Out-of-Distribution Detection and Fallback for Autonomous Vehicles

Ziheng Chen, Peiyuan Jin, Weihang Long, Heramb Joshi
zihengc2,peiyuan3,weihang6,hujoshi2@illinois.edu
University of Illinois Urbana-Champaign
Champaign, Illinois, USA

ABSTRACT

Handling out-of-distribution (OOD) scenarios remains a challenge in autonomous vehicles as more safety-related issues emerge. In the field of machine learning, OOD cases occur when a learned model encounters input that is not included in training, potentially resulting in decision-making failures. These OOD situations can manifest in various ways, but this project focuses specifically on OOD scenarios involving varying weather conditions that affect adverse weather, which can significantly distort visual inputs and sensor data. As the industry shifts towards end-to-end vision-based autonomous driving that increasingly relies on visual data, robust handling of OOD input becomes crucial. Current learning-based approaches to OOD detection, like RIP, are limited in their robustness and interpretability. Thus, our project aims to implement a statistical analysis to first detect OOD and then enable the fallback mode that encourages safer driving behaviors. We introduce three weather modifications that inject up to 100% more accidents and 50% change in vehicle route among all weathers on the baseline autonomous driving agent. We also proposed two OOD detection methods that can detect OOD with an accuracy of up to 92.2%. Finally, we achieved up to 50% reduction in collision score and 46% in route-change proportion with modified weather by enabling the fallback mode to promote safer driving actions when OOD is detected using our method. The results are available at https://github.com/zihengjackchen/OOD-Detection.

CCS CONCEPTS

• **Computing methodologies → Anomaly detection**.

KEYWORDS

Autonomous Vehicles, Computer Vision, Out-of-Distribution

1 INTRODUCTION

Machine learning-based decision-making has enabled remarkable progress in the field of autonomous vehicles, bringing us closer to realizing self-driving systems that can navigate complex environments. However, one significant challenge remains: handling out-of-distribution (OOD) cases—situations where the vehicle encounters conditions or scenarios not present in its training data, potentially resulting in critical decision-making failures [11]. This phenomenon is particularly concerning for autonomous vehicles due to the high-stakes nature of their operations, where safety is paramount.

For self-driving systems, OOD cases can emerge in various forms:

- Unusual Road Users: Encountering pedestrians, cyclists, or animals crossing the road unexpectedly, especially in rural or less predictable urban environments.
- Erratic Behavior of Other Drivers: Sudden or unusual maneuvers by other drivers, such as abrupt lane changes, running red lights, or aggressive driving [2].
- Unusual Road Layouts: Unfamiliar roundabouts [4], sharp turns, or temporary road configurations due to construction can confuse the vehicle's navigation system.
- Obstacles or Debris on the Road: Encountering unexpected obstacles like fallen branches, construction debris, or unmarked road hazards can challenge the vehicle's ability to navigate safely.

Out of the above categories, this project focuses on handling OOD situations involving different weather conditions, such as rain, haze, and shade. These conditions drastically alter visual inputs and disrupt sensor readings, impacting the vehicle's ability to perceive its surroundings accurately. The urgency of developing robust OOD detection is underscored by recent shifts in industry practices [8]. For instance, Tesla's transition to an end-to-end vision-based autonomous driving system represents a growing trend of relying heavily on visual data [10]. However, this dependence also magnifies the impact of vision-based OOD scenarios, which could severely undermine vehicle performance. In previous discussions, we observed significant limitations in current learning-based OOD detection methods like Robust Inference Pipeline (RIP) [4]. Although well-advertised, these models often struggle with interpretability and robustness when handling OOD data [2]. Therefore, in this project, we aim to implement a statistical-based analysis to detect OOD inputs more effectively and enable a fallback mode that encourages safer driving behaviors. We introduce three weather modifications, i.e., rain, haze, and shade, each with varying intensities, that lead to a significant increase in accident rates with the baseline autonomous driving agent. Our proposed OOD detection methods achieve up to 92.2% accuracy in identifying these weather-induced scenarios. By enabling the fallback mode to promote safer driving actions upon OOD detection, we achieved up to 50% reduction in collision score.

2 RELATED WORK

2.1 Learning-based OOD Detection

Robust Imitative Planning (RIP) is a learning-based method for detecting and recovering from out-of-distribution (OOD) scenarios in autonomous vehicles (AVs) [4]. RIP employs an ensemble of networks to imitate expert driving behavior, identifying distribution shifts by analyzing the variance between their generated plans. If a shift is detected, RIP optimizes the worst-case scenario to enhance AV safety. Although this approach offers promising robustness and adaptability, RIP and its adaptive variant, AdaRIP, can fall short due to several inherent limitations. First, the computational demands of using ensemble models for robustness increase processing requirements, potentially impacting system performance. Furthermore, it is uncertain that any of the models in the ensemble produces the perfect prediction, but RIP assumes an accurate imitation of expert policies, which may falter in unpredictable or high-energy environments, limiting its effectiveness in real-world OOD scenarios. The reliance on the CARNOVEL benchmark for evaluation also raises concerns about generalizability, as the method's performance outside this controlled environment remains uncertain [2].

Our approach makes sense because it uses statistical analysis to detect OOD scenarios more reliably and activates a fallback mode to encourage safer driving behavior. By focusing specifically on varying weather conditions that distort visual inputs and sensor data, our method provides improved accuracy in identifying OOD situations and enhances safety by enabling fallback actions that reduce accident rates during adverse weather.

2.2 Vision-based ADS

Our project uses the Learning-by-Cheating (LBC) agent, a state-of-the-art vision-based ADS, as the base agent to perform tasks [1]. LBC leverages a two-stage training process to train a sensorimotor agent to imitate a privileged agent effectively. In the first stage, a "white box" privileged agent is trained via conditional imitation learning, providing action predictions for various commands in any environment state. This privileged information is used to train the sensorimotor agent in parallel, enhancing data augmentation and learning capacity. The LBC agent architecture involves the privileged agent receiving a bird's-eye view image to generate heatmaps and waypoints for different commands, while the sensorimotor agent processes genuine sensory input to produce waypoints in the egocentric camera frame. These waypoints are selected based on the command, projected into the vehicle's coordinate frame, and utilized by the low-level controller to output steering, throttle, and brake commands. In short, the input to the LBC agent includes images from a forward-facing camera in the CARLA Simulator [3], and the output consists of waypoints in the egocentric camera frame that guide the low-level controller in generating driving commands based on the current environment and desired actions.

3 APPROACH OVERVIEW

Our proposed approach comprises multiple stages, as illustrated in the flowchart. Each stage serves a specific purpose and is introduced in detail in later sections. Together, they contribute to a comprehensive and robust solution from detecting to enabling fallback mode when encountering OOD.

3.1 System Design Overview

This project utilizes the learning-by-cheating (LBC) agent, a visual-based ADS. It takes images from three front-facing cameras and produces an action upon inference. In this project, we assume that we have direct access to these camera images provided in the CARLA simulator. Our system consists of the following key steps:

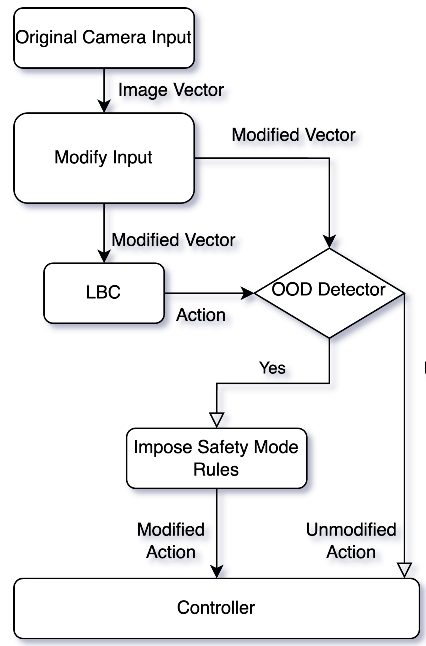


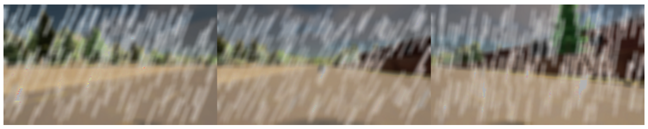
Figure 1: The flowchart illustrates the proposed pipeline for enhancing out-of-distribution (OOD) detection in autonomous vehicles. It begins with the modification of camera inputs to simulate OOD scenarios, followed by an OOD detector that determines whether the input data is within or outside the training distribution. In cases where OOD is detected, a fallback mode is triggered to enforce safety mode rules, prioritizing cautious driving behaviors such as reduced throttle and increased braking to mitigate potential risks. The learning-by-cheating (LBC) agent handles normal operations when the input data is within the training distribution.

(1) Input Modification: The process begins by modifying the camera inputs to simulate various OOD scenarios. These modifications are based on predefined parameters, such as adverse weather conditions, i.e., rain, haze, and shade. By altering the original camera input, we challenge the ADS using scenarios that are visually impaired that the vehicle may not have encountered during training.
(2) OOD Detector: The modified inputs are then passed through our robust OOD detection mechanism, explained in section 3.3. This module analyzes the visual data to determine if the current conditions differ significantly from the distribution that the LBC agent has seen and proven to perform well on. In the end, the LBC agent produces standard actions based on the given input.
(3) Fallback Mode Activation: If the data falls within the distribution, the action is unmodified and passed to downstream components. Upon detecting an OOD scenario, the system triggers a fallback mode that prioritizes safer driving behaviors. This fallback mode imposes specific safety rules, such as limiting throttle application and prioritizing braking. The modified action is passed to the controller and processed in downstream components.

This comprehensive approach provides a mechanism to effectively handle OOD scenarios and improve the reliability of autonomous systems.

3.2 Image Modification

(a) Original Camera Input



(b) Camera Input with Rain Overlay

(c) Camera Input with Haze Overlay

(d) Camera Input with Shade Overlay

Figure 2: Some example images that provide a clear comparison of the modifications, showing visual impact to various degrees.

3.2.1 Weather Parameters. Creating a rain effect involves 3 components: the real-world emulation of raindrops with varying drop transparency, the motion of raindrops with respect to the previous state of the image, and distortion in the camera added by moisture. We utilize a function that takes the input parameter as rain intensity, with values in a safe range of 0 (least amount of rain density) to 1350 (highest amount of rain density under test), with a blur effect proportional to the input rain density. The visual results are as per Figure 2b.

Haze emulation is tricky and computation-heavy because the haze effect is with respect to the current state of the input image, unlike the rain and shade effect. To solve this problem, we make use of a function that takes in input as haze intensity in a range of 0 to 225, where intensity reflects the number of haze blocks in perception, which are randomized in size and location, superimposed on the image as shown in Figure 2c.

To create an image with a shadow effect, we create a function that takes the intensity of shade from 0 (darkest) to 1.35 (as bright as the original image). A transparent transformation image is created in HLS format, with the Lightness coefficient reduced to the intensity specified, and this image is overlaid on the original input image to get the output as per Figure 2d.

3.2.2 Comparing with CARLA's Native Weather Settings. We chose to manually modify camera images based on weather conditions rather than relying on CARLA's native settings. CARLA's game engine-based architecture presents limitations in accurately simulating intricate and realistic weather patterns, particularly at close

Figure 3: This is the camera input using the native *heavy rain* weather configuration. Compared to Figure 2b, the weather is not severe enough to impair vision.

range to the image sensors, where the images are collected for the ADS. For instance, Figure 2b illustrates CARLA's depiction of heavy rain, which, although noticeable, neither adequately mimics real-world conditions nor substantially impairs visibility. Therefore, we directly enhanced these weather effects, as Figure 3 showcases the result—significantly diminished visibility that provides a more rigorous evaluation for our vision-based autonomous driving system.

3.3 Out-of-Distribution detection

We introduce two methods of detecting OOD images from our in-distribution image pool — Mahalanobis (Maha) distance and Autoencoder. We sample 11 scenarios where the ADS can perform normally without accidents. From each scenario, we collect a total of 97 camera inputs at 4Hz during the duration of each simulation, where each camera input consists of images from three sensors. Each image from a single sensor has dimension 256*144, and each pixel contains r (red), g (green), b (blue), and a (alpha) channels. We then construct our in-distribution dataset using these images.

3.3.1 Mahalanobis distance. The Mahalanobis distance [7] is a measure between a sample point and a distribution. Given our probability distribution Q from our dataset, with mean $\mathbf{u} = (u_1, u_2, u_3, \ldots, u_N)^T$, and a positive-definite covariance matrix S, the Mahalanobis distance of a point $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_N)^T$ from Q is defined as

$$d_M(\mathbf{x}, Q) = \sqrt{(\mathbf{x} - \mathbf{u})^T S^{-1} (\mathbf{x} - \mathbf{u})}$$

S^{-1} is guaranteed to exist because S is positive-definite. Constructing a covariance matrix from over 1000 images is extremely expensive due to its time complexity of $O(n^2)$, where n denotes the number of data points, which is roughly 2.2×10^8. We propose using a PCA model to achieve linear dimensionality reduction while retaining 95% of the variance of the sample.

Given the extensive number of pixels encompassed by the statistical models, we employ a Gaussian distribution[12] to represent them. This model aggregates the mean and variance of the Mahalanobis distance across all testing images. By comparing if the relative deviation of the training images exceeds twice its variance, the model determines whether they are OOD. Finally, since three images are collected, the input is confirmed OOD only when all left, middle, and right images are classified as OOD.

3.3.2 Autoencoder. An autoencoder[5] is a type of artificial neural network used for unsupervised learning. It works by compressing input data into a lower-dimensional representation, known as the encoding or latent space, and then reconstructing the original input data from this compressed representation. This process enables the network to learn a compact representation of the input data, capturing its essential features while discarding noise and irrelevant information.

In our design, the model architecture comprises a total of 12 layers, structured into an encoder and a decoder. The encoder incorporates three linear layers and three activation layers. We preprocess the training and testing images by flattening the position and RGBA information of every pixel into a 1D vector of size 147,456 as input. The model then transforms this input into a reduced feature space of 1024 features, followed by 512 features, and ultimately 256 features. Conversely, the decoder reverses this process, mapping the 256 features back to 512 features, then to 1024 features, ultimately reconstructing the original 1D vector of size 147,456. The model compares newly fed images with its output and calculates a mean-squared error (MSE) value. Similar to the last method, this model aggregates the mean and variance of the MSE across all testing images and determines whether the input image is OOD using a predetermined threshold. The input is confirmed OOD only when all left, middle, and right images are classified as OOD.

4 EXPERIMENTS

4.1 Testing Out-of-Distribution Detection

Our approach to validating the OOD detection mechanism involved two primary evaluation stages: the False Negative Test and the True Positive Test.

4.1.1 False Negative Test. We first tested the system's ability to recognize unmodified images correctly as in-distribution. We ran our OOD detection mechanism on all unaltered images representative of standard conditions to assess whether any false negatives would occur. The unaltered images are the images from three front-facing cameras collected in the CARLA simulator for training OOD detection methods. As expected, the system **accurately classified all unmodified images as in-distribution**.

4.1.2 True Positive Test. The second phase focused on assessing the system's ability to identify images modified with adverse weather conditions as out-of-distribution. We used a single frame of camera input and simulated a range of environmental changes across all three predefined weather types, each at 50 different intensities, which is at a higher granularity. This ensured comprehensive coverage of potential weather conditions that an autonomous vehicle might encounter.

Table 1: Detection rate given OOD images

Detection Method	Effect Name	Not Detected (%)	Detected (%)
Mahalanobis Distance	Rain	0 (0%)	**52 (100%)**
	Haze	51 (100%)	**0 (0%)**
	Shade	14 (18%)	**36 (82%)**
Autoencoder	Rain	0 (0%)	**52 (100%)**
	Haze	0 (0%)	**51 (100%)**
	Shade	12 (14%)	**38 (86%)**

4.1.3 True Positive Test Results. Table 3 provides a comparative analysis of the performance of two methods, Mahalanobis distance (Maha) and Autoencoder (Encoder), in predicting the in-distribution status for different effects. In this scenario, the Encoder method demonstrates strong performance across all effects, achieving high percentages of true positives for Shade, Rain, and Haze. Notably, Encoder-Rain and Encoder-Haze both achieve perfect true positive rates of 100%. Conversely, while the Mahalanobis distance method performs well for the Shade and Rain effects, achieving true positive rates of 82% and 100%, respectively, it notably fails for the Haze effect, where it achieves a true positive rate of 0%. This suggests that both methods exhibit relatively strong capabilities in predicting OOD scenarios under challenging circumstances, thereby ensuring the robustness of our pipeline.

Through this two-part evaluation strategy, we ensured that our OOD detection mechanism is accurate in identifying in-distribution cases and reliable in detecting challenging out-of-distribution scenarios. It also provides a solid foundation for safe fallback mode activation.

4.2 Testing Fallback Mode

4.2.1 Experiment Setup. For all testing, we rigorously tested the fallback mode of our system using the *lead slowdown* scenario. This specific scenario simulates a situation where the vehicle directly ahead suddenly decelerates, requiring the AV to respond promptly and brake appropriately. Our testing established 11 variations of the *lead slowdown* scenario. In their original state, none of these scenarios contained accidents, providing a baseline for comparison. Each scenario was tested under 3 different weather conditions, with each condition varied across 10 intensity levels. This allowed us to observe the impact of increasingly adverse weather on the AV's performance.

Our experiment has the following stages for comparison:

(1) Without Weather Modification: The unmodified camera inputs from the 11 configurations were used to establish a baseline performance level.
(2) Baseline with Modified Input: The inputs were modified with varying weather conditions, and the baseline performance was evaluated using a vanilla LBC agent across 330 unique configurations.
(3) With Fallback Mode Using Mahalanobis Distance: The final configuration utilized our fallback mode when the out-of-distribution (OOD) detection mechanism, based on Mahalanobis Distance, identified altered weather inputs as out-of-distribution. The fallback mode was then triggered to enforce safer driving behaviors across the same 330 configurations.

The Mahalanobis Distance-based OOD detection was chosen for its efficiency and computational lightness compared to neural network-based autoencoders. Autoencoders, while effective, pose a high computational overhead when layered onto the neural network-driven LBC agents. The Mahalanobis Distance method enabled swift and accurate OOD detection without imposing significant processing delays, allowing the fallback mode to activate promptly.

4.2.2 Results. Table 2 and 3 provide analytical figures that represent deterrent behavior from the vehicle in a given weather condition. Route change probability gives the number of scenarios in which cross-lines/lanes are broken with respect to all scenarios for a particular weather condition. It is desirable to have 0% Route change probability. The route-change proportion is an aggregate quantifier of the amount of detouring happening for a route/lane diversion scenario. While it doesn't necessarily represent accidents or collisions, it is more of a safety protocol metric for which higher values are observed in most collision scenarios. The vehicle collision score is the representation of the % of routes not covered due to collision with another vehicle from the overall route prescribed. This comes from the overall safety score achieved by the vehicle in all scenarios and is desirable to be 0%. The vehicle collision probability gives a proportion of a number of scenarios in which the vehicle actually collided, w.r.t. all scenarios. It is desirable to have this metric as 0% as well, but it represents just the collision probability, isolated from the overall safety of the vehicle. Layout collision score, similar to vehicle collision score, is the probability of a vehicle colliding with static objects like walls/trees, etc.

Table 2: Fallback activation using Baseline

Ego-vehicle behaviour	Haze	Shade	Rain
Route change probability	0.0	0.0	**50.0**
Route change proportion	0.0	0.0	**15.4**
Vehicle collision score	40.0	4.0	**28.00**
Vehicle collision probability	100.0	10.0	**10.0**
Layout collision probability	0.0	0.0	**35.45**

Table 3: Fallback activation using Mahalanobis-distance

Ego-vehicle behaviour	Haze	Shade	Rain
Route change probability	0.0	0.0	**31.82**
Route change proportion	0.0	0.0	**8.44**
Vehicle collision score	40.0	4.0	**14.00**
Vehicle collision probability	100.0	10.0	**11.82**
Layout collision probability	0.0	0.0	**2.73**

4.2.3 Observations. Our system has a significant improvement in overall vehicle safety when it uses Mahalanobis distance as compared to the baseline mechanism in Rainy weather scenarios, as observed from Table 3. From the results, we can make the following observations: Route change probability reduces in Mahalanobis distance by 38%, meaning that the number of scenarios in which cross-lines/lanes are broken reduces in the Maha-distance mechanism. The route-change proportion falls by 46% as well. This denotes an overall improvement in vehicle driving safety.

The vehicle collision score falls to 50% of its counterpart, suggesting a fair improvement. However, the vehicle collision probability increases from 10% to 11% in the maha-distance mechanism. When we put together the results of the above two metrics, we can state that the possibility of collision increases in our system, but the collisions are slower and safer. Along with this, the Layout collision score also reduces by a factor of 92.4%, accounting for improved fallback triggering when avoiding static elements in the scenario.

From these results, we can state that the Mahalanobis distance mechanism works well for OOD detection when the image is distorted by noise, which is similar to the effects of weather conditions, such as rain combined with humidity. The Fallback mechanism, when combined with Maha-distance, improves the safety of the vehicle by reducing collision probability by an aggregate of 69.2% for Rainy-weather scenarios. Along with this, we can also observe that there's neither any improvement nor deterioration in the Fallback system when it comes to other weather/light conditions like Haze and Shade.

5 DISCUSSION

The Mahalanobis Distance we chose to implement is robust when it comes to rainy-weather images. However, it cannot correctly identify hazed images and shaded images as OOD. On the other hand, the Autoencoder performs well with all three types of images. However, achieving this level of performance comes at a cost, as its heavy computational requirements hinder the flow of our CARLA simulation. As we choose to go with Mahalanobis distance, it significantly improves the performance of Fallback mode in rain scenarios. Based on observations, it reduced the probability of collisions by up to 46% and the probability of detouring by up to 38%.

6 FUTURE WORK

In future work, we intend to strengthen our OOD detection system by integrating a perception module that preprocesses and denoises camera inputs [6, 9], specifically addressing rainy and hazy conditions to improve compatibility with our autonomous vehicle. Moreover, we will extend our fallback mode to initiate not only safer actions like braking but also to engage a specialized agent that can safely park the vehicle in critical situations. These enhancements will substantially improve the robustness and safety of our autonomous driving system, equipping it to handle unexpected scenarios with even greater reliability.

7 CONTRIBUTIONS

Jack: (1) Testing and modifying OOD detection method and fallback mode. (2) Integrating OOD detection method and fallback mode into the pipeline. (3) Designing and running experiments.
Barney: (1) Implementation & testing of OOD detectors. (2) OOD distribution results analysis & visualization.
Weihang: (1) Implementation & testing of OOD detectors. (2) OOD distribution results analysis.
Heramb: (1) Implementation of image noise modification. (2) Implement an automated pipeline to analyze results from testing.

REFERENCES

[1] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. 2019. Learning by Cheating. In *Conference on Robot Learning (CoRL)*.
[2] Shengkun Cui, Saurabh Jha, Ziheng Chen, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. 2024. iPrism: Characterize and Mitigate Risk by Quantifying

Change in Escape Routes. In *Conference on Dependable Systems and Networks (DSN)*.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. arXiv:1711.03938 [cs.LG]

[4] Angelos Filos, Panagiotis Tigkas, Rowan Mcallister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. 2020. Can Autonomous Vehicles Identify, Recover From, and Adapt to Distribution Shifts?. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 3145–3153. https://proceedings.mlr.press/v119/filos20a.html

[5] M.A. Kramer. 1992. Autoassociative neural networks. *Computers Chemical Engineering* 16, 4 (1992), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A Neutral network applications in chemical engineering.

[6] Guofa Li, Yifan Yang, Xingda Qu, Dongpu Cao, and Keqiang Li. 2021. A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems* 213 (2021), 106617. https://doi.org/10.1016/j.knosys.2020.106617

[7] Prasanta Chandra Mahalanobis. 1936. On the Generalised Distance in Statistics. *Proceedings of the National Institute of Sciences of India* 2, 1 (1936), 49–55. http://library.isical.ac.in:8080/xmlui/bitstream/handle/10263/6765/Vol02_1936_1_Art05-pcm.pdf Retrieved 2016-09-27.

[8] Monirul Islam Pavel, Siok Yee Tan, and Azizi Abdullah. 2022. Vision-Based Autonomous Vehicle Systems Based on Deep Learning: A Systematic Literature Review. *Applied Sciences* 12, 14 (2022). https://doi.org/10.3390/app12146831

[9] Yuhang Qiu, Yongcheng Yang, Zhijian Lin, Pingping Chen, Yang Luo, and Wenqi Huang. 2020. Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV. *China Communications* 17, 3 (2020), 46–57. https://doi.org/10.23919/JCC.2020.03.005

[10] Inc. Tesla. 2024. *Tesla Vision Update: Replacing Ultrasonic Sensors with Tesla Vision*. https://www.tesla.com/support/transitioning-tesla-vision

[11] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized Out-of-Distribution Detection: A Survey. arXiv:2110.11334 [cs.CV]

[12] Xinhua Zhang. 2010. *Gaussian Distribution*. Springer US, Boston, MA, 425–428. https://doi.org/10.1007/978-0-387-30164-8_323