Driver Drowsiness Detection

Karen Kuribayashi
Universität Bremen
Bremen, Germany
kar_kur@uni-bremen.de

Md. Mustafizur Rahman
Universität Bremen
Bremen, Germany
mdrahman@uni-bremen.de

Abstract— Drowsy driving is a dangerous combination of driving and tiredness. Drowsy driving is responsible for one of every four car accidents. To prevent such accidents, we propose an edge computing system which can alert the driver if he/she feels drowsy. The driver's face is detected by using a camera that captures the face of the driver. The face of the driver is given as input to an algorithm which is trained with a data set of images of opened eyes and closed eyes. This system can be used in any vehicle on the road to ensure the safety of the people who are travelling and prevent accidents. This prototype is aimed at the final semester project of the module "Edge Computing Lab" at the University of Bremen in the summer of 2022.

Keywords — Edge Computing, IoT, Machine Learning, Neural Networks and Deep Learning, Wireless Communication, Drowsy Driving, SDGs

INTRODUCTION

Edge computing has been becoming an increasingly significant part of our everyday lives with the development of IoT and machine learning. In 2022, consumer spending on smart home products and services across the globe is expected to reach $134 billion. (Howarth, 2022) We believe that using edge computing with the application of IoT and machine learning to prevent drowsy driving is a step toward creating a cost-effective option for anyone with any vehicle and background.

I. MOTIVATE PROBLEM

Drowsy driving is responsible for one out of every four car accidents. (Mounika et al.) Drowsy driving is a dangerous combination of driving and tiredness. In fact, 1 in 25 adult drivers report that they have fallen asleep at the wheel in the past 30 days, and drowsy driving results in over 71,000 injuries, 1,500 deaths, and $12.5 billion in monetary losses per year in the US. (Columbus Sleep Consultants, 2022) From this data, we can easily state that traffic accidents due to human errors cause many deaths and injuries worldwide. There is a need for a system developed with the technologies available today to overcome this situation. Meanwhile, there are cars and expensive tools for preventing driver drowsiness detection, however, we believe that they are costly and could be hard to get in developing countries. The aim of our prototype is to reduce the number of accidents by developing a model which can generate an alert if the driver is feeling drowsy so that the driver can become aware and take necessary actions for everyone without buying a new car or an expensive computing system.

II. WHY EDGE COMPUTING

Edge computing is an emerging computing paradigm which refers to a range of networks and devices at or near the user. Edge is about processing data closer to where it's being generated, enabling processing at greater speeds and volumes, leading to greater action-led results in real time. (Accenture, 2022)

We may use image detection with the concept of Edge Computing by using Machine Learning and IoT to detect if he or she is drowsy by checking whether it's regular blinking or drowsy blinking, then an alarm will be generated so that the driver will get cautious and take preventive measures. There are several reasons why we believe there are advantages to using edge computing.

Edge computing eliminates the need to move data from endpoints to the cloud and back again. By using this, it decreases the amount of time that travel shaves time off the entire process, and this time savings is needed for our case as it needs to let them know right away. Also, by computing without transferring any data anywhere else, security issues are more protected. (Microsoft, 2022)

Also, it is necessary to use the advantage of the flexibility of adding edge devices for their uses in any vehicle such as cars, motorcycles or bicycles. (Microsoft, 2022) It needs to have camera support that fits a various range of vehicles, or it needs to have a waterproof container to protect from the rain for bicycles or motorcycles which can be discussed after creating a solid prototype. However, the edge computing device itself does not need to be edited. Furthermore, it also can be attached and taken out easily.

Finally, The endpoint hardware and edge devices cost less than adding more computing resources to each vehicle. (Microsoft, 2022)

III. CONCEPTUAL DESCRIPTION OF SOLUTION

Considering the significance of this issue, we believe that the development of a system to detect drowsiness, especially in the early stages, is essential to prevent accidents. The concept of our solution is to alarm when he or she is drowsy by using an alarm so that the driver will get cautious and take preventive measures. The timing to give an alarm is absolutely important before getting into an accident. A blink is supposed to last 200-300 milliseconds, and a drowsy blink would last for 800-900 milliseconds. (adityachechani, 2022) Therefore, when a driver closes his eyes for around 800 milliseconds, the alarm goes off to wake the driver up. Since it is sometimes needed to know if the driver is able to drive safely at the moment, it is also a great idea to send a warning to the cloud so that managers or family members can get the notification when the alarm gets activated.

IV. TECHNICAL DETAILS OF THE PROTOTYPE

A. Required Software and Hardware

a) The initial plan

The programming language we used was MicroPython which is a lean and efficient implementation of the Python 3 programming language. (MicroPython, 2022) MicroPython is optimized to run on microcontrollers and in constrained environments. With the programming language, the microcontroller that we used was FiPy (esp 32) with LTE Antenna. FiPy contains five networks including LoRa in one small board and programmable with Micropython. (Pycom, 2022)

The camera that we used was ArduCAM Mini Camera Module w/ 2MP Plus OV2640 as an input. And, the audio output that we used is Active Buzzer. Active buzzers have polarity and the polarity is the same as an LED and a capacitor. (Henry, 2022) Also, for the other output, The wireless network we used to give a notification is LoRaWAN with The Things Network. LoRaWAN is a Low Power Wide Area Network specification for wireless battery-powered systems in a regional, national or even global network. (The Things Network, 2022)

b) The final product

The programming language we used was C++ with header files. The reason why we programmed our prototype in C++ is that it can work well with TensorFlow Lite to detect images and can be run independently with a battery through microcontrollers. For the battery, we used a CR2032 coin cell battery. (Allan, 2022) With the programming language, the microcontroller that we used was Sparkfun Edge. Sparkfun Edge is "Ambiq Micro's latest Apollo 3 Blue microcontroller which has an ARM Cortex-M4F 48MHz processor that can run TensorFlow Lite using only 6μA/MHz. It contains six configurable I2C/SPI masters, two UARTs, one I2C/SPI slave, a 15-channel 14-bit ADC, and a dedicated Bluetooth processor that supports BLE 5. Furthermore, the Apollo3 Blue has 1MB of flash and 384KB of SRAM memory. (Allan, 2022) The camera that we used was Himax CMOS Imaging Camera HM01B0 as an input. For output, the wireless method we chose to play audio was through Bluetooth Radio. (Allan, 2022)

B. DATASET

The dataset was used as a subnet of a dataset. It initially has 4 folders which are for training 1) Closed_eyes - having 1760 pictures, 2) Open_eyes - having 1640 pictures, and training, 1) Closed_eyes - having 240 pictures, 2) Open_eyes - having 360 pictures. (akshitmadan, 2022) With this dataset, we created a model for our machine learning system.

Fig. 1. Example of The Datasets of Closed Eyes

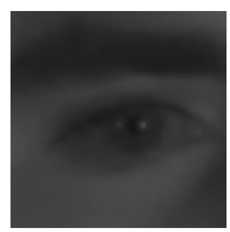

Fig. 2. Example of The Datasets of Opened Eyes

C. TRAINING MODEL

First of all, we split the database into two non-overlapping sets; the training dataset and the testing dataset which we mentioned above. After making all the input databases, 96*96 size, we used Keras to generate the model. After training the model, its performance should be measured. There are different metrics to evaluate a Classification model like Loss, and Accuracy. We use Matplotlib to plot the graphs of different metrics of the model. Testing is used to evaluate the generalization ability of the model by giving unseen data to it. We evaluated the performance of the model by finding loss and accuracy with the test data. As we did 20 epochs, we achieved okay accuracy from the model with gray scaled small images.

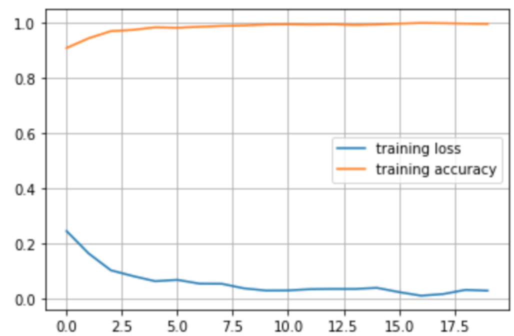

Fig. 3. The Final Performance of the Training Loss and Training Accuracy

After training a model through Keras and saving it as an h5 file, we had to convert the model to a tflite file so that we could use the model with TensorFlow Lite to make it work on FiPy. And then, In order to make it less weight for Sparkfun Edge, we run the Terminal to convert it into C++ Array.

We found out later that Keras is the recommended interface for building models in TensorFlow 2 and future versions, but does not support all for TensorFlow 1 and we believe that was the cause of kept having an error with several models including the fact some of the training was too complicated.

C. Machine Learning

Machine learning is an application of Artificial Intelligence (AI) that provides systems with the ability to learn and improve from experience without being explicitly programmed automatically. (IBM, 2022) Machine learning focuses on developing computer programs that can access data and use it to learn for themselves. Furthermore, there are also some types of machine learning algorithms that are used in particular use-cases, however, the type of machine learning algorithm we used in our system is CNN. How to generate a 250 KB binary classification model to detect if a person is present in an input image or not. MobileNets are a family of efficient Convolutional Neural Networks for Mobile Vision, designed to provide good accuracy for as few weight parameters and arithmetic operations as possible. (Warden, 2022)

Our prototype is to detect the drowsiness of the driver by programming. Through this, the system captures the driver's eyes by continuously taking an image as an input and prompts if it is closed or open. If the eyes are closed once first, and once the next image is also with closed eyes, it will play the alarm to get the driver's attention, to stop because it's drowsy.

D. Code

As we originally tried to work it out with FiPy, we used ov2640 for input, and update the firmware to add a Tensorflow for the microcontroller in MicroPython. However, once we combined all the codes for the camera, alarm, and LoRaWAN to run with the model, it crashes and the Fipy stopped working properly.

However, since we learned a lot by working with FiPy, we will go ahead and explain how to connect the microcontroller, OV2640 camera and an active buzzer. To connect the ov2640 camera to the microcontroller, we used the female and male cables in both directions as shown in the diagram below. Since Pin P22 on the FiPy that the active buzzer and the camera use are the same, a relay device has to be used to shift the pin for the active buzzer.

Camera Pin	**Pycom Pin**
CS	Pin ´P9`
MOSI	Pin ´P11`
MISO	Pin ´P14`
SCK	Pin ´P10`
SDA	Pin ´P21
SCL	Pin ´P22`
VCC	3.3V
GND	GND

Fig. 4. The Pin Usages of OV2640

As we both are really beginners at programming and working with all the devices for the first time, we decided to stick with following the example we got from an assignment to work with Sparkfun Edge as it did not have any other resources. As it loads an image as an input with a size of 96*96 pixels, it checks with the trained model on grayscale and gives a score as output to determine whether the image is an open eye or a closed eye.

SparkFun's Himax HM01B0 consumes an extremely small amount of power which is less than 2 mW when capturing at 30 frames per second. (Allan, 2022) As the prototype is taking an image and processing it constantly through a microcontroller, it should save a battery more than constantly taking a video and should last up to 10 days.

E. IoT

For our first prototype, we used LoRaWAN to get a notification once the buzzer got activated to notify the other users, for instance, if a worker is using the car, the manager of the worker can know if the worker is drowsy through the website of LoRaWAN. We used a server called The Things Stack to receive the notification from LoRaWAN. The Things Stack is a LoRaWAN Network Server which is the critical component for any LoRaWAN solution to securely manage applications, end devices and gateways.

In order to connect LoRaWAN and The Things Stack, we created an application on The Things Stack and added our prototype as an end device. By doing so, our prototype got unique IDs for application information and became available in the Europe area. For our prototype, we tested to see if it works fine at building NW1 on the campus of the University of Bremen as there is a gateway on the campus. Since for this prototype, we do not need to receive precise data, we updated the uplink payload formatter to simply print that the system got activated once the server receives a notification on the live data page.

For our final prototype, we tried to make Bluetooth work for the speaker, however, we were not able to figure it out. Therefore, instead, we decided to put LED light as the output for our initial prototype. Once the camera detects a closed eye, the LED lights up. If again, the camera detects a closed eye right after, then the LED light will change to red. Until it detects an open eye, it keeps being red.

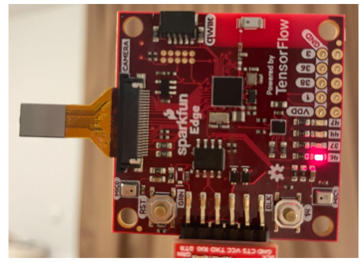

Fig. 5. The LED Output

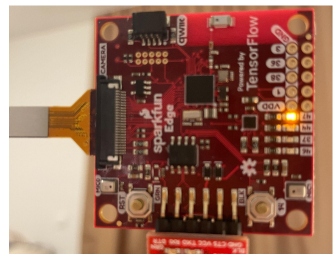

Fig 6. The LED Output

REFLECT ON OUR SOLUTION

We have developed the below model to allow the application to capture the image and send this data to the main loop, so no initial processing is required in this application. To detect drowsiness, this machine learning runs in an infinite loop. To write the input data to the tensor, the camera provides the input data. Then the interpreter runs TensorFlow lite to remodel the input data into a series of probabilities. The model, including the data, is run by the interpreter. A buzzer or LED gives the final output based on the probabilities the model runs.

Even though the model itself has good accuracy, once it runs from the device, we do not think that the prototype works efficiently enough or is accurate. Also, as we stated above to make the detection within 800 ms, it seems almost impossible to do with the current system as it needs to take 2 inputs and process them within 800 ms.

Moreover, we still need to work on making it work with the wireless part and need to get the input and process it faster.

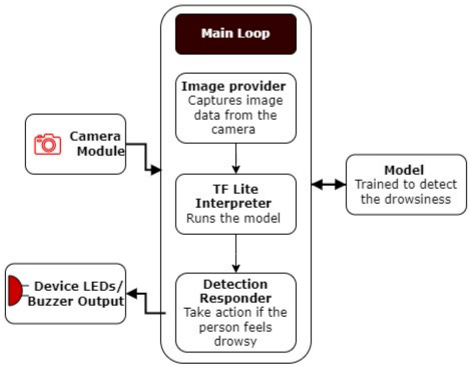

Fig. 5. Workflow prototype of the drowsiness detection application.

RELATION TO UN'S SDGS

The 2030 Agenda for Sustainable Development, adopted by all United Nations Member States, provides a shared blueprint for peace and prosperity for people and the planet with 17 Sustainable Development Goals. (The United Nations, 2022) The 11th goal is for cities and human settlements inclusive, safe, resilient and sustainable can be helped by our prototype. (The United Nations, 2022)

Research by the World Health Organization (WHO) has shown that road traffic incidents claim the lives of nearly 1.3 million people every year and are the number one cause of death in people aged 5 to 29 years old. (IISD's SDG Knowledge Hub, 2021) From the research results, providing a way to help to prevent an accident is necessary to help achieve the 11th goal of SDGs.

Also, by implementing the concept of drowsiness detection, the project also strives to nurture and support local digital innovation ecosystems and develop use cases that can be transferred to other countries to create sustainable and inclusive initiatives to accelerate inclusive digital transformation. The project will provide a way to improve digital innovation at the local and international level in support of the 9th goal of SDGs.

The drowsiness driver alarm that can be attached to any vehicle such as cars, tracks, or bicycles provides drivers to be more cautious knowing that it will send the notification to others and provides a way to give a head up when drivers get drowsy to drive safely, no matter if they have access to luxury or high tech devices or vehicles.

CONCLUSION AND FUTURE WORK

The computer vision application we developed is one step to an amazing product. This machine learning application has worked with data that is easy to understand for drowsy driving. The algorithm we developed for this neural network program can successfully detect drowsiness. To train the model, the data selection was good. After detecting the person, the device gives us a wonderfully simple output. Due to limited experience with using the Bluetooth antenna on the device, Sparkfun Edge with LED lights gives us a nice output whether the person is drowsy or not. Also initially, FiPy gives us the individual output for the buzzer, camera, and LoRaWAN. We believe this vision application prototype could develop further and help to prevent accidents by alerting drivers when they are drowsy.

We learned how to train a model from scratch given only a suitable data set for the first time. We also learned how to put the model into a shape that is optimized for embedded appliances. We believe that this tiny but vital project gives us a good foundation for solving machine vision problems that we need for more complex products.

For future work, after figuring out the output of the Bluetooth speaker to give an output with sound to actually give a good head up to the driver, it needs to get some design work done in order to attach it to any vehicle. How to attach the device would be different based on the type of vehicle as well. For example, in cars, one needs to think about how to attach it based on the shape of the car. Moreover, It needs to be water-proof for bikes and bicycles. It can get a small monitor to see where the camera is pointing for the user to set it up easier as well.

Additionally, it needs to work further on the different situations for drivers and the environment around them. For example, it is difficult to read the driver's face if they are wearing a pair of sunglasses, or when outside is extremely dark at night. From the model we developed, there was a dataset of yawning and not-yawning faces. (Perumandla, 2020) We saw several datasets that already included the images of the driver who is wearing sunglasses as well. With this, we can further implement detecting the user getting drowsy from yawning so that even if the system cannot check the eyes through sunglasses, it still could work as a detector to find that the driver is drowsy beforehand.

Fig. 5. The example of yawning faces

ACKNOWLEDGMENT

We would like to express our deep gratitude to our professors Dr. Anna Försterand and Dr. Peter Haddawy, and TAs and intern students for their guidance with unsurpassed knowledge and immense encouragement while completing the course Edge Computing Lab. We are grateful to Dr. Thomas Barkowsky for providing us with the required facilities and tech support for the completion of the project work. Also, we would like to thank our parents, friends, and classmates for their encouragement throughout our project period.

REFERENCES

[1] Accenture. "What Is Edge Computing & Why Is It Important? | Accenture." Accenture | Let There Be Change, Accenture, https://www.accenture.com/ae-en/insights/cloud/edge-computing-index. Accessed 30 Sept. 2022.

[2] adityachechani. "GitHub - Adityachechani/Driver-Fatigue-Detection: Detect Driver Drowsiness and Gaze Direction." GitHub, https://github.com/adityachechani/Driver-Fatigue-Detection. Accessed 30 Sept. 2022.

[3] Allan, Alasdair. "Introducing the SparkFun Edge - Hackster.Io." Hackster.Io, Hackster.io, 6 Mar. 2019, https://www.hackster.io/news/introducing-the-sparkfun-edge-34c9eb80a000.

[4] Columbus Sleep Consultants. Sleep & Driving. https://columbussleep.com/sleep-driving. Accessed 30 Sept. 2022.

[5] grahamPatico. "GitHub - GrahamPatico/Micropython-Ov2640-Pycom: A MicroPython Class for the Ov2640 Camera." GitHub, https://github.com/grahamPatico/micropython-ov2640-Pycom. Accessed 30 Sept. 2022.

[6] Happiest Minds. "Convolutional Neural Networks (CNN) with Deep Learning." HappiestMinds, https://www.facebook.com/pages/Happiest-Minds-Technologies/115759811859178, https://www.happiestminds.com/insights/convolutional-neural-networks-cnns/#:~:text=Convolutional%20Neural%20Networks%20(CNNs),like%20a%20human%20brain%20does. Accessed 30 Sept. 2022.

[7] Henry. "Difference between ACTIVE BUZZERS & PASSIVE BUZZERS - The Engineering Knowledge." The Engineering Knowledge, https://web.facebook.com/The-Engineering-Knowledge-717699708674235/?modal=admin_todo_tour, 28 June 2022, https://www.theengineeringknowledge.com/difference-between-active-buzzers-passive-buzzers/.

[8] Howarth, Josh. "Smart Home Market Value & Industry Growth (2022-2032)." Exploding Topics, Exploding Topics, 24 Aug. 2022, https://explodingtopics.com/blog/smart-home-market.

[9] IBM. "What Is Machine Learning? | IBM." IBM - United States, https://www.ibm.com/cloud/learn/machine-learning. Accessed 30 Sept. 2022.

[10] IISD's SDG KnowledgeHub. "Guest Article: Driving Home the Importance of Safe and Sustainable Mobility | SDG Knowledge Hub | IISD." SDG Knowledge Hub | Daily SDG News | IISD, Nov. 2021, https://sdg.iisd.org/commentary/guest-articles/driving-home-the-importance-of-safe-and-sustainable-mobility/.

[11] Kaggle. "Drowsiness_dataset | Kaggle." Kaggle: Your Machine Learning and Data Science Community, https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset. Accessed 30 Sept. 2022.

[12] MicroPython. "MicroPython - Python for Microcontrollers." MicroPython - Python for Microcontrollers, https://micropython.org/. Accessed 30 Sept. 2022.

[13] Microsoft. "What Is Edge Computing and Why Is It Important? | Microsoft Azure." Cloud Computing Services | Microsoft Azure, https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/. Accessed 30 Sept. 2022.

[14] Mounika, D., et al. A DRIVER DROWSINESS DETECTION FRAMEWORK USING DEEP LEARNING HEURISTIC. 2021.

[15] Perumandla, Dheeraj. "Drowsiness_dataset | Kaggle." Kaggle: Your Machine Learning and Data Science Community, 2020, https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset.

[16] Pycom. "FiPy - Pycom -Five Network Development Board for IoT." Pycom, http://www.facebook.com/pycomiot, https://pycom.io/product/fipy/. Accessed 30 Sept. 2022.

[17] The Things Network. "The Things Network." The Things Network, https://www.thethingsnetwork.org/. Accessed 30 Sept. 2022.

[18] The United Nations. "THE 17 GOALS | Sustainable Development." Home | Sustainable Development, https://sdgs.un.org/goals. Accessed 30 Sept. 2022.

[19] Warden, Pete. "9. Person Detection: Building an Application - TinyML [Book]." O'Reilly Online Learning, O'Reilly Media, Inc., https://learning.oreilly.com/library/view/tinyml/9781492052036/ch09.html#idm45271175481096. Accessed 30 Sept. 2022.