Pedestrian Detection and Avoidance at Night Using Multiple Sensors and Machine Learning

Hovannes Kulhandjian[†], Jeremiah Barron[†], Megan Tamiyasu[†], Mateo Thompson[†], and Michel Kulhandjian[‡]

[†]Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740, U.S.A.
E-mail: {hkulhandjian,jjbarron,mtamiyasu427,mateothompson27}@mail.fresnostate.edu
[‡]Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, U.S.A.
E-mail: michel.kulhandjian@rice.edu

***Abstract*—We develop a machine learning-based pedestrian detection and alert system that can operate both during the day and at night using a visual camera, an infrared camera, and a radar sensor. The visible camera is used to detect pedestrians during the daytime while the infrared camera during the nighttime. Whereas the radar sensor is utilized to detect the presence of pedestrians including their range and directions of motion. We have developed and conducted actual experimentation of the system in a vehicle. We achieved an average accuracy of 98% based on our proposed multi-sensor data analysis using a deep learning algorithm that classifies a pedestrian's presence during the day and at night and alerts the driver in a real-time monitoring system.**

***Index Terms*—Machine learning, pedestrian detection, accident prevention, intelligent transportation systems.**

I. Introduction

On average, a pedestrian is killed every 88 minutes in traffic crashes in the United States. That is more than 16 people a day, almost 115 people a week. The traffic fatalities for the first 9 months of 2020 show that an estimated $28,190$ people died in motor vehicle traffic crashes, which is a 4.5% increase compared to 2019 [1]. In 2019, vehicle accidents in the United States killed more than $6,500$ pedestrians, the highest annual total ever recorded, and sent more than $100,000$ to hospitals with injuries [2], [3]. Additionally, 75% percent of pedestrian fatalities occurred in the dark as compared to daylight (21%), dusk (2%), and dawn (2%). A recent study published by U.S. News & World Report found that roughly 430 Californian pedestrians were tragically killed in the first six months of 2018 [4]. Comparing traffic data from the first half of 2021 to the first half of 2020 reveals that there is a 40% increase in the number of pedestrian deaths due to car crashes [5]. A research study conducted by the Volpe National Transportation Systems Center suggests that automatic emergency braking systems with pedestrian detection functionality could reduce up to $5,000$ annual vehicle/pedestrian crashes and 810 fatal vehicle/pedestrian crashes [6]. Pedestrian detection systems with automatic braking functionality have the potential to prevent or reduce the severity of collisions resulting in property damage, personal injury, and/or death. One possible solution is to use a video camera, a radar system, or a LIDAR (light detection and ranging) system in a vehicle. More recently, the advancement of thermal infrared (IR) cameras has shown a potential possible solution. The research on pedestrian detection and avoidance is still in its infancy. Several methods have been explored to detect a pedestrian and avoid an accident [7]–[14].

Related Work: In [9] Lim *et al.* propose a method for discriminating stationary targets in traffic monitoring radar systems. Their focus is not on pedestrian detection per se. In [10] Cao *et al.* propose a hierarchical reinforcement and imitation learning (H-ReIL) approach that consists of low-level policies learned by imitation learning (IL) for discrete driving modes, and a high-level policy learned by reinforcement learning (RL) that switches between different driving modes to achieve higher efficiency and safety. In [12] Luo *et al.* propose a pedestrian detection scheme based on active night vision operating in the near-infrared (NIR) region of the electromagnetic spectrum, and passive night vision operating in the far-infrared (FIR) spectrum. They also discuss the pros and cons of each type of night-vision system in terms of pedestrian detection capability. Similarly, Han and Song [13] propose a night vision pedestrian detection for automatic emergency braking (AEB) system using near-infrared camera. They use aggregated channel features (ACF) and AdaBoost algorithm to discriminate pedestrians reflected by an infrared camera. Adaptive pre-processing is done to improve contrast images of pedestrians. In [14], Fu presents a pedestrian detection scheme based on the "three-frame difference method" in which they make two improvements using the adjacent frame difference method of the original three-frame difference to "three-frame frame mutual difference". By doing so, it avoids the influence of pedestrian behavior on the system. In addition to that, the authors introduce the "double threshold parallel structure" to transform the image, which relatively increases the useful information contained in the image, and increases the accuracy of the system. However, non of these works conduct actual implementation and employ multi-sensor data fusion combined with machine learning for detecting and alerting the driver in case a pedestrian is detected.

The main contribution of this research work lies in utilizing three different sensors (i.e., a thermal infrared camera, a radar sensor, and a visible camera) combined with advanced machine learning (ML) for pedestrian detection and avoidance. Therefore, we believe that this research exploration could lead to new artificial intelligence (AI)-based application tools for drivers that can save lives. The goal of this research work is to maximize the detection capabilities of pedestrians, especially at night, by effectively making use of the information gathered from a thermal camera, a radar sensor, and a video camera along with the use of advanced machine learning algorithms to detect and avoid pedestrian collisions in real time. Using this multi-dimensional valuable data, it could make intelligent decisions during different conditions of the road, day or night. The proposed system could potentially be embedded into a smart vehicle system that provides real-time pedestrian detection and alerting mechanisms by vibrating the driver's wheel and displaying a message on a monitor/dashboard to warn the driver to avoid colliding with a pedestrian. We have developed and conducted actual experimentation of the system in a vehicle. We achieved an average accuracy of 98% based on our proposed multi-sensor data analysis using a deep learning algorithm that classifies a pedestrian's presence during the day and at night and alerts the driver in a real-time monitoring system.

The rest of the paper is organized as follows. In Section II, we describe the system overview, followed by the pedestrian detection using a video camera, IR camera, and a radar sensor in Sections III, IV, and V, respectively. After illustrating experimentation results in Section VI, we draw the main conclusions in Section VII.

II. System Overview

A. System Design

The system is designed to inform a driver in case a pedestrian is crossing a street and is in danger. Based on the machine learning framework combined with a video camera, an IR camera, and a radar sensor installed on a vehicle, it is capable of making an accurate assessment of road condition and determine if a pedestrian is in the proximity of the vehicle. As an example, the video camera captures red green, and blue (RGB) images whereas the other sensors simultaneously scan the road conditions in front of the vehicle. The two cameras are strategically placed so that they capture a wide view in front of the vehicle. Once a pedestrian in the captured images is detected, the pedestrian images are cropped automatically by running a computer script and are formatted to serve as the input to their respective deep convolutional neural network (DCNN) models that will be used to train them. On the other hand, the radar sensor is used mainly to detect the pedestrian's distance from the vehicle, direction of motion, and speed, respectively.

The DCNN algorithm after being trained using the RGB and the IR cameras images along with the micro-Doppler signal will be able to make predictions corresponding to the driver's behaviors captured in the sample images and the radar signals. The predictions made are then used to detect the presence and distance of a pedestrian. The DCNN algorithms are run on a Raspberry Pi paired with a Coral USB accelerator used to reduce inference times, and real-time processing time. The detections are made continuously over a sample period of sixty seconds. Once each sample period has elapsed, the cumulative detections of the presence of a pedestrian in front of the vehicle using the three different sensors are performed. Those predictions captured are displayed to the driver through the touchscreen display in real time. If the system detects the presence of a pedestrian in front of the vehicle it warns the driver.

B. Deep Convolutional Neural Network Design

A DCNN model is created in this project, as DCNN is most common for image classification [15], [16]. The architecture of a DCNN algorithm implemented for the IR camera image input is shown in Fig. 1. A similar DCNN architecture is used for the RGB image input. The captured video and IR images are first resized to 100×120 pixels RGB images. The input images undergo a feature extraction network by first being processed by the convolution layer consisting of 8 convolution filters of size 20×20. The output from the convolution layer goes through the rectified linear unit (ReLU) function followed by the pooling layer, which employs a max pooling process of 2×2 matrices. This process is repeated several times to create the output and train the machine with the inherent features of the image. The output of the pooling layer is fed into a second convolution layer consisting of 16 convolution filters of size 10×10. Similarly, after passing the output through the ReLU function it undergoes the pooling layer with a max pooling size of 2×2 matrices. Finally, it is passed through the third round of convolution layer consisting of 32 convolution filters of size 5×5 after, which is processed by the ReLU function and the pooling layer with max pooling size of 2×2 matrices.

Since the RGB and IR cameras images contain sharp edges max pooling instead of average pooling is used to extract the most important features such as edges. The classifier network consists of a fully connected layer comprised of 100 hidden nodes, which produce a Softmax output that in turn is used for classifying the presence or absence of a pedestrian. The output layer of the DCNN represents the probability distribution containing the probabilities that each class is assigned in accordance with the input images. Using maximum ratio combining by looking at the three gathered sensor data once the algorithm detects a pedestrian it will send an alert to the driver; if no pedestrian is detected it will forward to the algorithm to continue to perform the pedestrian detection on the new set of data received by the sensors.

III. Pedestrian Detection using Video Camera

A. Data Collection using the RGB Camera

A 4K resolution RGB camera ELP's USB4K02AF-KL100W was used to gather images for the dataset to configure a custom machine learning model and to detect a person with the custom machine learning model running. The gathering of data with this camera was done in the daylight exclusively since the nighttime data is taken care of by the IR camera, discussed in Section IV.

Data were gathered at different distances, several sample images are shown in Fig. 2. Bounding boxes were used on each image with a pedestrian to indicate the exact locations of the pedestrian to help the DCNN algorithm better train on detecting and classifying a pedestrian in an image, as shown in Fig. 2 (right). The number of images gathered and used in the final model is 1200 images, 600 of which are with

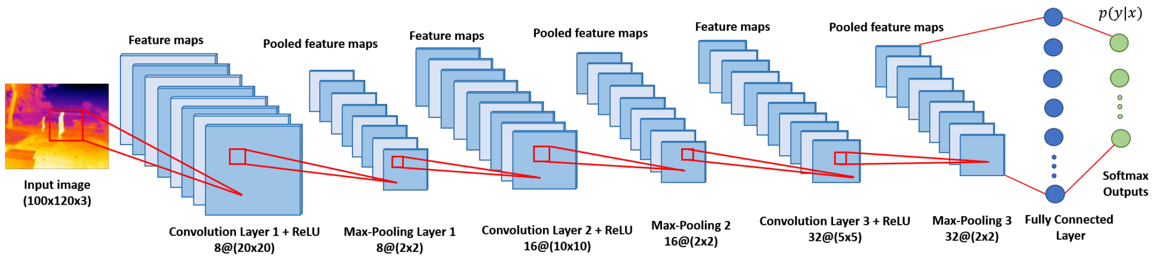

Fig. 1: Convolutional neural network architecture for the IR camera input.

pedestrians and the rest with no pedestrians. Additionally, we have used 800 images from the FLIR dataset [17], a sample image is shown in Fig. 2 (middle), of which 50% was with pedestrians and the rest without. We have used in total of 2000 images to train and experiment with our ML model.

Fig. 2: RGB Sample Images.

The data gathered was used to create a custom machine-learning model. This custom machine learning model used the Squeezenet pre-trained network, as it performed well with RGB images and can run at relatively fast speeds given the hardware used. Analysis of the trained model indicated that the model detects the images with high accuracy, indicated by the accuracy results depicted in Fig. 3.

Standard camera images from the FLIR dataset (Teledyne FLIR Thermal Dataset) were used to differentiate the person images when gathering the accuracy graphs in MATLAB. The training and creation of the custom model were done on a separate computer and then transferred over to a Raspberry Pi 4 as a TensorFlow Lite model. The Raspberry Pi 4 was used to test the custom-trained model in real time.

B. *Performance Results for RGB Camera*

After curating the dataset, the DCNN was trained. A validation accuracy of 99.6% was achieved, as shown in Fig. 3. Examining the validation accuracy, it begins to plateau after nine epochs of training, this indicates that the model has reached its peak performance. The validation accuracy represents the accuracy the model can practically achieve when new samples are input to the model. The loss metric describes how well the model is responding to training after each iteration or epoch. The loss is used to optimize the model so that the next prediction can be more accurate. Ideally, the loss will continue to decrease as training continues.

IV. Pedestrian Detection Using IR Camera

A. *Data Collection using Infrared Camera*

Data was gathered using the FLIR Lepton 3.5 IR camera that is smaller than a dime, as shown in Fig. 4. Several

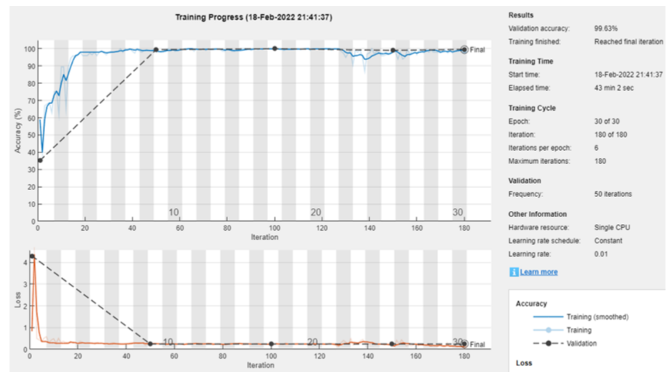

Fig. 3: Validation accuracy and loss plot for the RGB camera.

examples of images captured by the IR camera are shown in Fig. 5.

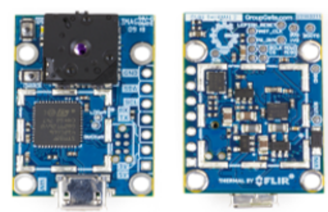

Fig. 4: IR Camera: FLIR Lepton 3.5 with Pure Thermal 2 breakout board.

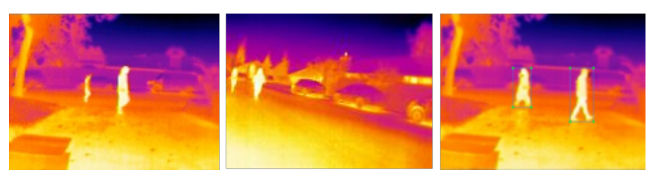

Fig. 5: Sample data images gathered with the IR camera.

Images were then labeled using LabelImg labeling software [18] a graphical image annotation tool written in Python that uses Qt for its graphical user interface to identify persons in the training and testing sets, as shown in Fig. 5 (right). All images were collected from the camera, no datasets were used in this study. Most images were taken at night ($\sim$ 85%) and the rest were taken during the day. After the images were collected the DCNN model was trained using the pre-trained network SSD MobileNet V2 FPNLite 320 $\times$ 320. This network was chosen due to its relatively high speed and the image size of this network is closest to the native resolution of the FLIR camera, so image distortion is kept to a minimum. Other pre-trained networks have been experimented but the results were not as good as using the MobileNet.

B. Performance Results using the IR Camera

After curating the dataset, the deep convolutional neural network was trained. A validation accuracy of 97.26% was achieved, as shown in Fig. 6. Examining the validation accuracy, it begins to plateau after four epochs of training. In the test data, all images with humans within 15 meters of range from the vehicle were detected successfully by the model. The accuracy quickly falls off outside of this range as the pedestrians are beyond 25 meters from the IR camera the algorithm starts to degrade. One way to further improve the accuracy in the future is to use a higher-resolution IR camera.

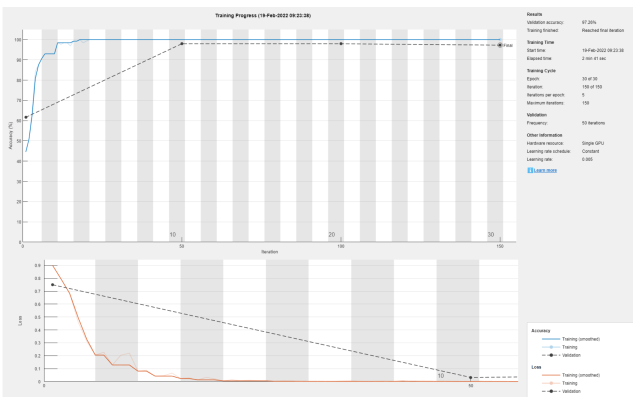

Fig. 6: Validation accuracy and loss plot for IR camera.

V. Pedestrian Detection Using Radar Sensor

A. Micro-Doppler Radar Setup

To capture the radar data OmniPreSense's OPS243-C sensor, shown in Fig. 7 is used, which is a complete short-range radar (SRR) solution providing motion detection, speed, direction, and range reporting.

Fig. 7: OmiPreSence OPS24-C FMCW Doppler sensor.

All radar signal processing is done on board and a simple application programming interface (API) reports the processed data. Flexible control over the reporting format, sample rate, and module power level is provided. The sensor is used to gather pedestrian's speed, the direction of motion, and range from the vehicle. This single-board radar sensor can detect objects up to 60 meters away.

The radar sensor is used to gather data in a form of spectrograms, which can be used as input to the DCNN algorithm to predict the driver's status. Figure 8 represents sample spectrogram plots of no pedestrian detected (left), pedestrian detected at 10 m (middle), and pedestrian detected at 20 m (right). As we can see, Fig. 8 right side has more micro-Doppler frequency variations, since the pedestrian is in closer proximity to the vehicle. When the pedestrian is farther away at 20 m it results in fewer perturbations, as indicated in Fig. 8 (middle).

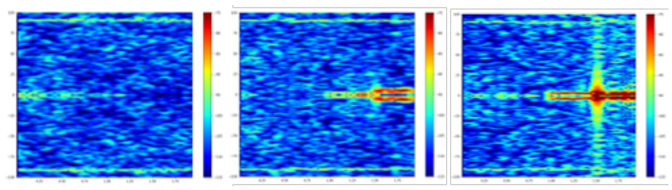

Fig. 8: Sample spectrogram plots for no pedestrian detected (left), pedestrian detected at 20 m (middle), and 10 m (right).

B. Experimental Results Using the Radar Sensor

The OPS243-C FMCW (frequency-modulated continuous wave) radar is capable of not only measuring the speed of moving objects but also, the distance from the Doppler. The FMCW radar uses two antennas, a transmit antenna that sends an FMCW signal and another antenna that receives the echoed signal. The OPS243-C FMCW Doppler was configured through its onboard application programming interface (API). Data was collected and compiled through a Python script, which was executed on a Raspberry Pi 4. This data was then used to detect whether a human has come into proximity to the vehicle. Within this proximity, the RGB and IR cameras will be used to determine the possibility of an accident. Due to the limitations of the micro-Doppler found in testing, the OPS243-C can only pick up a human signature within tens of meters. This hinders the plan for the use of Doppler to generate spectrograms that would be fed into a DCNN used for ML to read the frequency patterns of a car approaching a human. After finding the human-detected range of tens of meters, the Raspberry Pi could only produce measurable spectrograms for an average of 2 seconds. This time difference within a short amount of time does not give the detection system enough time to respond. Instead, when capturing linear data and using the radar as a range finder, range depths were able to be taken instantly with no time delay. This range finder was used to detect an object within the detected human range of about ten to twenty meters of the vehicle. All other objects found outside this range were ignored. If an object was detected within that range, the radar would set a flag high and allow the ML algorithms created for the RGB and IR FLIR cameras to be implemented. For the system to determine whether a possible accident may occur with a pedestrian, the system operates in the following way. All the sensors will be gathering data and writing it to a file to be read. When the radar detects an object, it will set a flag high. Once this flag is set high, the two cameras' detection scores will be calculated with their corresponding weights of 75% and 25% for the IR FLIR and RGB cameras, respectively. Higher weights for IR camera was selected as it can detect pedestrians at night as well as during the day. These detected scores are weighed and evaluated. If the combined detection weights between the IR FLIR and RGB cameras are above 50%, then the system will output a signal to the vibrating motor to alert the driver of the possibility of an accident with a pedestrian.

VI. Prototype Experimentation

In this section, we present the experimentation of the proposed pedestrian detection scheme.

A. Testbed Experimentation in a Vehicle

The pedestrian detection system was deployed in a vehicle, as shown in Fig. 9. The RGB and IR cameras and the radar sensor are all installed on the vehicle. A motor is attached to the steering wheel to alert the driver as soon as it detects a pedestrian that might be in danger. A live demonstration is performed in a vehicle during the day and night times, which is shown in Fig. 10. Overall, the proposed system did perform quite well. On average over 98% accuracy was achieved in detecting pedestrians during the day and night.

Fig. 9: Pedestrian detection system installed on a vehicle.

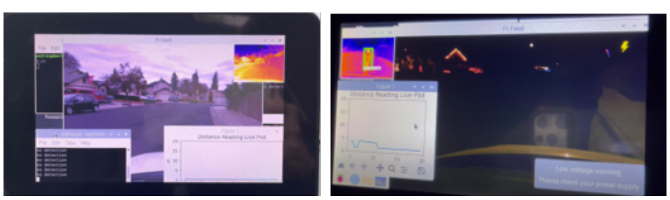

Fig. 10: Live demo during the day and night touchscreen snapshots.

VII. Conclusion

In this paper, a pedestrian detection and avoidance scheme using multi-sensor data collection along with machine learning is implemented and experimented with, which can be used for intelligent transportation systems. The proposed system is composed of a video camera an infrared camera and a micro-Doppler radar for data collection and training. After training the machine learning models the proposed scheme is deployed for real-time experimentation on a vehicle. The system can be used during the day or night. The experimental results reveal the system can achieve an average accuracy of over 98% in detecting pedestrians during the day and at night.

Acknowledgment

Support and funding for this effort was partly provided by the Fresno State Transportation Institute (FSTI).

References

[1] "National Highway Traffic Safety Administration (NHTSA) united states. early estimate of motor vehicle traffic fatalities for the first 9 months (jan–sep) of 2020." https://www.nhtsa.gov/risky-driving/drowsy-driving, accessed: 2022-08-09.

[2] J. Feese, "Highway Safety Association, New Projection: 2019 Pedestrian Fatalities Highest Since 1988. Feb. 2020." https://www.ghsa.org/resources/news-releases/pedestrians20, accessed: 2022-08-09.

[3] E. Edmonds, "AAA Warns Pedestrian Detection Systems Don't Work When Needed Most," https://newsroom.aaa.com/2019/10/aaa-warns-pedestrian-detection-systems-dont-work-when-needed-most/, accessed: 2022-08-09.

[4] G. Galvin, "The 10 Worst States for Pedestrian Traffic Deaths, Feb. 2022." https://www.usnews.com/news/healthiest-communities/slideshows/10-worst-states-for-pedestrian-traffic-deaths?slide=3, accessed: 2022-08-09.

[5] S. Romero, "Pedestrian Deaths Spike in U.S. as Reckless Driving Surges, Feb. 14, 2022." https://www.nytimes.com/2022/02/14/us/pedestrian-deaths-pandemic.html, accessed: 2022-08-09.

[6] M. Yanagisawa, E. Swanson, P. Azeredo, and W. Najm, "Estimation of potential safety benefits for pedestrian crash avoidance systems," 2017.

[7] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 287–302, 2009.

[8] N.-E. E. Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges – a survey," *Information Fusion*, vol. 12, no. 1, pp. 4–10, 2011.

[9] H.-S. Lim, J.-E. Lee, H.-M. Park, and S. Lee, "Stationary target identification in a traffic monitoring radar system," *Applied Sciences*, vol. 10, no. 17, pp. 4–10, 2020.

[10] Z. Cao, E. Bıyık, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh, "Reinforcement learning based control of imitative policies for near-accident driving," Jul. 2020, arXiv:2007.00178 [eess.SP]. [Online]. Available: https://arxiv.org/abs/2007.00178

[11] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A Survey of Vision-Based Traffic Monitoring of Road Intersections," *IEEE Trans. on Intelligent Transportation System*, vol. 17, no. 10, pp. 2681–2698, Apr. 2016.

[12] Y. Luo, J. Remillard, and D. Hoetzer, "Pedestrian detection in near-infrared night vision system," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 51–58.

[13] T. Y. Han and B. C. Song, "Night vision pedestrian detection based on adaptive preprocessing using near infrared camera," in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016, pp. 1–3.

[14] X. Fu, "Pedestrian detection based on improved interface difference and double threshold with parallel structure," in *2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2019, pp. 133–138.

[15] H. Kulhandjian, P. Sharma, M. Kulhandjian, and C. D'Amours, "Sign Language Gesture Recognition using Doppler Radar and Deep Learning," in *Proc. of IEEE GLOBECOM Workshop on Machine Learning for Wireless Communications*, Waikoloa, HI, USA, Dec. 2019, pp. 1–9.

[16] H. Kulhandjian, N. Ramachandran, M. Kulhandjian, and C. D'Amours, "Human Activity Classiffication in Underwater using sonar and Deep Learning," in *Proc. of ACM Intl. Conf. on UnderWater Networks and Systems (WUWNet)*, Atlanta, GA, USA, October 2019.

[17] T. FLIR, "Teledyne FLIR Thermal Dataset for Algorithm Training." https://www.flir.com/oem/adas/adas-dataset-form/, accessed: 2022-08-09.

[18] Tzutalin, "(2015) Labelimg. git code." https://github.com/tzutalin/labelImg, accessed: 2022-08-09.