Distraction-Related Driving Incidents

Elisa Onder and Elisa Terzini

February 5, 2024

SAPIENZA
UNIVERSITÀ DI ROMA

Contents

1 Introduction **2**

2 Our Approach **4**

3 System Design **5**

3.1 Facial Landmarks Detection 6

3.1.1 Eyes Aspect Ratio 7

3.1.2 Mouth Aspect Ratio 8

3.2 Emotion Recognition 9

3.2.1 Dataset 9

3.2.2 The model 9

3.2.3 Accuracy & Loss Graph 11

3.2.4 Confusion Matrix 12

3.3 Distraction Recognition 13

4 Interface **13**

5 Conclusion **14**

Abstract

Drowsy and inattentive drivers represent a great threat to road safety, contributing to a substantial percentage of traffic accidents globally. This project aims to raise awareness and emphasize the need for such monitoring systems with the end goal of lowering the number of driving accidents due to sleepy and inattentive drivers. Many of these accidents can be eliminated by alerting the drivers once they start feeling drowsy or when a distraction is detected. This project introduces a non-intrusive mechanism for real-time detection of driver drowsiness by leveraging visual characteristics. These features are extracted from videos. The proposed system uses facial landmarks to locate the regions of interest: mouth aspect ratio, eye aspect ratio, and head position.

1 Introduction

Drowsy driving, the dangerous combination of an operating vehicle and a fatigued driver, causes a great threat to road safety. The escalating number of road accidents attributed to drowsiness has become a matter of global concern.

According to global surveys (Figure 1), approximately 20% of European car drivers reported driving at least once in the past month while experiencing such profound sleepiness that they struggled to keep their eyes open. These data show the pressing urge to find a solution, our project aims to propose one.

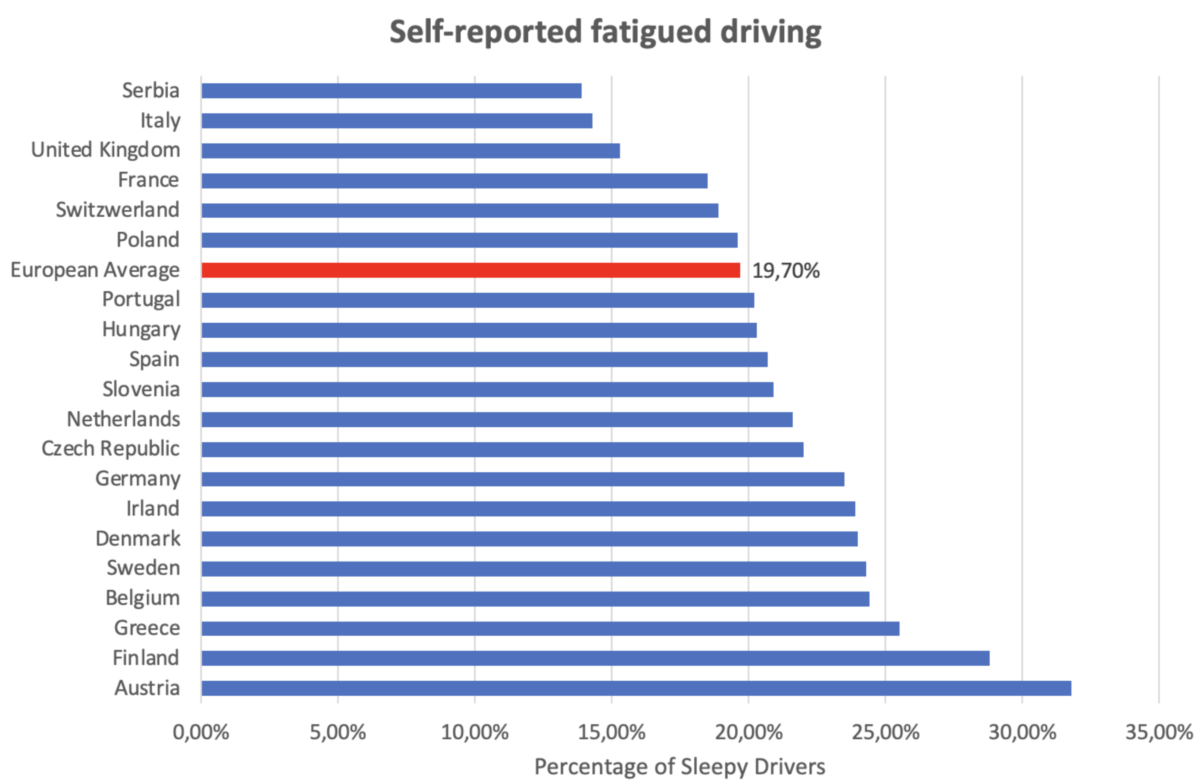

Figure 1: Self-reported fatigued driving. Source: ESRA survey, Goldenbeld & Nikolaou, 2019. [1]

In his compelling work "Why We Sleep," neuroscientist Matthew Walker investigates into the risks of driving while drowsy. In the United States, 250,000 individuals find themselves nodding off behind the wheel daily, with a concerning 56 million Americans acknowledging the struggle to stay alert while driving each month.

What we see from Walker's studies is a disturbing correlation: operating a vehicle with less than five hours of sleep triples the risk of a car crash. The danger escalates dramatically for those who have slept a mere four hours or less the night before, making them 11.5 times more likely to be involved in an accident.

What Walker emphasizes is that the link between reduced sleep hours and the heightened risk of a fatal accident is not a straightforward linear progression. Instead, it resembles an exponential growth, where each hour of lost sleep significantly magnifies the likelihood of a crash, rather than marginally increasing it.

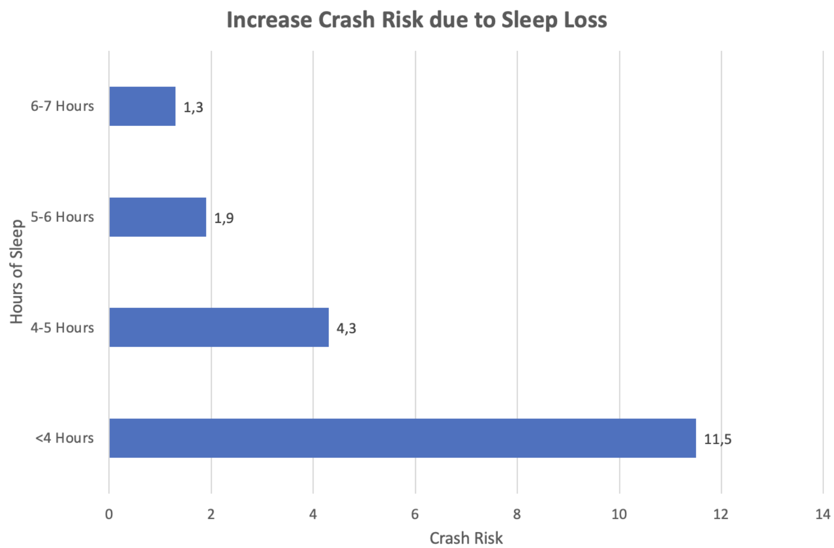

Figure 2: Sleep loss and car crashes. [2]

On the other hand; data collected in the graph below, from the National Occupant Protection Use Survey (NOPUS) conducted by NHTSA, shows that the total number of fatal distraction-affected crashes in America has increased by 11% in 2021 compared to 2020. Distraction-affected fatal crashes have also increased by 5.4% since 2011 and reached the number of 3,211.

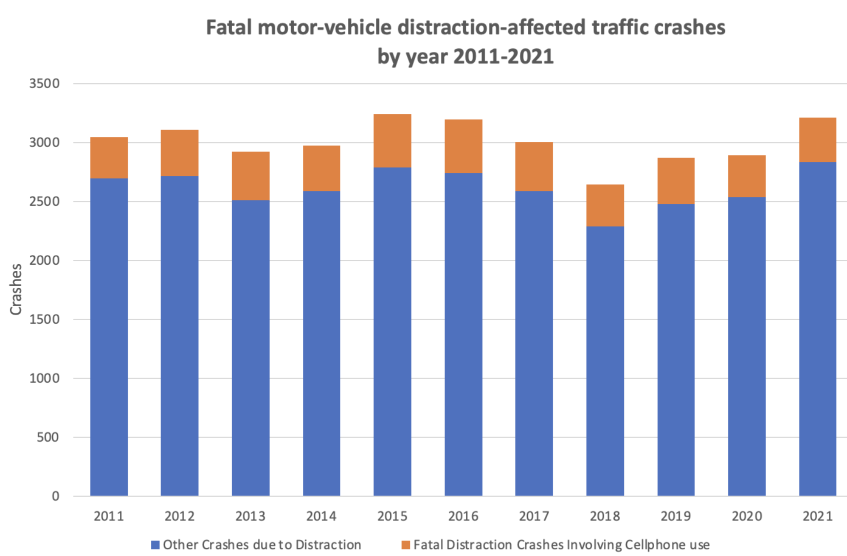

Figure 3: Fatal crashes caused by distraction. [3]

Individuals experiencing fatigue typically exhibit reduced attentiveness, slower reaction times, and less effective responses, such as increased steering movements, greater variability in speed, and altered headway distance.

Fatigue extends its influence beyond physical aspects, affecting mood and behavior, often leading to

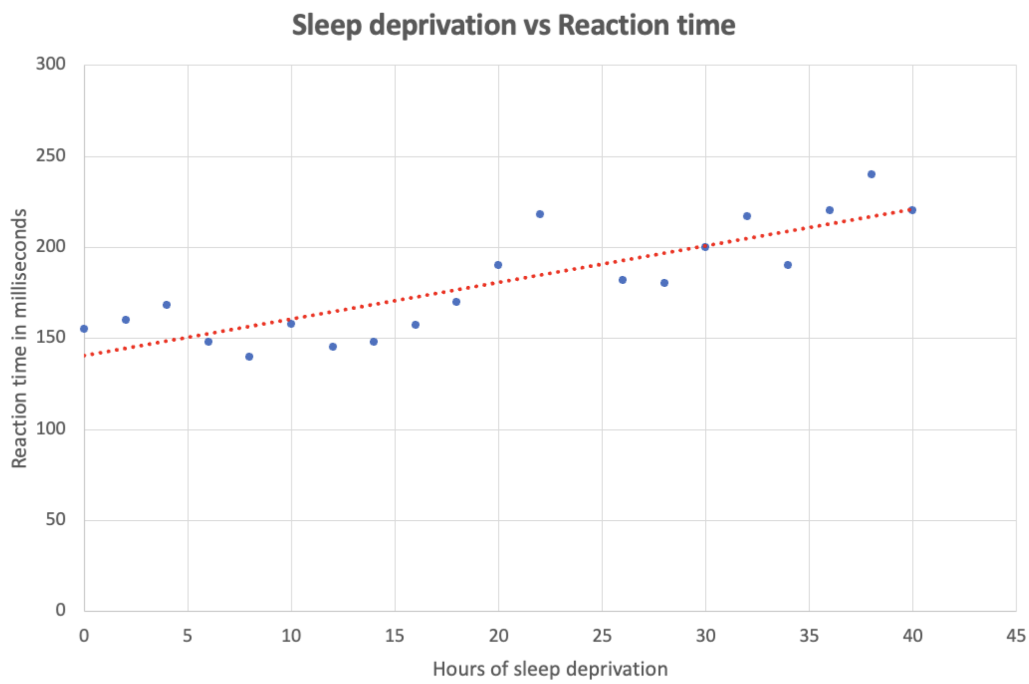

Figure 4: Linear trend between 40 hours of sleep deprivation and reaction time in milliseconds [4]

heightened irritability and frustration.

Research papers, such as "The Impact of Emotions on Driving Behavior" [5], underline the significant role of emotions in shaping how individuals behave while driving. Anger has been identified as closely linked to the driving context, exhibiting detrimental effects on a driver's behavior. Moreover, sadness and depression have been specifically linked to diminished performance, attributed to rumination and self-focus. Consequently, negative emotions exhibit nuanced and moderated adverse effects on driving, thereby raising safety implications.

2 Our Approach

Our research lead us to the main idea of this project, combining machine learning techniques for emotion detection and computer vision skills for detecting fatigue and distractions symptoms (such as yawns and the closing of the eyes for the first one and eyes not on the road for the second one). This project's aim is to propose a solution for the impending problem of car crashes, using a Python script for driver safety monitoring our project combines facial landmarks detection, emotion recognition, and drowsiness detection to assess the risk of accidents based on the driver's behavior. The script uses popular libraries such as OpenCV, dlib, TensorFlow, and Keras.

This project's primary function is to accurately detect the driver's fatigue level and promptly issue real-time alarms, thereby enhancing safety during potentially hazardous driving scenarios. The intended users of this device encompass a broad spectrum, including truck drivers, cab drivers, long-distance travelers, and individuals grappling with narcolepsy.

3 System Design

The flowchart in Figure 5 shows the design flow of the proposed drowsiness-detection and accident probability detector system. The system design consists of 4 main steps. The aim is to avoid car crushes, this project takes into account various possible causes from drowsiness, to distractions, to the emotional aspect of driving and its consequences and wants to propose possible a solution. Based on various research papers and data analyzed this project utilizes emotions as a parameter to calculate the percentage of possible car crush, negative emotions knowingly diminish driving performance [5], so when emotions such as sadness, fear, surprise and anger the probability gets increased by a certain amount (depicted in Figure 5). Then again p gets increased when yawns are detected. And if 4 yawns are detected within 60 seconds the alarm goes off. The third parameter we took into account was the detection of closed eyes while driving. we count the number of seconds the eyes of the driver are closed and if it surpasses 3 seconds the alarm goes off. Lastly the distracted driver detection uses a face detection feature which detects if the driver is not looking at the road for more than 2 seconds and makes the alarm go off.

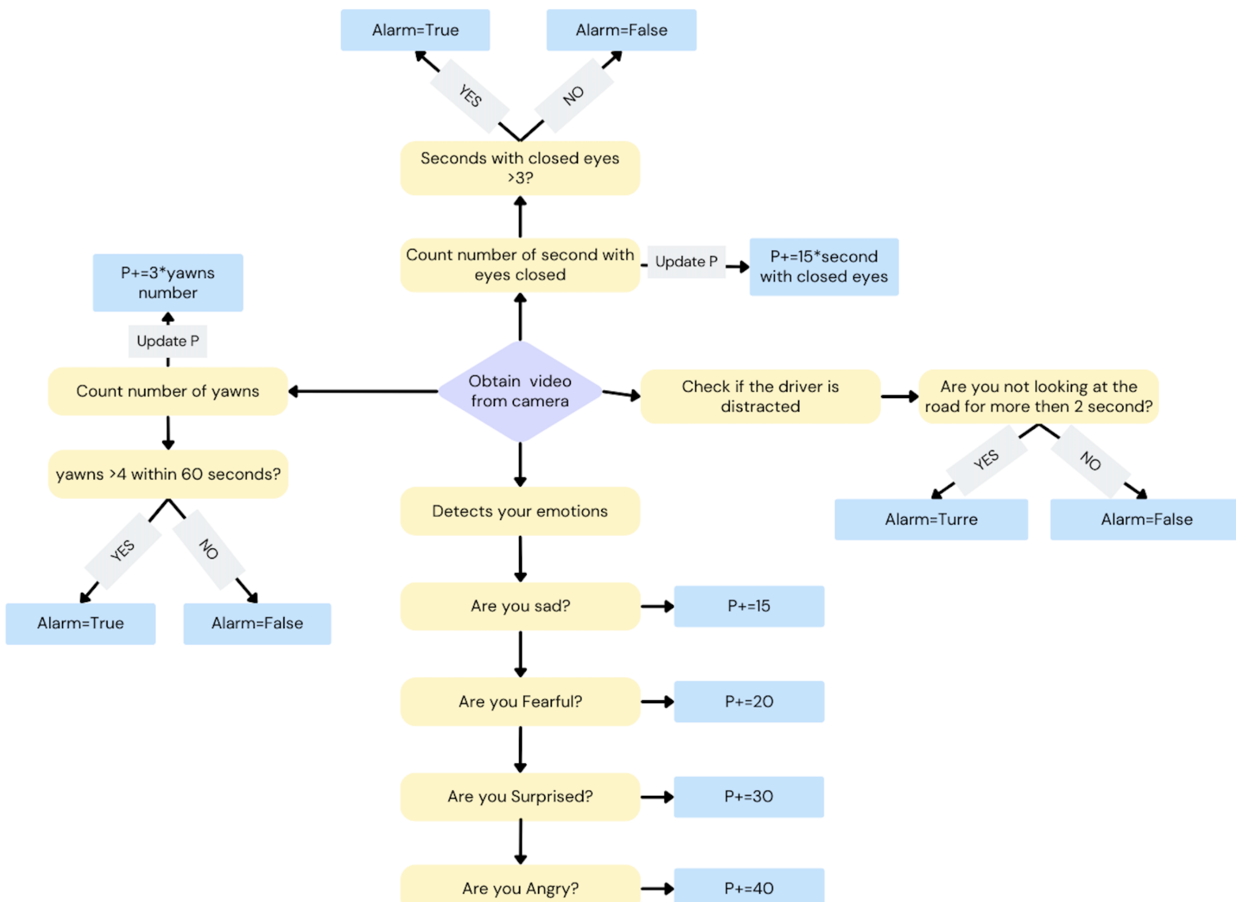

Figure 5: Schema to understand when the alarm goes off.

3.1 Facial Landmarks Detection

This project uses the dlib library to perform facial landmark detection and face detection in images. It loads a pre-trained model (shape_predictor_68_face_landmarks.dat [9]) for facial landmark prediction and initializes a face detector for identifying faces in images. The pre-trained model estimates predicts the location of 68 points on the face, which will be used for the detection of mouth and eye region. Here below Figure 6 shows the map representing the 68 points.

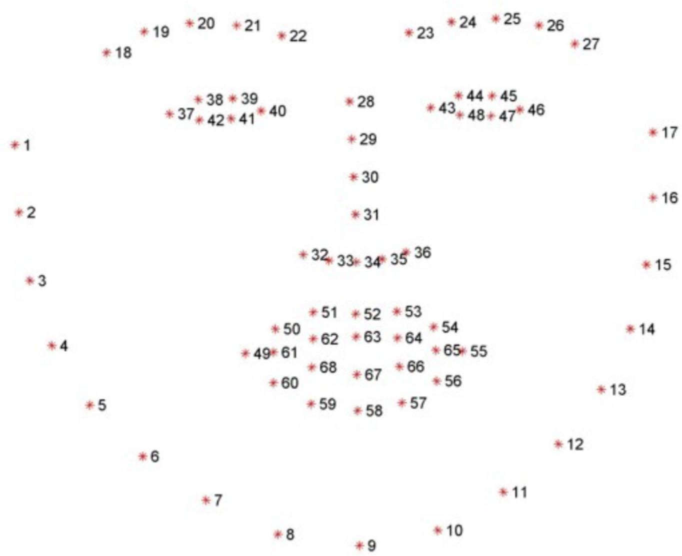

Figure 6: Facial landmarks using dlib library. [8]

Figure 7: Facial landmarks on the project's Team Members.

3.1.1 Eyes Aspect Ratio

The EAR feature relies on calculating the ratio between facial landmarks of the eyes, making it a straightforward solution. In essence, the EAR metric computes a ratio derived from the horizontal and vertical distances of six eye landmark two-dimensional coordinates, as depicted in Figure 8. These coordinates, labeled from the eye corner starting with p_0 and progressing clockwise to p_5, form the basis for the EAR computation.

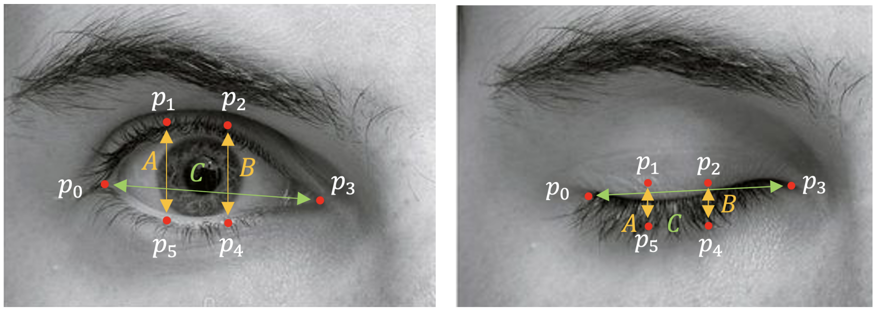

Figure 8: EAR landmarks and Euclidean distances.

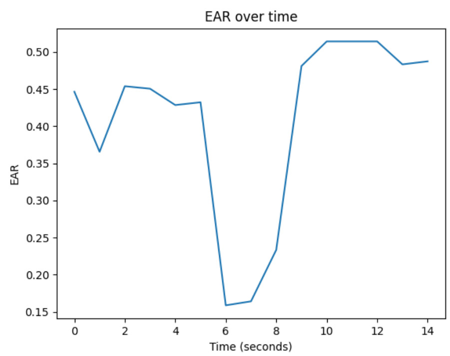

Figure 9: EAR over time.

Under the condition of open eyes, the EAR value remains relatively constant. Conversely, when the eyes are closed, the disparity between coordinates p_3 and p_5, as well as p_2 and p_6, diminishes, resulting in the EAR value dropping to zero, as illustrated in the graph.

To extract the EAR feature, Equation number was employed. As delineated in the equation, the numerator calculates the distance between vertical landmarks, while the denominator computes the distance between horizontal landmarks and multiplies it by two to maintain balance with the numerator. Utilizing (1), EAR values were computed for each frame and stored in a list.

$$EyesAspectRatio = \frac{\|\mathbf{p_1} - \mathbf{p_5}\| + \|\mathbf{p_2} - \mathbf{p_4}\|}{2\|\mathbf{p_0} - \mathbf{p_3}\|} = \frac{A + B}{2C} \tag{1}$$

The EAR tends to remain stable when an eye is open and approaches zero during eye closure. Given that eye blinking occurs synchronously in both eyes, the EAR values of both eyes are averaged, (2).

The EAR parameter will be then used as a checking point, if the parameter is smaller than or equal to 0.25 it would mean that the eyes have been closed once and a counter would be started (as shown in Figure 5). If the counter will reach 3 it'll trigger the alarm to go off making it sure that the driver won't risk falling asleep.

$$ear\_Average = \frac{\text{leftEAR} + \text{rightEAR}}{2} \tag{2}$$

3.1.2 Mouth Aspect Ratio

By calculating the mean of the selected landmarks for the top ([50-53] and [61,64] on the Figure 3 dlib landmark map) and bottom lip ([65,68] and [56,59] on the Figure 3 dlib landmark map), instead of using directly the points themselves we aim to mitigate the impact of outliers and reduce the dimensionality of the data. By then calculating the lip distance (3) we generate the parameter that will work as a yawn detector.

$$lip\_distance = |top\_lip\_center - bottom\_lip\_center| \tag{3}$$

The lip_distance will serve as the checking point to detect yawns, if the distance grows bigger than 40 a yawn will be counted and if the number of yawns reaches 5 by 60 seconds from the first yawn, the alarm will go off.

Figure 10: Lips landmarks and distance.

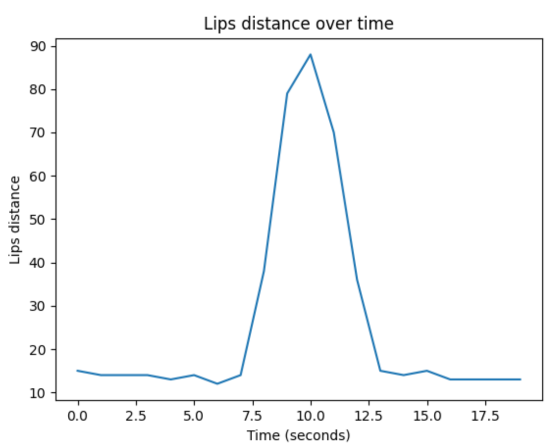

Figure 11: Lips distance over time.

3.2 Emotion Recognition

This project uses a trained convolutional neural network (CNN) implemented with Keras to recognize the driver's emotion based on facial expressions, using the Face expression recognition dataset [10]. The recognized emotion is then used to assess the risk of accidents. It calculates a risk percentage based on the number of yawns, the time with closed eyes, and the detected emotion. The risk is displayed on the screen, providing an indication of the driver's current state.

Using OpenCV then it to display a real-time video feed from the webcam, annotated with facial landmarks, yawn count, and risk percentage.

3.2.1 Dataset

The dataset used in this project [10] is composed of 35.904 48 by 48 pixeled pictures divided in training set (28.829 pictures, 80.29%) and validation set (7.074 pictures, 19.7%). Each of the sets is then divided in 7 folders representing the 7 possible labels of the classification (Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise).

Figure 12: Example pictures of the validation set of the 7 Emotions.
From left to right surprise, happy, sad, disgust, angry, fear and neutral.

3.2.2 The model

The model is a Convolutional Neural Network (CNN) designed for image classification tasks (Figure 12).

The model begins with an input layer representing images with dimensions of 48 pixels in height, 48 pixels in width, and a single channel (grayscale).

The first significant operation is a Conv2D layer with 32 filters, each of size 3x3. Convolutional layers are crucial in capturing spatial hierarchies and identifying features within the input images. The output of this convolutional operation has dimensions of 46x46, reflecting the impact of the convolutional filter on the input size. A Leaky Rectified Linear Unit (LeakyReLU) activation function is then applied, introducing non-linearity to the network and allowing it to learn complex relationships (using LeakyReLu mitigates the dying ReLU problem which refers to the problem when the ReLU neurons become inactive and only output 0 for any input).

Subsequently, a MaxPooling2D layer is employed with a 2x2 pooling size. This operation downsamples the spatial dimensions by taking the maximum value within each 2x2 region. This reduction to 23x23 helps decrease the computational load while retaining essential features. Following this, a Batch Normalization layer is introduced to normalize the activations, adding in faster convergence during training. A Dropout layer with a rate of 25% is applied, randomly disabling a quarter of the neurons to prevent overfitting.

The network then repeats this pattern with additional Conv2D, LeakyReLU, MaxPooling2D, Batch Normalization, and Dropout layers. The second Conv2D layer consists of 64 filters, and the third utilizes 128 filters. These progressively deeper layers aim to extract increasingly abstract and complex features from the input data.

After the convolutional layers, a Flatten operation reshapes the output into a one-dimensional array. The dense layer then introduces a fully connected layer and a Rectified Linear Unit (ReLU) activation function (introduces non-linearity to the model). This layer allows the network to learn complex patterns and relationships in the flattened feature vector obtained from the convolutional and pooling layers.

The subsequent Dense layer contains 512 neurons, and it is followed by a LeakyReLU activation and a Dropout operation with a rate of 50%. This dense layer serves as the bridge between the spatially aware convolutional layers and the final output layer.

The final layer is a Dense layer with 7 neurons, employing a softmax activation function. This indicates a multi-class classification task with seven possible classes. The softmax function normalizes the output into probability distributions, facilitating the assignment of class labels.

In terms of weights, the majority (91.5%) are concentrated in the final Dense layer, emphasizing its significance in making the ultimate classification decisions. Regularization techniques, such as dropout, are strategically placed throughout the model to enhance generalization and prevent overfitting.

```
         OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

             Input   #####     48   48    1
            Conv2D    \|/  -------------------       320     0.0%
                     #####     46   46   32
         LeakyReLU   ????? -------------------         0     0.0%
                     #####     46   46   32
      MaxPooling2D   Y max -------------------         0     0.0%
                     #####     23   23   32
BatchNormalization    μ|σ  -------------------       128     0.0%
                     #####     23   23   32
           Dropout    | || -------------------         0     0.0%
                     #####     23   23   32
            Conv2D    \|/  -------------------     18496     1.6%
                     #####     21   21   64
         LeakyReLU   ????? -------------------         0     0.0%
                     #####     21   21   64
      MaxPooling2D   Y max -------------------         0     0.0%
                     #####     10   10   64
BatchNormalization    μ|σ  -------------------       256     0.0%
                     #####     10   10   64
           Dropout    | || -------------------         0     0.0%
                     #####     10   10   64
            Conv2D    \|/  -------------------     73856     6.4%
                     #####      8    8  128
         LeakyReLU   ????? -------------------         0     0.0%
                     #####      8    8  128
      MaxPooling2D   Y max -------------------         0     0.0%
                     #####      4    4  128
BatchNormalization    μ|σ  -------------------       512     0.0%
                     #####      4    4  128
           Dropout    | || -------------------         0     0.0%
                     #####      4    4  128
           Flatten   ||||| -------------------         0     0.0%
                     #####        2048
             Dense   XXXXX -------------------   1049088    91.5%
                     #####         512
         LeakyReLU   ????? -------------------         0     0.0%
                     #####         512
           Dropout    | || -------------------         0     0.0%
                     #####         512
             Dense   XXXXX -------------------      3591     0.3%
           softmax   #####           7
```

Figure 13: Model for classifying emotions.

In summary, this CNN architecture is designed to process 48x48 grayscale images for a multi-class classification task. It employs convolutional layers to capture spatial features, pooling layers to reduce dimensionality, and fully connected layers for classification. Regularization techniques are incorporated to ensure robust and generalized performance during training and evaluation.

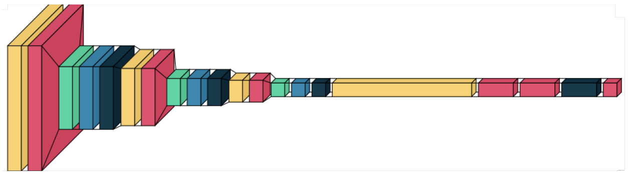

Figure 14: Model representation.

3.2.3 Accuracy & Loss Graph

Accuracy and loss curves serve as crucial tools for assessing the performance and learning dynamics of a machine learning model during training. These curves provide insights into how effectively the model is improving over successive epochs. The Accuracy curve is a representation of how precise the model's predictions are on the provided data. Essentially, it tracks the model's ability to make correct predictions as it undergoes training. The curve is plotted over epochs, with the x-axis denoting the number of training cycles and the y-axis indicating the accuracy achieved by the model. A rising accuracy curve signifies that the model is learning and improving its predictive capabilities.

Accuracy Graph An accuracy of 0.75 on the training set and 0.6 on the validation set for emotion detection suggests that the model is learning reasonably well from the training data but might face challenges when generalizing to new, unseen examples. The higher accuracy on the training set indicates that the model is capturing patterns present in the training data. However, the slightly lower accuracy on the validation set suggests that there might be some overfitting, where the model is memorizing training examples instead of learning underlying patterns.

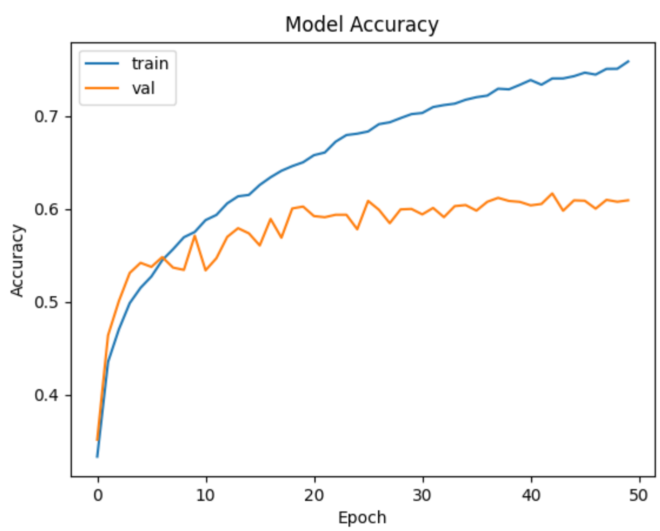

Figure 15: Training and validation accuracy graph.

Loss Graph The Loss curve measures the disparity between the model's predictions and the actual true output. It quantifies how much information the model is failing to capture or predict accurately. Similar to the Accuracy curve, the Loss curve is plotted over epochs. The goal during training is to minimize this loss, meaning the model becomes increasingly adept at making accurate predictions. A descending Loss curve indicates that the model is converging towards making more accurate predictions. A loss value of approximately 0.65 on both the training and validation sets in the context of emotion detection indicates that the model is making predictions with a moderate level of error. The loss metric represents the disparity between the predicted outputs and the actual ground truth, and a value around 0.65 suggests that the model's predictions are not perfectly aligned with the true emotions in the dataset. While the model may be effectively capturing certain patterns in the training data, the presence of a similar loss on the validation set indicates that the performance might be consistent across seen and unseen examples.

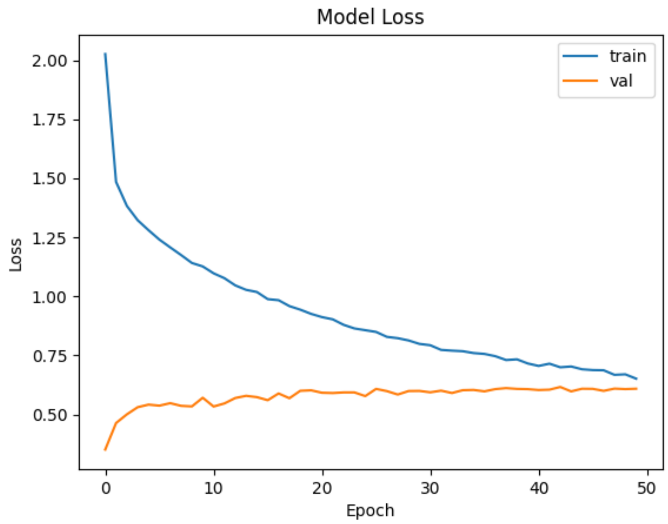

Figure 16: Training and validation loss graph.

3.2.4 Confusion Matrix

A way of analyzing specific areas where the model is performing well or struggling is the creation of the confusion matrix, the following Figure 17 shows a 7x7 matrix where each row corresponds to the true class, and each column corresponds to the predicted class.

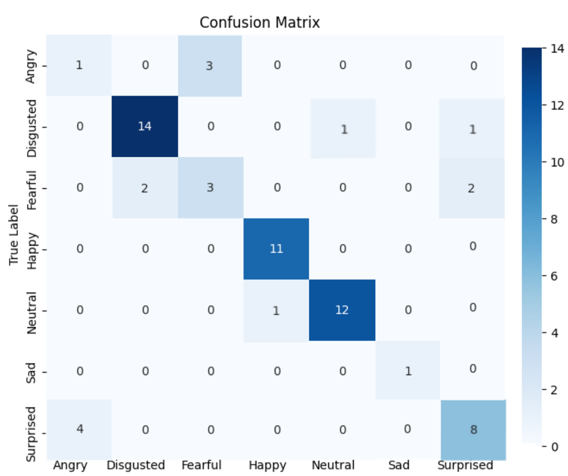

Figure 17: Confusion Matrix.

At first glance we immediately notice how the majority of instances lay on the diagonal of the confusion matrix, making it clear that our model satisfies our emotion detection necessities without that many incorrect classifications. The calculation of the accuracy (4) from the confusion matrix shows an accuracy of 0.78, similar to the accuracy reached by the validation set with the CNN model shown in Figure 15.

$$Accuracy = \frac{DiagonalValues}{DiagonalValues + NonDiagonalValues} = \frac{50}{64} = 0.78 \tag{4}$$

3.3 Distraction Recognition

This system also aims to detect instances where the driver is distracted, possibly due to activities like using a phone or changing music on the radio, resulting in the driver not looking at the road and increasing the possibility of a car accident. If I divert my attention away from the road for a mere 3 seconds while driving at a speed of 50 kilometers per hour, the consequences become tangible — covering an approximate distance of 42 meters without looking at the road. This simple calculation emphasizes the critical importance of maintaining focus on the road, as even a brief lapse in attention can result in significant distance traveled. In the context of driving safety, this serves as a stark reminder of the potential hazards associated with distractions, urging drivers to prioritize attentiveness to mitigate the risks of accidents and ensure a safer journey. To help prevent accident due to this type of distraction our system plays an alarm if the driver is not looking at the road for more than 3 seconds.

This implementation serves as a simple yet effective mechanism to alert the driver when distracted behavior, such as prolonged inattention to the road, is detected. The system can contribute to enhancing driver awareness and potentially preventing accidents caused by distractions. However, the effectiveness of such a system in a real-world scenario would depend on various factors, including the accuracy of face detection and the reliability of the alarm mechanism.

4 Interface

Our interface responds dynamically to the user's emotions, detecting the face with a white rectangle and a label with the emotion. In the upper left corner three values are displayed "The number of seconds the eyes are closed" in blue, the "Yawn count" in green and the "Accident risk" percentage in red. If the subject yawns a red notice will appear under the "Accident risk". Another notice will then be triggered as described in the system design paragraph and will show "Drowsiness detected" written in red.

Figure 18: Example of our interface.

5 Conclusion

In conclusion, this project presents an innovative solution to the pressing issue of drowsy and inattentive driving, a significant contributor to global road accidents. Leveraging visual cues and facial landmarks, the system combines machine learning and computer vision techniques to detect driver fatigue and distractions in real-time. It's important to note that the accuracy of the system depends on the quality of the trained models and the effectiveness of the chosen features for risk assessment. Additionally, user feedback and real-world testing are essential for improving the reliability and responsiveness of the system. Our approach, utilizing popular libraries like OpenCV and TensorFlow, aims to issue timely alarms, thereby reducing the risk of accidents. By addressing key factors such as emotions, yawns, closed eyes, and distractions, the project offers a comprehensive solution applicable to a wide range of drivers, including truck drivers and long-distance travelers. Overall, this system holds the potential to significantly enhance road safety by minimizing accidents caused by drowsy and distracted driving.

References

[1] Ingrid van Schagen (SWOV), European Road Safety Observatory, Road Safety Thematic Report - Fatigue.

[2] Walker, M. Why We Sleep (p.139). Penguin Books Ltd.

[3] Distracted Driving- NSC analysis of NHTSA Fatality Analysis Reporting System (FARS) data.

[4] I. Lorenzo, J. Ramos, C. Arce, M. A. Guevara and M. Corsi-Cabrera, Effect of Total Sleep Deprivation on Reaction Time and Waking EEG Activity in Man.

[5] Pêcher, Christelle & Lemercier, Céline & Cellier, Jean-Marie. (2011). The Influence of Emotions on Driving Behavior.

[6] Real-Time Machine Learning- Based Driver Drowsiness Detection Using Visual Features Yaman Albadawi, Aneesa AlRedhaei and Maen Takruri.

[7] Cech, J.; Soukupova, T. Real-Time Eye Blink Detection Using Facial Landmarks; Center for Machine Perception, Department of Cybernetics. Faculty of Electrical Engineering, Czech Technical University in Prague: Prague, Czech Republic, 2016; pp. 1–8.

[8] Rosebrock, A. Eye Blink Detection with Opencv, Python, and Dlib.

[9] Pre-trained Model: shape-predictor-68-face-landmarks.dat at this link.

[10] Face expression recognition dataset at this link.