System-on-Chip Based Driver Drowsiness Detection and Warning System

Berkay Yazıcı, Arda Özdemir, Tuba Ayhan
Electrical and Electronics Engineering Department
MEF University
Istanbul, Turkey
yazicib, ozdemira, ayhant @mef.edu.tr

***Abstract*—The aim of this project is to detect the drowsiness level of the driver in the vehicle, to warn the driver and to prevent possible accidents. Percentage Eye Closure (PERCLOS) and Convolutional Neural Network (CNN) are used to detect drowsiness. The system is implemented on Xilinx PYNQ-Z2 development board. The system is tested under real world conditions in real time. A high accuracy rate of 92% and a fast working system with 0.8 s is achieved. A speaker is activated to warn the driver when drowsiness is detected. Moreover, the drowsiness information is sent to the cloud by using a Wi-Fi module.**

***Index Terms*—artificial intelligence, machine learning, SoC, driver drowsiness detection**

I. Introduction

Many accidents happen every year, and these accidents cause many injuries and deaths. Drowsy drivers often cause these accidents. According to a study conducted by the AAA foundation, around 328,000 drowsy driver-related accidents occur annually in the USA, and 109,000 of these accidents result in injury [1]. The number of vehicles in the world is increasing every year and these vehicles are actively used in traffic. The risk of traffic accidents also increases. These accidents not only damage public health and safety, but also have many social, environmental and economic effects. For this reason, the importance of driving safety systems in vehicles has increased considerably in recent years. Many systems such as automatic braking, blind spot detection system, 360-degree cameras and drowsiness detection systems began to take place in vehicles. With these systems, the number of accidents started to decrease.

Driver drowsiness systems are also one of the technologies that have become very popular in recent years and have started to be used in new generation vehicles. These systems, which work with various analysis methods, have advantages and disadvantages compared to one another. Considering these, we designed a succesful, low-cost and ready-to-use system on a PYNQ Z2 board [2]. The system processes camera input on the edge and generate a sound alarm promptly (under 1 second), as well as a warning over the cloud.

This paper is organized as follows. Literature on driver drowsiness systems and frequently used algorithms in these systems are reviewed in Section II. Moreover, the hardware and tool choices based on this review is given at the end on Section II. The algorithms which provide high accuracy are mapped on a single low-cost board without sacrificing the timing requirements. Hardware implementation is given in Section III. Finally, 3 hardware implementations are compared in terms of accuracy and throughput in Section IV and the paper is concluded in Section V.

II. System Modelling

A. Literature Review

There are many studies in the literature on driver drowsiness analysis using different methods. In a study conducted at the University of Alcalá, it detects the driver's drowsiness level by measuring the driver's pressure on the steering wheel and the driver's heart rate with sensors [5]. However, there is no visual control in this system. Even when the driver's eyes are closed, it is possible to still apply high pressure to the steering wheel and have a normal heartbeat.

In another study conducted at Tamkang University, the drowsiness detection of the driver was made with camera images taken from the driver. In this study, driver's current drowsiness situation was analyzed by looking at the state of the eyes [6]. However, as a result they achieved 88.9% precision rate.

In another study conducted in 2017, information was given about many driver drowsiness detection systems used in the world. For example, in the drowsiness detection system used in Ford brand cars, it gives a score between 1 and 5 points based on whether the driver brakes hard, changes lanes constantly, and moves in a straight direction. While 5 points indicate that the driver is concentrated, 1 point indicates that the driver is sleepy or tired [7]. However, in such a system, it is a disadvantage that the system needs to collect information about the new driver again in case of a driver change, and these systems are usually added to the vehicles as an option at a high cost.

B. Comparison of Driver Drowsiness Applications

In the Driver's Fatigue and Drowsiness Detection to Reduce Traffic Accidents on Road study, an accuracy of 81% was obtained with the SVM. However, when SVM, eye/mouth area confinement, and circular hough transform is used together, accuracy is increased up to 86%.

In the Real-Time Driver Drowsiness Detection System study, 80% accuracy was obtained in the naive Bayes method, 80% and 84% accuracy in the support vector machine method. When facial landmark detection and EAR (eye aspect ratio) methods were used, 84% success was achieved.

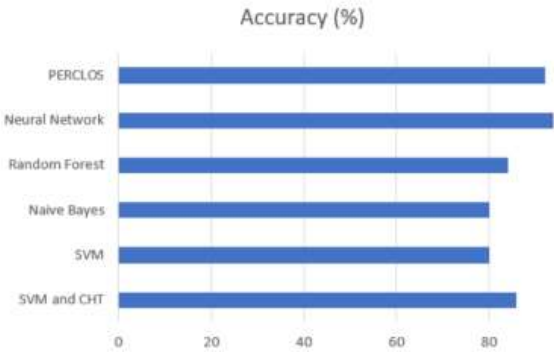
In the driver drowsiness detection using ANN image processing study, the neural networks were used. Hidden layer artificial neural network and deep learning autoencoder neural networks were used and an accuracy rate of 94% was achieved.

2022 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-8894-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASYU56188.2022.9925481

In the driver analysis study performed with the PERCLOS method, an accuracy rate of 92% was obtained. This one is the second best performing method in the given methods [8]. The accuracies of these models are compared in Table 1.

C. Methods

TABLE I. COMPARISON OF THE METHODS USED IN DRIVER DROWSINESS DETECTION STUDIES [7]



1. CNN Algorithm

CNN is a structure consisting of many layers that detect different features in the photo. Filters are applied to each image used for training and convolved images are used in the other layer. Starting from details such as edges and light level, more complex features are obtained by the filter and in this way, it helps us to recognize the object in a unique way. The CNN algorithm is a useful algorithm because it does not require manual feature extraction, has a high accuracy rate, and can be retrained, so it can be built on pre-existing networks.

However, since the CNN algorithm requires high computational power and memory, we decided that it is not suitable for use on our current development board. The advantages of the current system are explained in the results section.

2. Facial Landmarks

The parts as face, eyes, mouth and nose can be detected on a image with the facial landmarks method [10]. The model required for this method was obtained with facial images containing 68 different facial landmarks in the iBUG-300 W dataset [11]. These points can be seen in Figure 1. In this way, we can detect parts of the face such as eyes, mouth and nose. For example, in order to detect the left eye, the points from 43 to 48 in the Figure 3 needs to be detected on a given image. In this way, the analyzes can be done in the next step in order to define the status of the eye.

Fig. 1. Facial landmarks [10]

3. Eye Aspect Ratio (EAR)

Eye Aspect Ratio (EAR) is one of the most commonly used methods to detect whether the eye is open or closed. As mentioned in the literature review, it is one of the methods with a high accuracy rate. It is possible to determine the EAR by using the 6 points on the eye seen in Figure 4 by using the formula given in Eq. (1) below. If the EAR is constant within a certain period of time, we can say that the eyes are open, while when the eye is closed, EAR suddenly drops to zero, as can be seen from the image in the upper right on Figure 2. If the EAR value does not increase again after decreasing to zero, it will be determined that the driver does not open her eyes after closing and if she stays in this state for a long time, she sleeps. When we look at the graph below in Figure 2, blinking process with EAR throughout the given time period can be observed [12].

$$EAR = (||p2 - p6|| + ||p3 - p5||) / (2 * ||p1 - p4||) \quad (1)$$

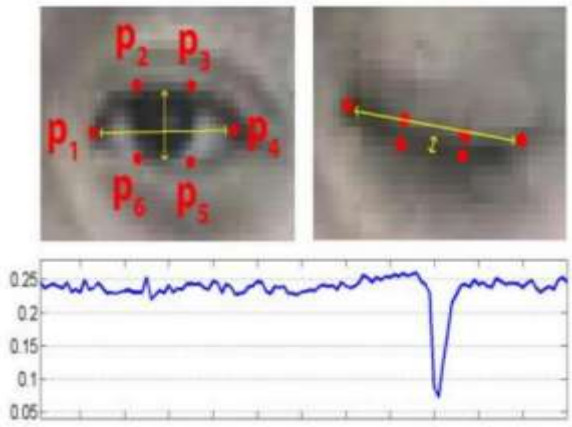


Fig. 2. Detection of blink with EAR [11]

4. Haar Cascade Classifier

There are already many algorithms for face and eye detection. However, they have their strengths and weaknesses. Some algorithms are too complex and take a long time to run. Haar Classfier makes up for these algorithms. Thanks to this algorithm developed by Viola and Jones, object detection can

978-1-6654-8894-5/22/$31.00 ©2022 IEEE

be done quickly. It is possible to detect face and eyes with AdaBoost classifier built on Haar cascade classifier [13].

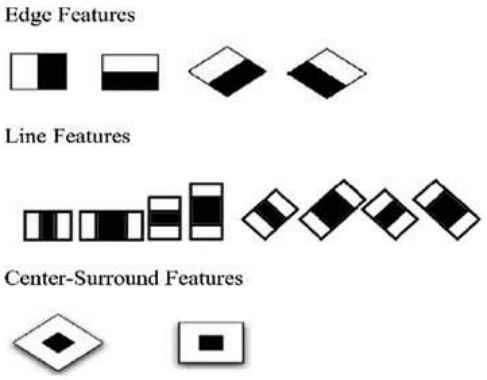


Fig. 3. Haar features [13]

Filters frequently used in haar cascade classifier algorithms can be seen in Figure 3. These filters have rectangles to be applied to the image. Haar like features are found by adding the difference of pixel values in black and white rectangles. The relevant formula is given in Eq. (2) [14].

$$f(x) = \sum Pixel\ Gray\ Level\ (Black\ Rectangle) - \sum Pixel\ Gray\ Level\ (White\ Rectangle) \quad (2)$$

Haar-like features are calculated with the integral image method. This method is adding the pixel values remaining in the top-left part of a point in the picture. In Figure 4, this area is shown in black. OpenCV library is used for this process. With this library, weak classifiers are eliminated at every stage and at the end all weak classifiers are eliminated. Afterwards, training is done with the AdaBoost algorithm. This algorithm estimates the region of the face. The model is trained with positive images (images with faces) and negative images (images without faces) and a cascade classifier is obtained. Then an XML file is created and face detection applications can be made using this file.

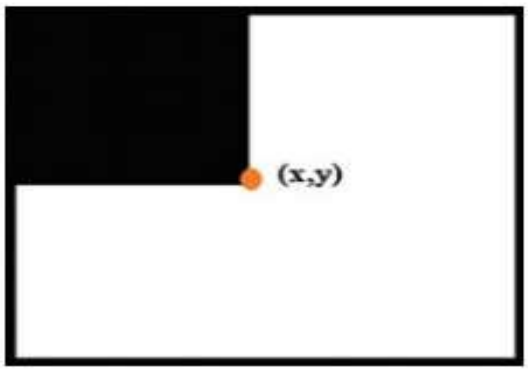


Fig. 4. Integral image application from a point [14]

D. Project Definition

In this paper, we determine the driver's status (drowsy or not) by taking the image of the driver with the help of the camera and then analyzing it with artificial intelligence, to warn with an alarm sound and then to send the current driver situation to the cloud system. The main three components are development board, camera and the communication module. Also, there are additional components such as speaker to give warning and a battery to power up the system.

We use the Xilinx PYNQ-Z2 development board in the system. With this card, it is possible to make embedded system applications using Python language and libraries. Moreover, by using the programmable logic, designed IP block can be used as a hardware library in the python environment. Images can be taken from the camera with the HDMI and USB inputs on it. It can also be connected to the Internet with an Ethernet port. Also, the fact that it can be powered via micro USB facilitates the use of the card. At the same time, a warning sound is sent with the 3.5 mm audio output on it [3].

Camera selection is also very important for the success of the system. The camera needs to be of high quality in order to analyze well, and it also needs to present a good image in low-light environments. We used Logitech Brio 4K Webcam in our system to get the driver's image, which is analyzed in the board. It has 4K/30 fps and 1080p/60 fps video capturing capability. The camera performs well in different light conditions thanks to HDR technology. In addition, thanks to the field of view (FOV), which can change from 65 degrees to 90 degrees, we can easily detect the driver in all conditions [4].

The Wi-Fi module and the Ethernet port are used to send the information obtained after the analysis on the card to the cloud. USB Wi-Fi adapters are supported by the PYNQ-Z2 board. We sent information to a cloud system by using the Wi-Fi module.

III. Design and Implementation

The system installed includes the PYNQ-Z2 development board, as we mentioned before. This board is a SoC. It has ARM Cortex A9 dual core processor, FPGA, 512 MB RAM, USB port, HDMI input and output ports, audio output port, Ethernet port and more.

In our design, we are taking images by connecting a Logitech Brio 4K camera [4] via USB. We determine the driver status by analyzing this image with our algorithm that we chose. A simple analysis was done using the OpenCV library with the photos uploaded to the card. The condition of the driver's eyes are determined according to whether OpenCV can find the eyes or not. Also, the success and duration of the operation was measured. The success rate, which is low, increased since we are using a better algorithm to determine the status of driver. In addition, by using IP blocks prepared by PYNQ, both speed advantage and the use of ports on the card were provided.

It is possible to access the card via the Ethernet port or with a program like Putty after connecting to the same internet network. The image from the camera can be taken via USB and HDMI. We get our images by using the USB port. It is also possible to output images with HDMI output. Input and output ports of the PYNQ-Z2 board can be seen in Figure 5.

978-1-6654-8894-5/22/$31.00 ©2022 IEEE

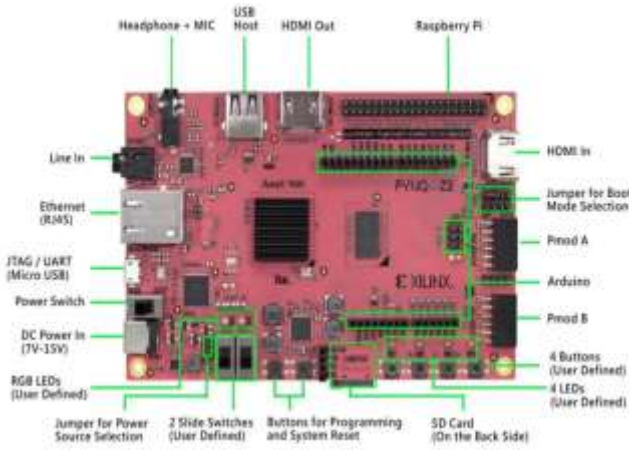


Fig. 5. Inputs and outputs of the PYNQ-Z2 board [16]

We implemented the driver drowsiness analysis to run in real-time on PYNQ-Z2. Although the CNN method was successful, its speed was slow due to its complexity. That is why we continued our work with facial landmarks, haar cascade and EAR methods [20].

The haar cascade files needed to find the face are the 68-point facial landmark files created with the iBug 300W dataset (it contains 600 images) [11], and the .bit file containing the PYNQ-Z2 IP blocks. After that, we installed the OpenCV and Dlib libraries and moved on to the coding part.

The block diagram of the implemented system is shown in Figure 6. First of all, we imported the files we prepared. Then we started to take images using the camera. This image was coming in 1080p. However, since processing on 1080P images takes a long time, we converted this image to 480p.

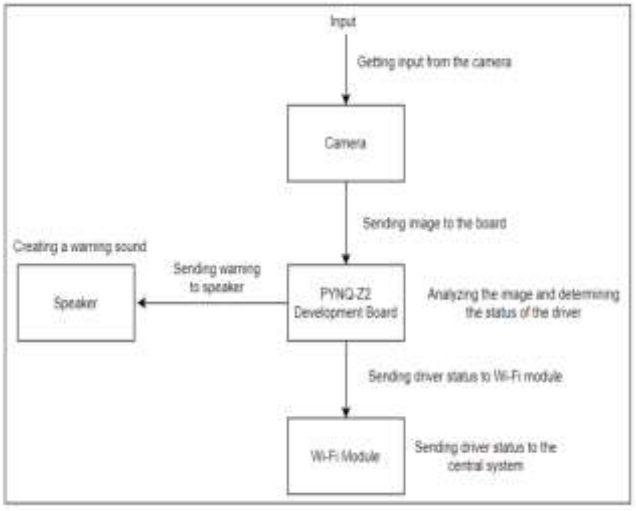


Fig. 6. Block diagram of the system

Then the image is converted to binary. After this stage, we were making face detection using facial landmarks. However, we realized that this slowed down the system and decided to detect the face using haar cascade. Once the face is found, we detect the parts on the face using the facial landmarks method and then convert it to the Numpy array for easy EAR calculation. If there is no face found, which means driver is not looking straight, we also warn the driver. We then obtain the coordinates for the right and left eyes and calculate the EAR for each eye. We take the average EAR of the two eyes.In our trials,We saw that we achieve the best performance when the EAR threshold value is 0.22, and we determined that the eyes are closed if it is below this value. We determine that the driver is drowsy if this value is lower than the threshold value more than 4 times in a row which means eyes are closed for about 3-4 seconds.

Then we move on to the audio warning part. We import the audio block of the PYNQZ2 base overlay into the code. Then we show the path of the audio file and play it with the play command. This sound is played when no face is found or eyes are closed for a long time. Also, the Wi-Fi module is activated. The Wi-Fi function in the PYNQ library is called and the Wi-Fi name and password are entered using the connect function, and the internet is connected without any problems.

Finally, connections are made with the Iothook cloud service. For this, API URL and API key are obtained from Iothook. The API key and driver status are written in JSON format and sent to the cloud service using the request function. It is possible to see the status of the driver every second on the cloud.

IV. RESULTS

The system we prepared by combining facial landmarks, Haar cascade and EAR methods instead of the slower CNN algorithm due to its complexity met our expectation of at least 90% accuracy. As seen in Table 2, the system is more successful than the systems using SVM, Random Forest, Naive Bayes methods in the literature. Only 2% less accuracy is achieved on average than neural network applications. The system can detect driver drowsiness with 92% accuracy.

TABLE II. COMPARISON OF METHODS

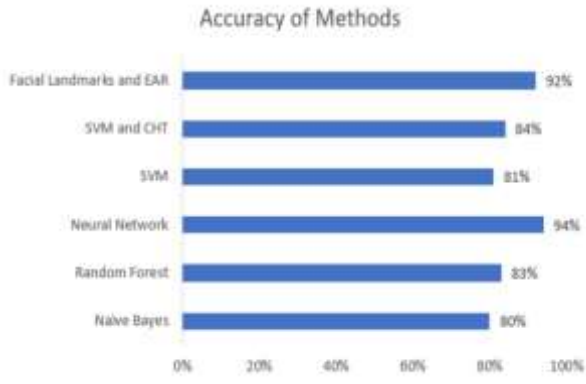


Table 3 shows the performance metrics we have obtained as a result of the improvements we have made in the system. Although we achieved a high accuracy using the first CNN, its speed was well below our expectations. Later, we gave up the accuracy with facial landmark and EAR methods and gained speed advantage. However, this speed was not enough for us. Finally, with the addition of the Haar cascade method, we are able to detect driver drowsiness in 800 ms. In addition, the system consumes only 2W of power. Using a powerbank over 10000 mAh, the system can be used all day long. The only downside of the system is the low performance at night.

978-1-6654-8894-5/22/$31.00 ©2022 IEEE

But this problem can be solved by using a camera with night vision.

TABLE III. RESULTS OF DIFFERENT METHODS

Method	Accuracy	Speed
CNN (MobileNet V2)	%98	>2 s
Facial Landmarks and EAR	%90	2 s
Facial Landmarks, EAR and Haar Cascades	%92	0.8 s

V. CONCLUSION

In this study, we developed the driver drowsiness detection system that has high accuracy and low response time. As a result of our research, we determined that the most suitable solution for us is to make a driver analysis with the camera image since it was budget friendly when compared with other methods. We implemented and compared different processing methods. We first created a system that combines the PERCLOS and CNN methods, where the highest accuracy is obtained. We set up a system on the computer using MobileNetV2 network (usually preferred in mobile devices), which achieves over 98% accuracy. However, this complex system takes more than 2 seconds to detect drowsiness when it is mapped on a low-cost single processor system. Therefore, facial landmarks, Haar cascades and EAR methods are evaluated to be used in this system. Finally, a drowsiness detection and warning system with a 92% accuracy and 0.8 s latency is obtained. Thanks to the base overlay of the PYNQ-Z2, the audible warning was given through a speaker connected to the 3.5mm port. To connect the Wi-Fi function in the PYNQ-Z2 library is used for Wi-Fi adapter and the system is finalized to send warning over the cloud.

ACKNOWLEDGMENT

This work is supported by TÜBİTAK 2209A program, with the project entitled "Yapay Öğrenme Ile Sürücü Yorgunluk Tespit Sistemi Ve Soc Üzerinde Gerçeklenmesi".

REFERENCES

[1] "Drivers are falling asleep behind the wheel," National Safety Council. [Online]. Available: https://www.nsc.org/road/safety-topics/fatigued-driver.

[2] TUL Corporation, "PYNQ-Z2 Reference Manual v1.0," PYNQ-Z2 datasheet, May. 2018.

[3] "Xilinx Pynq-Z2 Overlays," PYNQ. [Online]. Available: https://pynq.readthedocs.io/en/v2.4/pynq_overlays/pynqz2.html.

[4] "Logitech 4K Pro webcam with HDR and rightlight 3," 3. [Online]. Available: https://www.logitech.com/en-roeu/products/webcams/brio-stream-4k-hd-webcam.960- 001194.html.

[5] E. Rogado, J. L. Garcia, R. Barea, L. M. Bergasa, and E. Lopez, "Driver fatigue detection system," 2008 IEEE International Conference on Robotics and Biomimetics, 2009.

[6] W. Horng, C. Chen, Y. Chang, C. Fan, "Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching," 2004 International Conference on Networking, Sensing & Control.

[7] Sałapatek, Damian & Dybała, Jacek & Czapski, Paweł & Skalski, Paweł. (2017). Driver Drowsiness Detection Systems. Proceedings of the Institute of Vehicles. 3. 41-48.

[8] P. Arora, G. Chaudhary, R. G. Crespo, M. Khari, and S. Srivastava, Concepts and real-time applications of Deep Learning. Springer, 2021.

[9] "What is a convolutional neural network?," What is a Convolutional Neural Network? – MATLAB, Simulink. [Online]. Available: https://www.mathworks.com/discovery/convolutionalneural-network-matlab.ht.

[10] Kazemi, Vahid & Sullivan, Josephine. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. 10.13140/2.1.1212.2243.

[11] "Facial Point Annotiations," [Online]. Available: https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/.

[12] Soukupová, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks

[13] Wanjale, K., Bhoomkar, A., Kulkarni, A., & Gosavi, S. (2013). Use Of Haar Cascade Classifier For Face Tracking System In Real Time Video.

[14] P. B. Pankajavalli, V. Vignesh, and G. S. Karthick, "Implementation of haar cascade classifier for vehicle security system based on face authentication using wireless networks," International Conference on Computer Networks and Communication Technologies, pp. 639–648, 2018.

[15] A. Acioglu and E. Ercelebi, "Real Time Eye Detection Algorithm for perclos calculation," 2016 24th Signal Processing and Communication Application Conference (SIU), 2016.

[16] "MRL Eye Dataset," MRL. [Online]. Available: http://mrl.cs.vsb.cz/eyedataset.

[17] "PYNQ-Z2 Development Board," Mouser, 07-Jun-2019. [Online]. Available: https://www.mouser.com/new/dfrobot/dfrobot-pynqz2-dev-board/.

[18] DeepAI, "One hot encoding," DeepAI, 17-May-2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/one-hot-encoding.

[19] S.-H. Tsang, "Review: MOBILENETV2-light weight model (image classification)," 01- Aug-2019. [Online]. Available: https://towardsdatascience.com/review-mobilenetv2-lightweight-model-image-classification-8febb490e61c.

[20] S. Sharma, "Epoch vs Batch Size Vs iterations," 05-Mar-2019. [Online]. Available: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9.

[21] "Base overlay," Python productivity for Zynq (Pynq). [Online]. Available: https://pynq.readthedocs.io/en/v2.7.0/pynq_overlays/pynqz2/pynqz2_base_overlay.html.

[22] Bittrich, Sebastian & Kaden, Marika & Leberecht, Christoph & Kaiser, Florian & Villmann, Thomas & Labudde, Dirk. (2019). Application of an interpretable classification model on Early Folding Residues during protein folding

978-1-6654-8894-5/22/$31.00 ©2022 IEEE