HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 3 (2024)

Third Eye: Wearable ML Device Using CNN and Advanced Distance Estimation for Enhanced Cyclist Road Safety

Dohyun Yang

Cupertino High School, USA

ABSTRACT

Annually, over 1,000 cyclists lose their lives, and 130,000 sustain injuries in the US due to insufficient awareness and collisions with vehicles. Every few weeks there's always news of someone being killed or hurt in crashes with cars. According to the National Highway Traffic Safety Administration, an enormous 82.3% of fatal collisions between vehicles and cyclists occur with the point of impact at the front of the vehicle, indicating that collisions frequently occur because vehicles approach from behind and strike the cyclist. Current existing solutions, such as handlebar mirrors or bike sensors, do not provide adequate situational awareness for the biker, since fatal encounters between vehicles and cyclists continue to occur frequently. To help mitigate this problem, this study introduces Third Eye, a novel safety innovation aimed to address the substantial risks encountered by cyclists on urban roads. Using machine learning on a lightweight computing system, Third Eye provides cyclists with a reliable rearview warning system. Third Eye's distance algorithm generates audio warnings, promptly alerting cyclists to approaching dangers from the rear. With a primary emphasis on accessibility and functionality, Third Eye represents a significant advancement in bike safety technology, potentially able to revolutionize the cycling experience and potentially save lives on a global scale.

Introduction

Project Origin

As lifelong bikers, we noticed the physical danger cyclists face when navigating urban roads. Living in the Bay Area, we've heard many stories from family and friends about collisions and injuries with cars. Additionally, as carbon emissions are on the rise and the global environmental crisis continues to worsen, using bicycles as a means of transportation rather than motor vehicles should not entail risk to the cyclist. As students on our robotics team, we've had experience with Camera Vision (CV) and its implementations within an educational setting, but we decided that it could be further utilized in real life scenarios. Our work on this project began in May of 2023.

Engineering Goal

Problem: Bikers face significant safety concerns due to insufficient rear view visibility, leading to potential collisions with vehicles.

Goal: To improve cyclist rear view visibility by developing a wearable device that uses machine learning and Computer Vision to enhance situational awareness for cyclists and to reduce risk of collisions.

Current Solutions

Vehicle LIDAR: Many car brands (Mercedes-Benz, Honda, BMW, etc.) utilize LIDAR sensors for their self-driving vehicles for situational awareness on the road. We assumed that by researching advanced detection systems used by cars, we would be able to create our own system for cyclists. However, upon realizing the computational intensity to use a LIDAR sensor on a portable computing system such as a Raspberry Pi, we decided LIDAR and similar heavy sensors would not fit our use case due to the bulky application of such sensors for cyclists, who require a lightweight and mobile implementation. Ultimately we resorted to CV due to our previous experience and its convenient application to our use case.

Tesla Vision: Tesla vehicles rely purely on cameras to detect the surroundings. Instead of heavy sensors with high running costs, Tesla cars use wide-view cameras and machine learning algorithms to constantly scan and detect vehicles. We were inspired by Tesla's purely optics-based detection system and realized using cameras instead of LIDAR or similar sensors better fit our use case, as they provide accessible and lightweight implementation.

TF SSD MobileNet V2 FPNLite: Developed by Google researchers M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, MobileNet v2 is a CNN (Convolutional Neural Network) based on the TensorFlow framework, specifically designed for lightweight model training on mobile devices. CNNs are feed-forward neural networks commonly used for time series, audio, and visual applications. We chose this model in particular because it is one of the least computationally demanding models. MobileNet uses Single Shot Detection (SSD), which means it only needs to run a single CNN to return the bounding box. SSD models are extremely efficient, relying on a single CNN, but is slightly less accurate than heavier models. This model in particular provided the best results using the least computing power, thus fitting well with our use case.

Competition: We face direct competition from two sources: traditional bike mirrors, and the Garmin Varia. Mirrors, directly fixed to the handlebar, offer a restricted view due to their small size. Bikers also cannot adjust the handlebar to see past the initial frame, forcing cyclists to turn their head and body, distracting riders from the road. Finally, vibrations and glare from sunlight or headlights are further disadvantages to mirrors, limiting their effectiveness. The Garmin Varia is an advanced vehicle detection system that utilizes infrared beams and provides a reliable alternative. Its main drawback is that they are generally catered and marketed towards high-end and racing cyclists. To use the Varia, users must also invest in a separate biking computer called the Garmin Edge. On average, the cost to purchase all necessary components is well over $500. Meanwhile, Third Eye provides a reliable detection system, actively searching for vehicles. It also stands at a more accessible price point for casual bikers, at around $150.

Methods

Development Timeline

Phase 1: Conduct background research on pedal-cyclist injuries and deaths, with a focus on understanding the causes and underlying factors contributing to such incidents. This involves reviewing existing literature, analyzing statistical data, and consulting with experts in the field and bikers.

Phase 2: Develop and train the object detection model, including gathering training and testing images, labeling, and optimizing the final model and tuning parameters. Begin initial testing and performance cuts. Implement single-camera distance estimation.

Phase 3: Continue refinement and optimization of the model. Implement image stabilization, model quantization, and image segmentation for enhanced stability and performance. Maintain a 70Hz refresh rate and 95% model mAP.

Prototyping & Testing

Although we were never able to use our design with a bicycle on a real street with moving cars because of safety concerns from our school, we thoroughly tested each segment of our software with simulations.

Third Eye's software can be broken down into three main segments: scanning and detecting objects, estimating distance, and then emitting an audio warning. To detect objects, we created an object detection model using the SSD MobileNet, with over 250 training images and 10,000 training steps. To detect objects, the image captured from the camera is scanned for a match to what it is trained as a car. Once detected, the exact coordinates and dimensions of the car are sent to our distance estimation algorithm. The algorithm, also split into two segments, takes into account the width and height of the car, the angle the camera is facing, the position of the car relative to the video frame, and other factors to provide a decently accurate estimation of the distance from the biker to the car. Using this estimation, as well as other calculations to estimate speed and direction, our algorithm determines if a warning is to be issued. Overall, our system is highly refined and optimized for the most accurate detections and appropriate warnings.

Distance Estimation

Our distance estimation algorithm is divided into two segments (figure 1). While vehicles are relatively far away (greater than 6 meters), we estimate distance purely by analyzing bounding box dimensions. Although it provided consistent overestimations, bounding box estimation required very little computing power, which was crucial to reduce computing time and refresh rate. Once a vehicle enters a threshold (pixel width is greater than 100 px), we use triangulation to estimate the distance. Using trigonometry, we can calculate the distance $X = H / \tan(\theta)$. H is a constant (height of the bicycle seat, ≈1m) and θ is a constant (within 100px, θ ≈-10°).

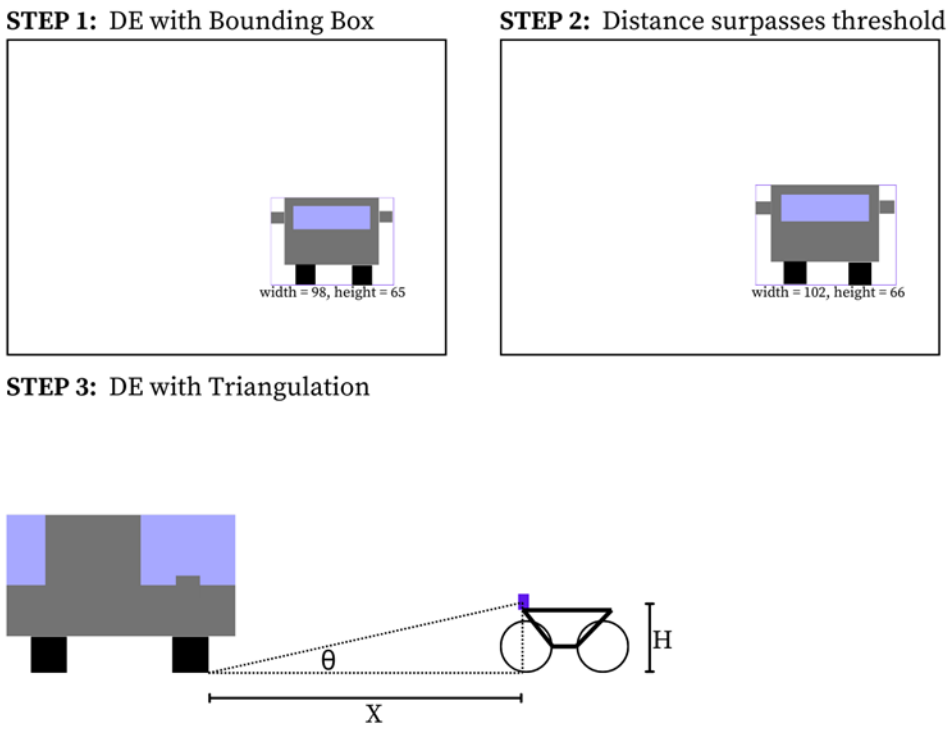

Figure 1. Step 1 & 2: Vehicle is further than 6 meters, pixel width is less than 100 px. Step 3: Triangulation algorithm.

Object Detection Model

To capture training images, we set up an iPhone mounted on a tripod (figure 2) recording a video of cars passing on the street. We had multiple recording sessions at different times of day, camera angles, and locations to have a varied dataset from which the model could be trained. Once a sufficient number of images had been collected, we used Roboflow's image labeling software to construct bounding boxes. To reduce labeling and training time, we scaled the image down to 320x320 pixels.

We also only labeled images of cars heading towards the camera, and did not label those driving in the direction away from the camera. This was because we are only interested in information about vehicles driving behind and approaching the cyclist. Those driving away pose no significant threat to the cyclist. We were able to significantly reduce labeling and training time with this limitation.

Figure 2. Camera mounted on a tripod 1.5m off the pavement. This setup allowed us to capture testing footage for our model and simulate the effectiveness at an average biker's height.

We visually tested our model with a simple setup (figure 3). We set up a large monitor playing a video of cars passing by on a street. We recorded this video from a street sidewalk. Directly facing and perpendicular to the screen, we placed a USB camera directly connected to the Raspberry Pi running our model. The Raspberry Pi, running RealVNC to screencast onto a laptop computer, displayed the processed image data with an overlaid bounding box (figure 4).

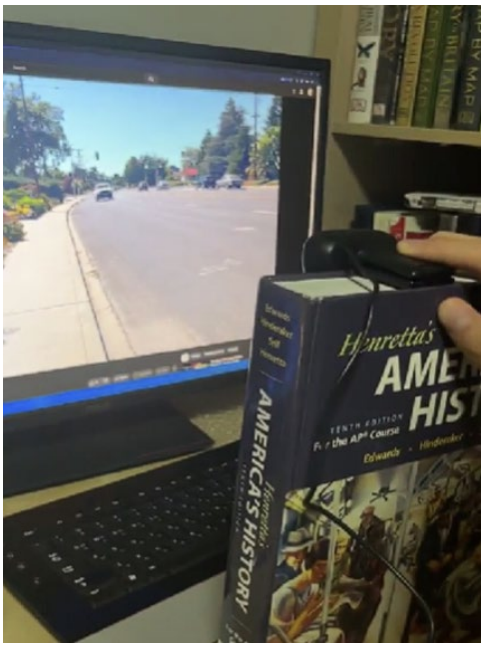

Figure 3. Model testing setup. A secondary USB camera is connected to a separate computer running the model. The camera is centered, but not fixed to test stability.

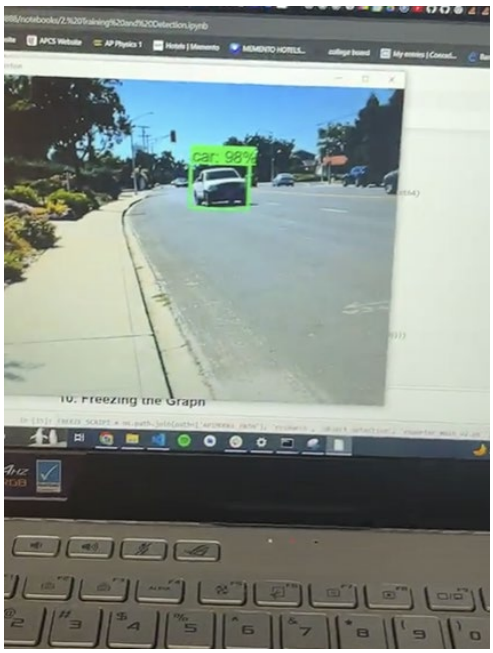

Figure 4. Display output from our model. The video feed is overlaid with a bounding box with the model's confidence. Analyzing the confidence and the model's ability to track moving cars allows us to make performance cuts and adjustments.

Physical Device

Our prototype consists mainly of a Raspberry Pi 4, a case, wideview USB camera, external mini-speakers, and a Raspberry Pi battery pack (figure 5). The Raspberry Pi is connected to the camera via USB cable, and to the battery pack with a built-in USB-C cable.

We are currently developing a way to attach the model underneath the bike seat. This could be done with a plastic clip that latches on to the rod. Alternatively, the user could "wear" the device on their back, with straps tied to loop over the user's shoulders like a backpack. Ultimately the setup of the system is not critical, as only the starting height has to be recorded and inputted to ensure the software is accurate, thus the physical model can be worn in any way desired as long as the camera faces directly behind the user.

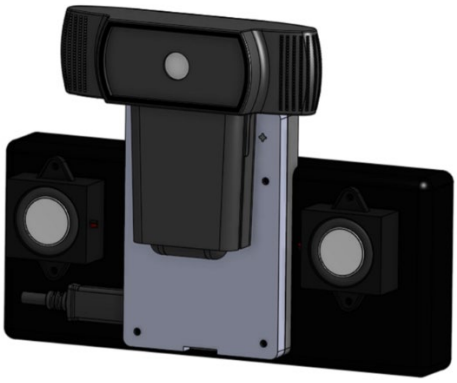

Figure 5. Through the CAD software Onshape, we created a 3D design model and prototype that fits our use case. Consisting of a Raspberry Pi 4, mini-speakers, portable battery, and a 1080p USB camera, the design is compact (2.4 x 4.9 x 7.0 inches).

Results

We independently tested the model's ability to detect cars and the distance algorithm's accuracy of predicting true distances. We decided to test both segments independently because it allowed us to identify each segment's effectiveness and determine which part needed more improvement in terms of accuracy and efficiency.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	243 (TP)	9 (FP)
	Negative	4 (FN)	52 (TN)

Model Figures

Model metrics were calculated with 308 independent test trials. We used a pre-recorded video and captured frames, and directly inputted the still images to the model. Images were captured in intervals of 3 seconds, interpreted manually, run through the model, then compared with our interpretations.

mAP: 96.4% $mAP = \frac{TP}{TP+FP}$
False Positive Rate: 0.147 $FPR = \frac{FP}{FP+TN}$
False Negative Rate: 0.0162 $FNR = \frac{FN}{FN+TP}$
Overall refresh rate: 71Hz (15 frames per second)

Distance Estimation

To test our distance estimation algorithm, we captured still images of a vehicle from varying distances from the front of the vehicle. Using a tape measure, we recorded the distance between the vehicle and the camera. Then for every "true distance" from the vehicle, our algorithm produced an "estimated distance".

We determined that at distances within 6.0 meters from the front of the vehicle, triangulation is the most accurate method for distance estimation. Although it is slightly more computationally intensive than bounding box estimation, we decided the slight efficiency drop was worth the greater accuracy.

From 6.0 meters and beyond, more rudimentary bounding box estimation was used. This was because triangulation lost some of its accuracy at greater distances, and also because it allowed us to reduce loop time by using simpler calculations.

The true distance and residuals (difference between true and estimated distances) were plotted (figure 6). The data displays that bounding box estimation (used for distances greater than 6.0 meters), on average, produced over-estimations of the true distance (our estimate is greater than the true distance). However, for distances between 2.0 to 6.0 meters, our triangulation estimation is more reliable. From 6.0 meters and beyond, the bounding box estimation consistently produces overestimations.

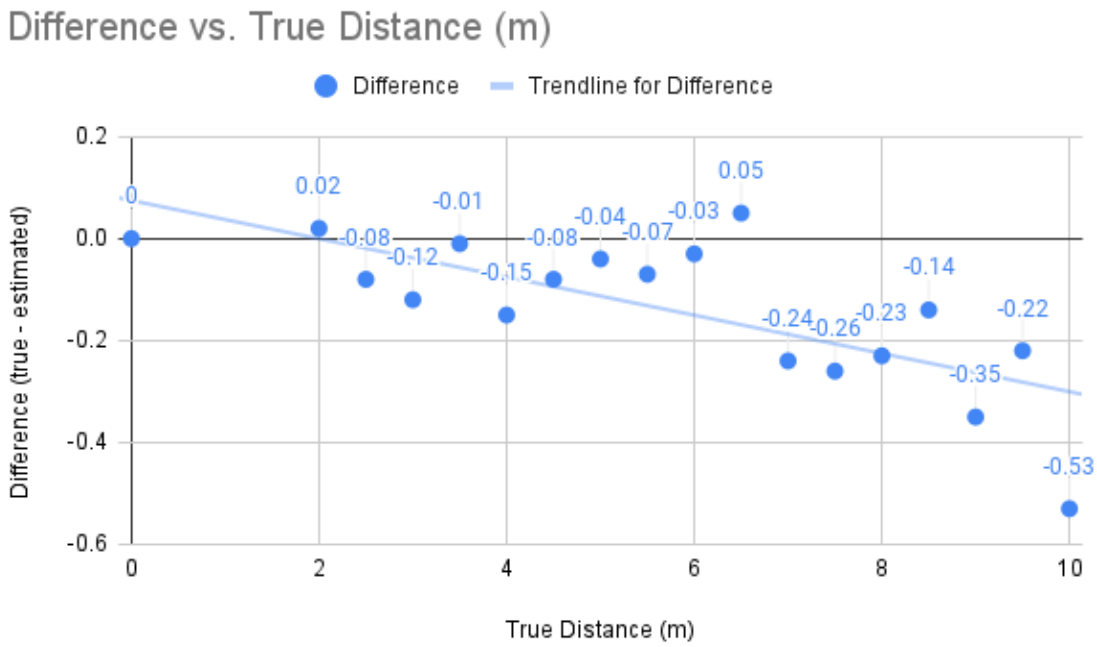

Figure 6. Effectiveness comparison of bounding box estimation (6m to 10m) and triangulation estimation (0m to 6m).

Discussion

Future Implementations

Image Segmentation: A common use of image segmentation is in object detection. Instead of processing the entire image, a common practice is to first use an image segmentation algorithm to find objects of interest in the image. Then, the object detector can operate on a bounding box already defined by the segmentation algorithm. This prevents the detector from processing the entire image, improving accuracy and reducing inference time.

Model Quantization: Quantization refers to techniques for performing computations and storing tensors at lower bit widths than floating point precision. This allows for a more compact model representation and the use of high performance vectorized operations on many hardware platforms. Quantization is primarily a technique to speed up inference and only the forward pass is supported for quantized operators.

Issues

During our testing phase, we realized bicycles are prone to vibrations, so the image input would not be as stable as we had assumed. To counter this issue, we researched software image software image stabilization. Image stabilization programs crop the raw image frame and adjust the bounding coordinates translationally when the camera shifts due to external forces, which provides a more stable camera input for our detection and warning system. Image stabilization will be another future implementation.

Conclusion

With continued development and successful implementation of all three phases and a physical prototype, Third Eye has potential to be extremely useful to cyclists searching for a low-cost and effective visibility/situational awareness tool. At the individual scale, this innovation may be able to save a life, especially for bikers in city environments. The warning system provides additional time for the cyclist to react to an approaching car, drastically reducing the chance of a collision. But this device's impact will not only influence individual users, but will also have an impact on a much grander scale. With a much safer biking experience, Third Eye may promote a greener city culture, where people are encouraged to take a bike rather than a car, especially because the experience will be safer than before. With multiple cities and thousands of people adopting this biking lifestyle, we can make a notable effort to reduce carbon emissions and help save our environment. Thus, Third Eye will make significant impacts at multiple levels, from the individual to the global.

Limitations

Some limitations include weather and environmental variations, such as lighting, rain, or biking at nighttime. For example, very bright sunlight may reflect off car windshields and glare into the camera. Fog or mist might obscure the camera's vision, and heavy rain may damage the electronic equipment. Since our model is not trained with every single possible weather and environment condition, it is extremely difficult to create a model that accounts for all cases. There may also be privacy concerns, where some people may not want to be recorded in public.

Acknowledgments

We would like to thank Mr. Hugo Steemers for his mentorship throughout the process. We would also like to thank Lilun Cheng for feedback on our initial idea for our project.

References

Hartley, R., & Zisserman, A. (2004). Multiple view geometry in computer vision. Cambridge University Press. https://doi.org/10.1017/CBO9780511811685

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 1, pp. 519-528). IEEE. https://doi.org/10.1109/CVPR.2006.19

Behley, J., & Stachniss, C. (2018). Efficient surfel-based SLAM using 3D laser range data in urban environments. In Robotics: Science and Systems (RSS). http://www.roboticsproceedings.org/rss14/p32.pdf

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., ... & Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 163-168). IEEE. https://doi.org/10.1109/IVS.2011.5940562

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://arxiv.org/abs/1704.04861

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European Conference on Computer Vision (pp. 21-37). Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961-2969). https://doi.org/10.1109/ICCV.2017.322

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Adam, H. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2704-2713). https://arxiv.org/abs/1712.05877

Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:1806.08342. https://arxiv.org/abs/1806.08342

Fishman, E., & Cherry, C. (2016). E-bikes in the mainstream: Reviewing a decade of research. Transport Reviews, 36(1), 72-91. https://doi.org/10.1080/01441647.2015.1069907

Lindsay, G., Macmillan, A., & Woodward, A. (2011). Moving urban trips from cars to bicycles: Impact on health and emissions. Australian and New Zealand Journal of Public Health, 35(1), 54-60. https://doi.org/10.1111/j.1753-6405.2010.00621.x

Rojas-Rueda, D., de Nazelle, A., Tainio, M., & Nieuwenhuijsen, M. J. (2011). The health risks and benefits of cycling in urban environments compared with car use: Health impact assessment study. BMJ, 343, d4521. https://doi.org/10.1136/bmj.d4521

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1512.02325. https://arxiv.org/abs/1512.02325