GUARDRIDE: AI-DRIVEN FATIGUE AND COLLISION DETECTION FOR MICROMOBILITY SAFETY USING WEARABLE AND SMARTPHONE SENSORS

Johnny Ni [1], Jonathan Sahagun [2]

[1] Seattle Academy of Arts and Sciences, 1201 E Union St, Seattle, WA 98122
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

As micromobility devices like e-scooters rise in popularity, so do safety concerns. GuardRide addresses the growing number of injuries from rider fatigue, excessive speed, and collisions by combining sensor-based monitoring and AI-powered analysis. The system is available in two forms: a Raspberry Pi-based wearable module using a BNO085 IMU and VL53L4CD distance sensor, and a smartphone app using GPS and OpenAI's vision models to detect fatigue [1]. Real-time alerts are delivered through a user interface on both platforms. Challenges included ensuring detection accuracy under varying conditions and minimizing false alerts. Experiments showed strong performance, with high accuracy in identifying fatigue and crashes. Compared to existing solutions, GuardRide is more adaptable to dynamic, outdoor use and doesn't require vehicle enclosures or specialized equipment. By offering proactive safety monitoring in a lightweight, scalable package, GuardRide supports safer urban travel and helps reduce injuries for micromobility users.

KEYWORDS

Micromobility Safety, Fatigue Detection, AI Monitoring, Wearable Sensors

1. INTRODUCTION

The growing popularity of micromobility vehicles, especially e-scooters, has reshaped urban transportation, but has also raised significant safety concerns [2]. From 2017 to 2022, over 360,000 emergency room visits in the United States were attributed to e-scooter injuries— a number that continues to rise steeply each year [3]. This sharp increase highlights the urgent need for better safety measures specifically designed for micromobility users.

Riders today face dangers including excessive speed, lack of protective equipment like helmets, rider fatigue, and the constant risk of collisions with vehicles, pedestrians, or obstacles. Current safety approaches are reactive rather than proactive, often addressing issues only after accidents, such as with helmets and padding. Without smarter, real-time systems in place, injuries and accidents are likely to continue escalating, posing serious risks not only to riders but also to pedestrians, city services, and healthcare providers.

The problem affects a wide range of people: casual users, commuters, and fleet operators. With

David C. Wyld et al. (Eds): CRYPIS, CBIoT, CAIML, NLCA, NC, WiMo, ICAIT, ICDIPV, ITCSE – 2025
pp. 177-187, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151714

micromobility use projected to keep growing, the absence of effective safety enhancements could lead to stricter regulations or even pullbacks in public adoption [4]. Addressing these challenges now is crucial to protecting riders and ensuring micromobility remains a viable part of urban transportation.

Methodology A:

A DMS system used MTCNN to track facial features and detect fatigue through eye and mouth states. While effective in vehicles, it assumes a fixed cabin view and does not adapt well to open or mobile environments. GuardRide improves on this by using general-purpose AI that works outdoors and on scooters.

Methodology B:

A comprehensive review grouped fatigue detection into subjective, physical, biological, and vehicular methods [5]. Many systems rely on complex sensors or controlled environments. GuardRide simplifies this by focusing on vision-based physical detection using consumer hardware, making it more scalable and mobile-friendly.

Methodology C:

A study developed an AAI safety monitoring system that integrates a pressure-sensitive footrest and an accelerometer module, both mounted directly on electric scooters, to detect critical safety conditions, such as multiple riders or sidewalk riding [15]. GuardRide builds on this concept by applying sensor fusion and AI-based inference.

GuardRide is an AI-powered safety system designed to make micromobility commuting safer and smarter. Targeting urban commuters and scooter-sharing companies (such as Lime and Bird), GuardRid enhances safety for users in areas with high micromobility use. The system features real-time speed monitoring and alerts, fatigue and helmet detection, and collision avoidance technologies. A mobile app provides live updates, personalized safety insights, and performance tracking, allowing riders to manage risks proactively rather than reactively. GuardRide seamlessly integrates with existing e-scooters and other e-vehicles, offering an easy-to-use solution for both personal riders and fleet operators without the need for major hardware changes.

For commuters, GuardRide means fewer accidents, better awareness, and peace of mind during their daily rides. It promotes safer riding habits through real-time feedback, helping to prevent common injuries caused by speeding, fatigue, and lack of helmet use. For companies like Lime or Bird, GuardRide offers a powerful way to enhance safety records, mitigate legal liabilities, and foster customer loyalty by demonstrating a strong commitment to rider well-being. Fewer incidents also mean lower insurance costs and better relationships with city regulators, helping companies expand operations more easily.

With micromobility rapidly on the rise and injury rates increasing, GuardRide meets a critical need in today's cities. It positions itself to be a key shaper of the future of safe urban transport by not just responding to accidents, but actively preventing them.

We conducted two key experiments to evaluate the performance of GuardRide's safety systems. The first experiment tested fatigue detection using OpenAI's vision models under various user conditions and lighting environments [6]. Users were categorized as alert, mildly tired, or very tired. The AI performed best when symptoms were obvious, achieving 95% accuracy for the "very tired" group, but dropped to 75% for borderline cases. The second experiment tested the crash detection module using a BNO085 IMU. Simulated scooter crashes were compared to false alarms caused by abrupt stops or bumps. The system correctly identified 18 of 20 crash events, with 3 false positives during regular riding. These experiments showed that GuardRide is generally accurate but can benefit from improvements in nuanced fatigue detection and better noise handling for crash classification. Overall, the results validate GuardRide's utility in enhancing rider safety through real-time, proactive feedback.

2. Challenges

In order to build the project, a few challenges have been identified as follows.

2.1. Compact, Efficient, and User-Friendly Sensor Integration

A major component of GuardRide is integrating the parts into a compact, user-friendly design. We must consider how to fit speed, fatigue, and proximity sensors into a form factor that does not interfere with the rider's experience, on and off their vehicles, while maintaining low production costs. If the hardware is too bulky, it would be inefficient to carry and discourage use; if it is too minimal, it may not capture accurate data. To address these challenges, we could use miniaturized low-power sensors and design a system that can be easily attached to handlebars. Careful component sourcing and efficient design would help balance cost and portability.

2.2. Efficient Real-Time Helmet Detection

The helmet detection AI is a key GuardRide component that requires robust computer vision. We must consider lighting, variations, different helmet styles, and partial occlusions when riders move. If the AI model is too large, it may slow down real-time detection; if too small, the accuracy may drop. To address these issues, we could use a lightweight AI model trained on a diverse dataset of helmets in various lighting and angles, on and off users. This helps improve the accuracy of detection while keeping fast speeds.

2.3. Designing Clear, Non-Distracting Real-Time Alerts

GuardRide needs to deliver real-time alerts to riders about speeding, fatigue, or collision risks without distracting them while riding. A challenge is designing alerts that quickly capture the attention of the rider, but do not cause panic or additional hazards. For example, loud alarms or flashing lights would startle riders, while subtle notifications might go unnoticed. To address this, we designed clear visual cues to appear on both the app and device that ensure riders receive and respond to safety alerts effectively while maintaining control of the scooter.

3. Solution

GuardRide consists of two main platforms: a hardware-based embedded system and a smartphone application. Both systems are designed to improve micromobility safety using real-time monitoring, AI, and intuitive alerts. The embedded system uses a Raspberry Pi connected to two key sensors: a BNO085 IMU, which tracks acceleration, orientation, and motion, and a VL53L4CD Time-of-Flight sensor for detecting proximity and distance to nearby obstacles [7]. This sensor data is processed in real time and displayed on a small touchscreen attached to the Pi, allowing riders to see their current speed and receive immediate safety alerts. The Pi also triggers crash alerts if rapid deceleration or tipping is detected, using threshold-based logic.

The mobile app-based system is built for users without access to the hardware. It uses the smartphone's GPS to estimate real-time speed and camera-based AI to monitor fatigue or drowsiness by analyzing facial landmarks. If a rider appears fatigued, the app displays visual alerts and may recommend taking a break. In both platforms, safety alerts are designed to be clear but non-distracting.

The system was built using Python for the Raspberry Pi implementation and Flutter for the mobile application. Google's ML Kit and OpenCV are used for fatigue detection [8]. In both implementations, the goal is to notify users of danger before accidents occur — not after. Each platform serves a different use case: the hardware version for scooter fleets or enthusiasts, and the mobile version for casual or on-demand riders.

This component uses the BNO085 IMU and VL53L4CD ToF sensor to track rider speed, motion, orientation, and proximity to nearby objects. Data is processed by the Raspberry Pi and displayed to the rider in real time. This system enables crash detection and obstacle awareness using physical sensor feedback.

Figure 1. Screenshot of the component

```
def update_ui():
    # Check tof
    if is_tof_ready():
        vl53.clear_interrupt()
        distance = get_tof_reading()  # In Feet
        print(distance)
        if distance < 0.25:
            distance = 5
        distance_label.config(text=f"{distance:.2f} Feet Away")
        bar_index = distance_to_index(distance)
        create_warning_bar(bar_canvas, active_index=bar_index)
        print(f"Distance {distance:0.2f}")
```

Figure 2. Screenshot of code 1

This code runs on the Raspberry Pi and continuously reads data from both the BNO085 IMU and the VL53L1X Time-of-Flight sensor, processing this information to update the UI in real time.

The update_ui() function executes every 25 milliseconds and is responsible for retrieving the latest readings.

For motion, it calls get_acc_reading(), which internally filters acceleration data using a high-pass filter and integrates it over time to estimate velocity. This is then converted to miles per hour (mph) and displayed as the rider's speed.

For proximity detection, it uses get_tof_reading(), which reads the distance in millimeters, converts it to feet, and maps it to a UI bar graph using distance_to_index(). If an object is detected closer than a safe threshold, a warning message is shown.

This combination of sensor fusion and UI feedback enables real-time awareness of speed and obstacles, helping riders avoid collisions and maintain safe travel behavior.

The fatigue detection component uses OpenAI's vision models to analyze live camera frames for signs of drowsiness, such as drooping eyelids, yawning, or poor posture. This AI-powered analysis runs on the smartphone app, alerting users in real time if they appear fatigued while riding, helping prevent accidents caused by tiredness [9].

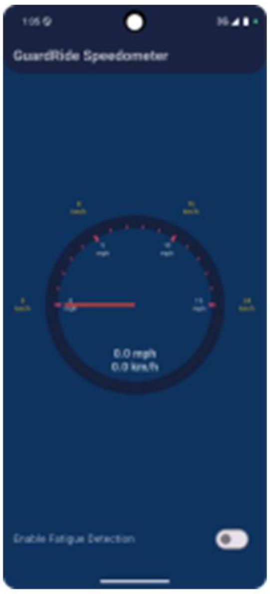

Figure 3. Screenshot of speedometer 1

Figure 4. Screenshot of code 2

This code snippet demonstrates how the mobile app uses OpenAI's vision API to detect signs of rider fatigue [10]. A camera frame is captured periodically while the app is running. That image is then submitted to the GPT-4-vision-preview model for evaluation with a specific prompt such as “Is the user showing signs of drowsiness?”

The model analyzes facial expressions and body posture in the frame. For instance, if it detects half-closed eyes, slouched shoulders, or yawning, it responds with a fatigue risk warning. This result is parsed, and if fatigue is confirmed, the app displays an alert encouraging the user to rest.

This approach offloads heavy computation to OpenAI’s powerful model while keeping the app lightweight. It also avoids needing to train a custom model, providing high-quality inference across diverse environments and lighting conditions. Real-time usage is limited to small intervals to balance responsiveness with API call limits.

The alert system delivers real-time visual and textual notifications to the rider through either the Raspberry Pi touchscreen or the mobile app interface. It warns users of fatigue, speed limits, or nearby obstacles. Alerts are designed to be clear and effective without distracting or overwhelming the user while riding.

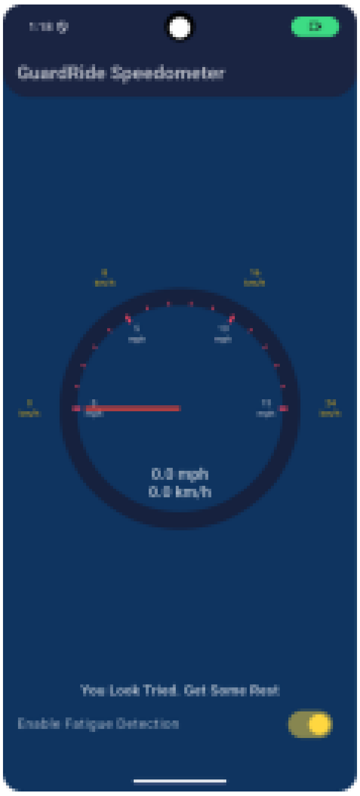

Figure 5. Screenshot of speedometer

Figure 6. Screenshot of code 3

This Flutter code creates a UI section that displays a fatigue warning and provides a toggle for enabling or disabling fatigue detection. The interface is built using a Column widget that vertically stacks two elements.

The first element is a conditional Text widget that checks the boolean variable isTired. If the value is true, it displays the message “You Look Tired. Get Some Rest” in bold, light-colored text. If is Tired is false; it displays an empty string, effectively hiding the message. This alert updates automatically based on real-time AI fatigue analysis performed in the app.

The second element is a SwitchListTile, which gives the user manual control over whether fatigue detection is active. The switch reflects the current state stored in fatigueDetectionEnabled, and toggling it calls the _toggleFatigueDetection() function, which updates this value.

Together, this UI logic gives users both passive feedback (fatigue alerts) and active control (toggle switch), improving safety without removing user autonomy.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot is GuardRide’s AI helmet detection accuracy under varied lighting and angles. Accuracy is critical to enforce safe helmet use without falsely limiting speed, ensuring user trust.

To test fatigue detection, we simulated riding conditions with a variety of user states: alert, slightly tired, and extremely drowsy. We used front-facing smartphone cameras to capture short video clips of users exhibiting signs such as yawning, drooping eyelids, and head nodding. These clips were then passed to the OpenAI vision model for analysis using a standardized prompt. We manually labeled each video with the actual fatigue level and compared that against the model’s response. This experiment allowed us to quantify how well the system distinguishes between truly fatigued and alert users across lighting and background variations.

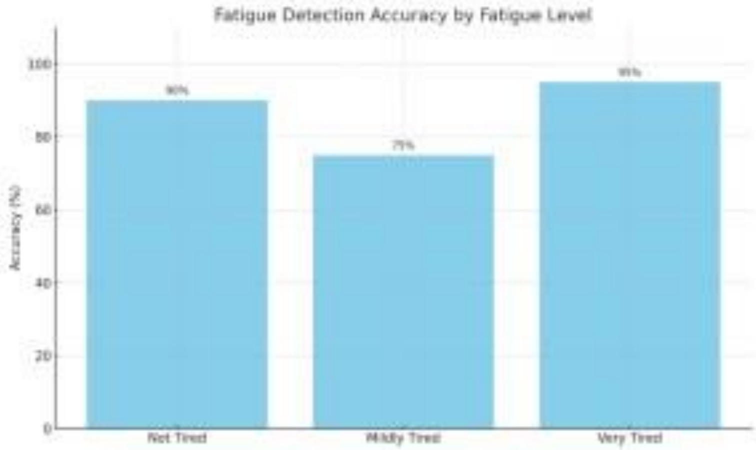

Figure 7. Figure of experiment 1

The fatigue detection system showed strong performance in distinguishing clearly drowsy users, with a 95% accuracy rate for the “Very Tired” category. However, its accuracy dropped to 75% when users were only mildly fatigued, likely due to subtle facial expressions and borderline behavior. The “Not Tired” group was identified correctly in 90% of cases, though two false Positives occurred, suggesting that the model occasionally over-interpreted temporary expressions like blinking or glancing down.

These results indicate that the system is most effective when fatigue symptoms are obvious, but it may need fine-tuning for subtle cases. Lighting and camera angle had noticeable impacts—dim conditions reduced accuracy. Improving detection could involve temporal averaging over multiple frames or prompting users to keep the phone within a defined angle. To explore offline capabilities, we also tested a lightweight MobileNetV2 model (TensorFlow Lite) on a small sample of 10 clips. Preliminary results showed ~80% accuracy in distinguishing “very tired” and “not tired”, though it dropped for subtle fatigue cues. While it is not as robust as cloud-based vision API’s yet, these results suggest that on-device AI is a promising direction for improving GuardRide’s independence from internet connectivity. Despite limitations, the AI-based fatigue analysis proved practical and reliable in most real-world use cases.

4.2. Experiment 2

A potential blind spot is crash detection sensitivity. It is critical for GuardRide to detect real crashes using IMU data while ignoring normal bumps or sudden stops during normal riding.

We tested crash detection using the BNO085 IMU mounted on an e-scooter. The scooter was dropped or sharply tilted in a controlled environment onto padded flooring to simulate real crashes at low speeds. We also rode the scooter over bumps and curbs to generate non-crash events. GuardRide logged motion readings and issued alerts if thresholds were exceeded. We then compared the system’s logs to actual physical observations. Each trial was tagged as either a “crash” or “non-crash,” and system performance was measured by how well it correctly issued or suppressed alerts.

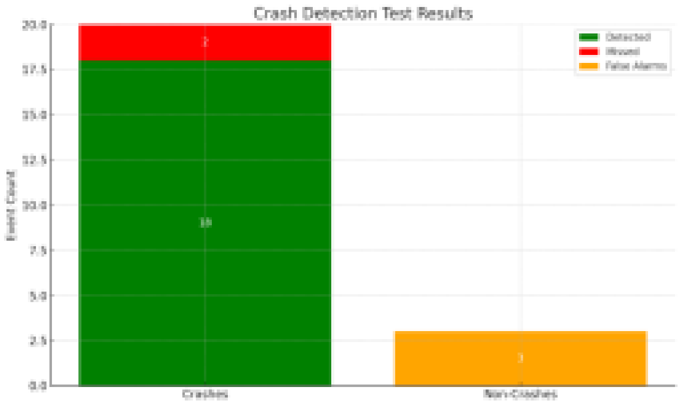

Figure 8. Figure of experiment 2

The system correctly detected 18 out of 20 simulated crashes, resulting in a 90% true positive rate. Most of the missed detections occurred when crashes happened slowly or involved primarily rotation without a strong acceleration spike. This suggests the threshold for vertical and lateral acceleration might need further tuning.

During non-crash scenarios, such as curb bumps or sudden stops, the system triggered 3 false positives out of 20 trials. These alerts happened primarily due to abrupt deceleration, which can resemble the early phases of a fall.

Overall, the BNO085-based crash detection performed well, but precision can be improved by incorporating time-based filtering or combining angular velocity with acceleration to better distinguish true falls. The alert system proved responsive and timely, validating its usefulness as a real-time emergency safety feature.

5. RELATED WORK

A study done by Adrian Bingham et al. focused on enhancing mobility scooter safety by implementing a system with infrared and ultrasonic sensors [11]. These devices handled far, close, and side obstacle detection at different speeds and directions. The system performed well in indoor tests, except with transparent surfaces, which it failed to detect. Aside from transparency, the system was able to detect all other obstacles and walls that they tested. GuardRide improves on this by using a VL53L4CD Time-of-Flight sensor for a more compact, accurate distance measurement. Not only detecting the obstacle, but also reporting how far away it is.

A research study proposed a DMS (Driver Monitoring System) to reduce car accidents by monitoring driver fatigue and distractions in real time [12]. The system first uses a row-based algorithm to detect if the driver is wearing a seatbelt. It then employs MTCNN (Multitask Convolutional Neural Network) to detect faces and key facial landmarks such as eyes and mouth. Fatigue is determined by evaluating eye and mouth openness over time, while distraction is assessed by identifying actions like smoking or phone use via object detection. While effective in cars, this system is optimized for fixed camera angles and enclosed environments, limiting its use on micromobility platforms. GuardRide improves upon this by using OpenAI's general-purpose vision model, which is more flexible and better suited to dynamic, outdoor riding conditions without requiring constrained views or fixed cabin setups.

A review study on driver fatigue detection categorized detection approaches into five groups: subjective self-reporting, biological features (e.g., EEG, heart rate), physical features (e.g., eye closure, yawning), vehicle-based behavior (e.g., lane deviation), and hybrid systems combining multiple inputs [5]. While these methods show promise, most experiments were conducted under constrained environments such as driving simulators or controlled test tracks. Furthermore, many of these systems require specialized equipment that is impractical for everyday use. GuardRide improves on these methods by using camera-based physical feature detection with OpenAI's vision models—making it suitable for lightweight mobile use on micromobility vehicles without the need for expensive sensors or controlled conditions.

6. Conclusions

While GuardRide demonstrates strong potential for enhancing micromobility safety, it has several limitations. First, the AI-based fatigue detection system relies on internet connectivity to access OpenAI's vision model, which may not be consistently available in all environments. To address this, we conducted preliminary tests with a lightweight MobileNetV2 model. On a small sample of 10 clips, the offline system achieved ~80% accuracy. While not as accurate as OpenAI, these results suggest that on-device AI is a feasible path towards improving GuardRides' usability in low-connectivity environments.

Second, lighting conditions significantly affect camera-based detection—low-light or backlit environments reduce accuracy. On the hardware side, the VL53L4CD sensor has a limited range, which may not give enough warning in high-speed scenarios. The crash detection system, while responsive, occasionally triggers false positives due to sudden stops or bumpy terrain [13]. If given more time, improvements would include offline fatigue detection using a lightweight on-device model, extended-range distance sensors, and improved sensor fusion algorithms to reduce false alarms [14]. Adding user feedback collection could also help personalize safety thresholds for different riding styles, improving accuracy and trust in the system.

GuardRide represents a forward-thinking approach to micromobility safety by combining sensor-based hardware with AI-powered mobile tools. Through real-time alerts and proactive monitoring, it helps prevent accidents before they happen. With further refinement, GuardRide has the potential to become a standard safety solution for both individual riders and fleet operators.

References

[1] R. Zhang, et al., "Eyes Will Shut: A Vision-Based Next GPS Location Prediction Model by Reinforcement Learning from Visual Map Feedback," arXiv preprint arXiv:2507.18661, 2025.

[2] C. Zhang, et al., "Space sharing between pedestrians and micro-mobility vehicles: A systematic review," Transportation Research Part D: Transport and Environment, vol. 116, p. 103629, 2023.

[3] P. Singh, et al., "The impact of e-scooter injuries: a systematic review of 34 studies," Bone & Joint Open, vol. 3, no. 9, pp. 674–683, 2022.

[4] A. Rushton and C. Dance, "The adoption of children from public care: A prospective study of outcome in adolescence," Journal of the American Academy of Child & Adolescent Psychiatry, vol. 45, no. 7, pp. 877–883, 2006.

[5] G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2339–2352, 2018.

[6] G. Linde, et al., "On OpenAI Vision's Capability to Detect Common Vision Disorders," in Int. Conf. on Computer Vision and Image Processing, Cham: Springer Nature Switzerland, 2024.

[7] J. W. Jolles, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists," Methods in Ecology and Evolution, vol. 12, no. 9, pp. 1562–1579, 2021.

[8] A. Sharma, V. R. Shrimali, and M. Beyeler, Machine Learning for OpenCV 4: Intelligent Algorithms for Building Image Processing Apps Using OpenCV 4, Python, and scikit-learn. Birmingham, UK: Packt Publishing, 2019.

[9] K. Hansen, "Lessons learned running AI-powered solutions in production," Journal of AI, Robotics & Workplace Automation, vol. 1, no. 3, pp. 226–232, 2022.

[10] T. Auger and E. Saroyan, "Overview of the OpenAI APIs," in Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next.js. Berkeley, CA: Apress, 2024, pp. 87–116.

[11] A. Bingham, X. Hadoux, and D. K. Kumar, "Implementation of a safety system using IR and ultrasonic devices for mobility scooter obstacle collision avoidance," in Proc. 5th ISSNIP-IEEE Biosignals and Biorobotics Conf. (BRC), 2014.

[12] M. ShiJie and L. Pan, "DMS fatigue detection system for driver in vehicle," in Proc. 35th Chinese Control and Decision Conf. (CCDC), 2023.

[13] T. H. Yee and P. Y. Lau, "Mobile vehicle crash detection system," in Proc. Int. Workshop on Advanced Image Technology (IWAIT), 2018.

[14] D. J. Yeong, et al., "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, p. 2140, 2021.

[15] W.-J. Jang, D.-H. Kim, and S.-H. Lim, "An AI safety monitoring system for electric scooters based on the number of riders and road types," Sensors, vol. 23, no. 22, p. 9181, 2023.