



ARTICLE

An Ultralytics YOLOv8-Based Approach for Road Detection in Snowy Environments in the Arctic Region of Norway

Aqsa Rahim^{*}, Fuqing Yuan and Javad Barabady

Department of Technology and Safety, UiT The Arctic University of Norway, Hansine Hansen veg 18, Tromsø, 9010, Norway

^{*}Corresponding Author: Aqsa Rahim. Email: aral13@uit.no

Received: 27 November 2024; Accepted: 01 April 2025; Published: 19 May 2025

ABSTRACT: In recent years, advancements in autonomous vehicle technology have accelerated, promising safer and more efficient transportation systems. However, achieving fully autonomous driving in challenging weather conditions, particularly in snowy environments, remains a challenge. Snow-covered roads introduce unpredictable surface conditions, occlusions, and reduced visibility, that require robust and adaptive path detection algorithms. This paper presents an enhanced road detection framework for snowy environments, leveraging Simple Framework for Contrastive Learning of Visual Representations (SimCLR) for Self-Supervised pretraining, hyperparameter optimization, and uncertainty-aware object detection to improve the performance of You Only Look Once version 8 (YOLOv8). The model is trained and evaluated on a custom-built dataset collected from snowy roads in Tromsø, Norway, which covers a range of snow textures, illumination conditions, and road geometries. The proposed framework achieves scores in terms of mAP@50 equal to 99% and mAP@50–95 equal to 97%, demonstrating the effectiveness of YOLOv8 for real-time road detection in extreme winter conditions. The findings contribute to the safe and reliable deployment of autonomous vehicles in Arctic environments, enabling robust decision-making in hazardous weather conditions. This research lays the groundwork for more resilient perception models in self-driving systems, paving the way for the future development of intelligent and adaptive transportation networks.

KEYWORDS: Autonomous vehicles; self-driving vehicles; road detection; snow-covered roads; YOLOv8; road detection using segmentation

1 Introduction

Adverse weather conditions, such as smoke, fog, rain, or snow, can reduce driver's visibility and increase the likelihood of an accident. In the United States (US), 1.5 million people are killed, and 800,000 are injured in automobile accidents each year [1]. Snow-covered environments present unique challenges that differ from those observed in clear-weather environments. The concept of self-driving vehicles was first proposed in the 1920s and has continued to evolve [2]. Self-driving vehicles play a crucial role in our daily lives, with applications ranging from exploration to transportation and logistics. Many research and vehicle manufacturing companies, such as Google, Audi, Bayerische Motoren Werke (BMW), Ford, Honda, Mercedes, Toyota, and Tesla, have invested heavily in developing self-driving automobiles [3]. Recently, several advances in self-driving vehicles have marked significant milestones, enabling complete autonomy, particularly in challenging urban environments. These achievements have been made possible by advancements in sensor technology, including Light Detection and Ranging (LiDAR), Radio Detection and Ranging (RADAR), high-speed cameras, and accurate positioning systems [2]. Snow's reflective and absorptive properties can impact



sensors like LiDAR, RADAR, and cameras, which are crucial for autonomous vehicle perception. Light scattering, reduced visibility, and variations in surface texture due to snow accumulation hinder accurate object detection and tracking, making it difficult for vehicles to discern obstacles, road boundaries, and other critical elements [4].

The effectiveness of lane identification and tracking depends on the state of adequately maintained roads and their distinct lane markings. Due to this, smart cities are essential in the current autonomous car research area. A smart city is often associated with ecologically conscious or sustainable urban environments, aiming to improve public services while reducing costs [5]. The core goal of smart cities is to balance technological innovation and the economic, social, and environmental challenges that cities will face in the future. Smart cities that adopt the tenets of the circular economy are striking this balance, requiring a deeper collaboration between the public and the government [6]. The movement of commodities and resources in response to people's changing needs will vary significantly in the context of smart cities, altering the fundamental layout of metropolitan regions. Specifically, top automakers such as Audi and Tesla have already released autonomous cars for private use, indicating a potential social impact when integrated into city transportation networks [3]. This progression can significantly alter cities and impact long-term urban planning, particularly when combined with the growth of smart cities and the integration of connected and autonomous vehicles.

The key components of self-driving car technology include environment perception, path planning, and intelligent control systems [5]. Object detection plays a vital role in developing smart cities and autonomous ecosystems. It provides valuable and precise traffic information for various applications, including traffic image analysis and flow control. This data includes vehicle counts, trajectory, tracking, flow, classification, density, velocity, lane changes, and license plate recognition [6]. It is also helpful in detecting other road assets, including traffic lights, vehicle types, persons, and natural elements. The perception module is crucial for accurately understanding the surroundings and ensuring the driver's safety. This is achieved by fusing multiple sensors, including cameras, LiDAR, RADAR, Global Positioning System (GPS), Inertial Measurement Unit (IMU), and sonar, to determine the vehicle's position in real time. In a typical environment, just a few sensors are sufficient for accurate object recognition and navigation. However, during challenging environments such as fog, rain, or snow, the accuracy and performance of the sensors become compromised, which can lead to the failure of some sensors [7]. Perception methods help autonomous cars and drivers in such challenging situations.

Additionally, the interaction between snowy ground and autonomous vehicles requires path planning and control algorithms. Vehicle dynamics, unpredictable road contact, and changes in tire grip can all necessitate an effective decision-making process [8]. Some complex methods eliminate the need to maintain the specific trajectory while ensuring passenger safety and effective navigation.

1.1 Motivation

Nowadays, fully automated vehicles are in the prototype phase for some commercial businesses, particularly logistics and courier companies. A notable example is Amazon delivery. This is not only an optimal opportunity for a company in terms of automation, but it also allows cost savings, increased safety, and reduced human intervention, ultimately providing high customer value. In the future, companies will entirely rely on such technologies. To achieve this, the structure and architecture of the technology should be fully robust, reliable, agile, and foolproof, enabling it to cope with any hazardous or harsh conditions without loss or with minimal loss.

1.2 Problem Statement

The snow-covered highways and severe weather conditions make it difficult to distinguish between drivable areas and obstructions such as sidewalks, curbs, and traffic signs. Ensuring the safe operation of autonomous vehicles in adverse environments, particularly in Arctic or snowy conditions, remains a significant challenge. Snow introduces complex issues that can impair an autonomous vehicle's perception and decision-making processes, demanding more accurate path detection and navigation systems. Recent advancements in artificial intelligence, particularly deep learning-based computer vision methods, have revolutionized perception systems through techniques like semantic segmentation. However, the performance of these models heavily depends on the quality and diversity of the training datasets. While several public datasets exist for semantic segmentation, most of which are captured under clear weather conditions with visible roads, traffic signs and obstacles [9]. This research aims to improve segmentation performance in heavy snow driving scenarios by applying semantic segmentation techniques tailored to challenging environments.

1.3 Our Contribution

This study introduces a novel framework that integrates SimCLR, which reduces reliance on annotated data, and Uncertainty Estimation (Monte Carlo Dropout and Bayesian YOLO) to quantify prediction confidence in extreme snowy environments. An extensive hyperparameter tuning exercise is also performed on YOLOv8 to optimize batch sizes and training epochs for the challenging Arctic Road conditions. The following are our main contributions.

- A custom dataset is build comprising 20,000 images from Tromsø, Norway, covers a range of snow textures, lighting conditions, and road geometries. This dataset enhances the generalization ability of road detection models in Arctic environments.
- A SimCLR-based contrastive learning approach is proposed to pre-train YOLOv8's backbone on unlabeled road images. This approach would reduce reliance on manually annotated data while significantly improving feature extraction in snowy conditions.
- An Integration of Monte Carlo Dropout and Bayesian YOLO to quantify prediction confidence, which ensure a more reliable real-time detection under challenging visibility conditions.
- Extensive hyperparameter tuning, which includes the adjustments of batch size and epochs, enhances the efficiency of YOLOv8. An mAP@50 score of 0.99 and an mAP@50-95 score of 0.97 are achieved using the proposed model on this custom-built dataset, and they outperform the state-of-the-art model.

These contributions address critical limitations in existing YOLO-based models, enhancing detection accuracy while improving generalization across diverse snow textures and illumination conditions.

2 Related Work

Advances in driver assistance systems often precede the growth of autonomous driving systems. Different algorithms, such as fusion algorithms, neural networks, and particle filters, have been employed to enhance the navigation systems of autonomous vehicles [10]. Tesla Motors has integrated autopilot technology into its electric cars, which uses cameras and sensors to predict crashes with up to 76% accuracy. This results in a collision avoidance rate of over 90%. Over the next 15 to 20 years, several automakers, including Tesla, Waymo, Uber, and others, predict a future dominated by autonomous vehicles [11]. They have discussed the different levels of vehicle automation, as defined by the Society of Automotive Engineers (SAE), and highlighted the contrast between level 5 (fully automated) and level 3 (partially automated) vehicles [12].

2.1 Artificial Intelligence-Based Method for Autonomous Vehicle Navigation

During the transition from partially to fully autonomous driving, an Artificial Intelligence (AI) system manages almost all functions, with minimal human intervention when necessary [13]. Big data plays a vital role in enhancing self-driving technologies, as it is impossible to create exhaustive rules for every scenario a vehicle might encounter. Machine learning (ML) and deep learning (DL) techniques allow vehicles to learn from real-world driving [14], detect patterns, and develop inference models [15]. The cloud-based real-time transmission of data required for autonomous vehicles has increased interest in multi-agent deep reinforcement learning (DRL) algorithms for managing various automotive systems. Traditional planning algorithms are impractical for path-planning tasks in self-driving cars in complex scenarios. Deep learning holds great promise for overcoming the limitations of traditional planning algorithms and learning to plan effective pathways under various scenarios for self-driving automobiles [16]. Deep learning techniques can help analyze the spatial-visual dynamics to understand the surrounding environment better, while cameras face challenges in adverse weather conditions, such as snow and fog.

A deep encoder-decoder network can be used to detect boundaries and find regions that are sensitive to obstacles [17]. Convolutional neural networks (CNN) are proven effective in object detection in self-driving cars due to their capacity to detect small patterns. Deep learning models can handle real-time data and communicate with surrounding vehicles and cloud servers using Graphics Processing Units (GPU) and cloud computing-based analysis. Transfer learning can also enhance accuracy, precision, and recall in object detection. Recent advances in computer vision algorithms have improved scene classification, obstacle detection, and lane recognition, primarily through the use of Artificial Neural Networks (ANNs) for self-driving cars. These systems keep track of an object in an ever-changing environment. Lei et al. [2] used a snowy driving dataset for object detection in adverse weather conditions. The dataset was later utilized for pixel-wise semantic labeling, achieving 92.7% accuracy and a mean Intersection over Union (IoU) score of 66.3% using the ICNet model. Convolutional autoencoders (CAEs), a recent architectural advancement, have enhanced feature representation while preserving spatial information [18]. In a convolutional autoencoder (CAE), the spatial information of a two-dimensional signal is preserved. It also replaces the connection layer in traditional autoencoders with a convolutional and pooling layer. In the decoding stage of the convolutional autoencoder, an operation opposite to convolution, also known as deconvolution, is carried out. A low-resolution feature representation is transferred to the input resolution by creating a deconvolution network, and the network achieves precise border localization with pixel-wise supervision [19]. Recurrent neural networks, particularly those with long and short-term memory capabilities, have proven extremely useful for maintaining model dependencies between current and prior input values [18]. There are many practical approaches available for controlling the hidden layer, such as introducing deviations from the input. Artificial intelligence-based systems can raise safety concerns when human intervention is required, significantly when the driver is impaired or incapable of operating the vehicle safely. An over-reliance on technology can make drivers passive, affecting their ability to take control when required [20], for example, in level 2 vehicles on autopilot, such as when a driver falls asleep.

Two key metrics assessed in deep learning models, especially in convolutional neural networks (CNNs), are generalization and robustness, as they are essential for real-world computer vision tasks. Generalization ensures that a model performs well on training and unseen data by minimizing the effects of overfitting. The model must be robust enough to withstand changes in illumination, weather, sensor noise, and other domain-specific variations to ensure reliability across various scenarios. The model must also perform reliably in dynamic situations. Since flexibility defines the model's effectiveness in real-world scenarios, it enhances the model's usefulness and dependability beyond the limited training conditions [21].

2.2 Traditional Image Processing for Autonomous Vehicles

Traditional image processing techniques, including image filtering, feature extraction, and edge detection, are essential for enhancing data used in object detection and scene understanding tasks. These methods can also detect snowcat tracks in snowy terrain.

Vachmanus et al. [22] combined noise-filtering techniques and morphological operations to classify image components. They collected a dataset in Tomakomai City, Hokkaido, Japan. In this study, the authors have highlighted the challenges faced while driving on snow and ice-covered roads. Their work emphasized the importance of object detection and identifying drivable areas in such extreme conditions. However, the complexity of autonomous driving systems under snowy weather conditions becomes challenging due to the unpredictable nature of ice and snow. A computer vision-based method was used for detecting tracks in the snow, which was further improved by applying a Hough transform—a method for feature extraction in image processing [23]. Their proposed method achieved better detection accuracy in challenging environments. On the other hand, edge detection methods were also used to explore snow-covered environments. Broggi et al. [23] tried several edge detectors but found that the canny edge detector provided more accurate results in terms of edge detection. Ushma et al. [24] employed an object detection approach utilizing a cascade object detector based on the Viola-Jones detection algorithm. They have collected images of cars from freeways in Southern California. For object detection and classification, Florinabel [25] proposed a real-time image-processing-based method on a real-time image dataset. They have achieved a high precision score of 97.88% and recall of 97.47%, demonstrating that their method is effective for real-time applications. Balaji et al. [26] compared object detection and tracking methods, such as frame difference, background subtraction, and optical flow methods. The experimental results found that the background subtraction method is the most efficient for object detection as it captures object features in detail. Vachmanus et al. [5] approached the semantic segmentation of roads in a snowy environment by using a multi-modal fusion of Red, Green, and Blue (RGB) images and thermal (T) maps. This RGB-T segmentation method takes color images and thermal data as input to the neural network, thereby improving the detection of human subjects in snowy conditions. They used a snowy dataset and evaluated their proposed model based on mean accuracy (mAcc), mean intersection over Union (mIoU), and the mean Dice coefficient score (mF1). The experimental results demonstrated that the proposed model, which utilizes RGB-T fused input, outperformed the RGB-only input.

2.3 Deep Learning Models Utilizing Feature Visualization

CNN and other deep learning models can be interpreted using gradient-weighted class activation mapping (Grad-CAM). It calculates the gradient of the target class score with respect to the feature maps of the final convolutional layer. It takes this into account when visualizing the regions of the image that may impact the model's ability to make decisions. The GRAD-CAM pipeline includes several steps, including the forward pass, gradient computation, global average pooling, heatmap generation, and image overlay. Initially, the model receives the input image, and the convolutional layer generates the feature maps. The gradient of the predicted class scores is computed using the generated feature maps. These gradients are averaged using the global average pooling layer to assign an appropriate weight to each feature map. The most relevant region of the image is highlighted by generating a heat map, which is calculated using the weighted sum of the feature maps. The generated activation map (or heatmap) is projected on the original image to identify the most relevant regions of the image that affect the model's prediction. The Grad-CAM helps understand model explainability, debugging, error analysis, trust, dependability, and bias detection.

Previous approaches for object detection in adverse conditions have primarily relied on supervised learning, which heavily depends on labeled datasets. However, in Arctic environments, road boundaries are

often indistinct, annotations are scarce, and existing datasets fail to generalize across varied snow textures and lighting conditions. Moreover, prior works have largely neglected the role of uncertainty-aware detection, leading to high false positive rates due to snow glare. This study directly addresses these limitations.

3 Methodology

The methodology for this study is shown in Fig. 1. The data was collected from snow-covered roads in Tromsø, Norway, covering diverse locations such as city centers, residential areas, main streets, and university surroundings to ensure a comprehensive dataset for training. The collected dataset was partitioned into training, validation, and testing subsets to allow for a more effective model learning and evaluation. To enhance the YOLOv8 model's feature extraction capabilities, a Simple Framework for Contrastive Learning of Visual Representations (SimCLR) on unlabeled images of snowy roads was employed, before the fine-tuning on the labeled dataset. The dataset was partitioned into three subsets: 70% for training, 20% for validation, and 10% for testing. The training dataset optimized YOLOv8's feature representations, while the validation dataset monitored performance to prevent the model from overfitting. The testing dataset was reserved for evaluating the final model on unseen snow conditions.

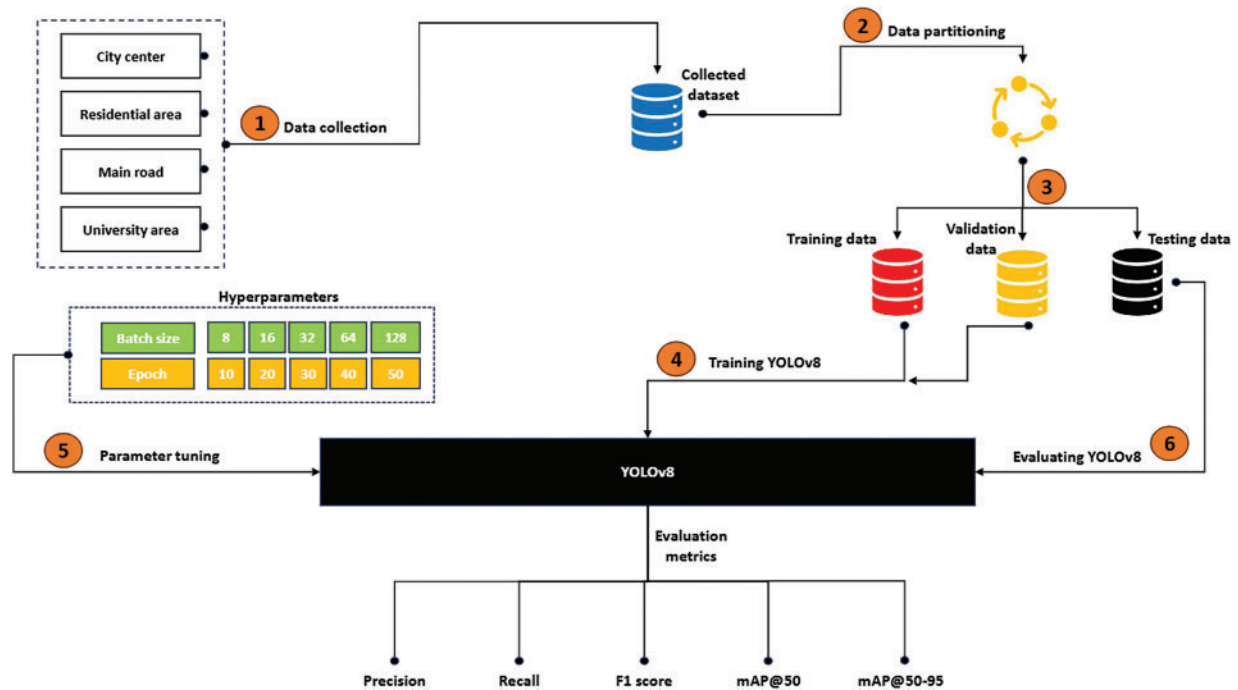


Figure 1: Flow diagram for proposed solution

Additionally, we incorporated uncertainty-aware detection techniques, integrating Monte Carlo Dropout and Bayesian YOLO during inference to estimate model confidence and reduce false positives caused by snow glare and occlusions. Hyperparameter tuning, including adjustments to batch size and epochs, was performed to maximize the model's performance and real-time efficiency. The performance of the proposed Self-Supervised pretraining and uncertainty-aware YOLOv8 model was evaluated using precision, recall, F1-score, mAP@50, and mAP@50–95. The following subsections provide a step-by-step breakdown of the methodology.

3.1 Data Collection

The data used in this research was collected in Tromsø, Norway, which is known as the capital of the Arctic Circle. Data was collected from November to April, when temperatures were consistently below freezing, marking a frigid winter season. During the creation of this data, adverse weather conditions, including snow, ice, fog, and slush, were encountered. This data was collected at multiple locations, including university areas, the city center, and residential neighborhoods. The data was collected for various road types, including straight and zig-zag roads, through images and videos. In total, 50 videos were recorded, with a maximum video duration of 5 min. A 108-megapixel camera with the specifications outlined in Table 1 was used to capture these images and videos. A data sample of a regular road, a residential area, a city center, and a university region is shown in Fig. 2.

Table 1: Camera specifications

Feature	Feature value
Resolution	1440 × 3088 pixels
Aperture size	F1.8
Focal length	23 mm
Sensor size	1/1.33"
Pixel size	0.8 μm
Video recording	7680 × 4320 (8K UHD) (24 fps)
	3840 × 2160 (4K UHD) (60 fps)
	1920 × 1080 (Full HD) (240 fps)
	1280 × 720 (HD) (960 fps)
Video capture	3840 × 2160 (4K UHD) (60 fps)

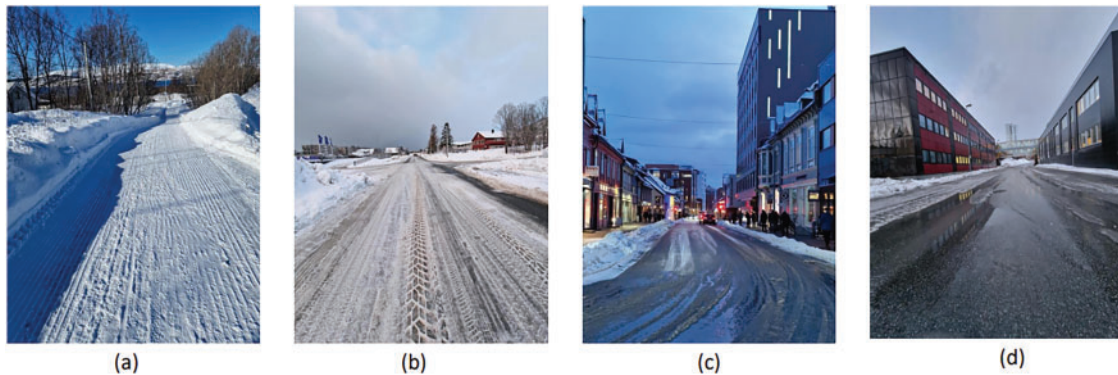


Figure 2: An example of (a) Regular road (b) Residential area (c) City centre (d) University area

The dataset comprises 20,000 labeled images, comprehensively representing various road conditions that may be encountered during the training of a road detection model. Out of the 20,000 images, approximately 12,000 feature straight road segments, while the remaining 8000 images capture zig-zag road patterns to ensure the model is exposed to different road geometries. This balanced dataset comprises 8000 images of snowy roads and 5000 images of dry roads with some snow on the sidewalks, providing a wide range of environmental variability. Additionally, 3000 images were acquired at night time to achieve a reasonable

variation in lighting and visibility. With its comprehensive coverage of road types and environmental conditions, this dataset is well-suited for training models that can handle a wide range of real-world road scenarios, thereby improving the generalization and robustness of models across diverse conditions.

3.2 Experimental Environment

The experiments were conducted on a powerful computer with a 12th Generation Intel Core i9-12900H processor at 2.50 GHz. The 32 GB of RAM provided significant processing capability. An NVIDIA GeForce RTX 3080 Ti laptop GPU was used to facilitate more effective experimentation. All experiments were conducted within an Anaconda environment. In the Jupyter Notebook, Python version 3.10.9 and Conda version 23.3.1 were used. The electricity required for training the model is a measure of energy consumption, which depends on the hardware, training duration, and workload. From a hardware perspective, the model can be trained on a Central Processing Unit (CPU) or a GPU and deployed on edge devices. The GPU consumes more electrical energy than the CPU. The hardware, dataset size, batch size, and number of epochs influence the training time of the YOLOv8 model. We utilized the GPU to train the YOLOv8 model, significantly reducing the training time.

3.3 Data Pre-Processing

A Python script was used to extract individual frames from the videos in the preprocessing step. The logic of extracting one frame per second from the video was implemented, resulting in a more diverse set of images with varying viewing angles. This approach was beneficial because it helped to avoid redundancy in consecutive frames and captured gradual environmental changes with varying lighting and textures. It also ensured a balanced dataset representing different road conditions. In total, 50 videos were processed, resulting in 20,000 high-quality images that improved the efficiency and precision of the dataset during training. Image annotation and labeling were performed after extracting individual frames from the videos. The dataset was created for semantic segmentation, utilizing polygon annotations. Unlike bounding boxes, polygon masks provide precise delineation of roads under snow. It also offers better edge detection for differentiating drivable paths from non-drivable areas. This requires the mapping of polygon points to all 20,000 images. In downstream operations, achieving reliable performance based on accuracy and detail is time-consuming. Several annotation platforms are available to create annotations according to project specifications. The Roboflow Universe platform created annotations on JSON files, making them compatible with the training model. Each image was labeled into drivable road areas.

In addition to data labeling, data augmentation was applied to improve generalization to unseen conditions. Each input image is transformed to generate two augmented versions, including random cropping, rotation, color jittering, and Gaussian blurring. Random cropping simulates partial obstructions due to snow. Rotation helps to recognize tilted perspectives of the road. Color jittering accounts for variations in lighting conditions, such as overcast or nighttime settings. Finally, Gaussian blurring simulates noise in conditions with foggy and snowy weather. These augmentations enable the model to better understand variations in road conditions without overfitting specific patterns. By integrating these preprocessing techniques, the proposed model can help improve generalization and enhance robust feature representations in adverse weather conditions.

3.4 YOLOv8

This study integrates the SimCLR pre-trained backbone with YOLOv8 for detecting snowy road conditions. Unlike standard YOLOv8 training, which requires extensive labeled data, this approach leverages SimCLR to pretrain YOLOv8's backbone on unlabeled road images, learning robust feature representations

even without ground-truth labels. This is particularly beneficial in low-contrast snowy conditions, where conventional models struggle to differentiate drivable paths from surrounding snow. The process begins with pre-training the YOLOv8 backbone using SimCLR, a self-supervised contrastive learning approach. Since collecting labeled data for snowy roads is resource-intensive, contrastive learning allows for leveraging large volumes of unlabeled road images. A contrastive loss function was applied to encourage representations of the same image under different augmentations, as discussed in the preprocessing stage. It is done to bring similar views closer in the feature space while pushing dissimilar views apart. The contrastive loss function is calculated using Eq. (1). During this phase, the backbone of YOLOv8, CSPDarknet, is trained without any labeled data using these contrastive learning techniques. This self-supervised learning approach enabled YOLOv8 to learn essential road textures, edges, and snow patterns before fine-tuning for the labeled dataset. Once pre-training is complete, the projection head used in SimCLR is removed, and the YOLOv8 detection head is attached.

$$L = - \sum \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^N \exp(\text{sim}(z_i, z_k)/\tau)} \quad (1)$$

After pretraining, the detection head was added to the model, and the entire network was fine-tuned with 20,000 labeled images using polygon-based segmentation masks to identify drivable paths on the road. Initially, the backbone layers are frozen to preserve the valuable features learned during pretraining, and only the detection head is trained. This allowed the model to adapt to detecting snowy road conditions while retaining the general features learned from SimCLR pretraining. After a few training epochs, the backbone layers are gradually unfrozen and fine-tuned with a lower learning rate to prevent overwriting the pre-trained features. To minimize false positives, hyperparameters were optimized using a grid search approach for Non-Maximum Suppression (NMS). Bayesian YOLO was incorporated to estimate uncertainty and enhance reliability in challenging Arctic conditions. In the traditional YOLO model, the dropout layers are only active during training. Bayesian YOLO enables dropout layers during inference, allowing the model to generate multiple predictions for the same input image. This provides a distribution of predictions, from which we compute variance to estimate uncertainty caused by snow glare and occlusions. The MC-Dropout process involves running T forward passes of YOLOv8 on the same input image. Then, we generate T different predictions due to the stochastic nature of the dropout. Finally, we compute the variance to quantify uncertainty using Eq. (2).

$$\sigma^2 = \frac{1}{T} \sum_{t=1}^T (y_t - \bar{y})^2 \quad (2)$$

where y_t represents the prediction at iteration t , $\bar{y}$ is the mean prediction over T iterations, and σ^2 quantifies the uncertainty in the model's output. Traditional YOLO models use a fixed confidence threshold; however, Bayesian YOLO introduces adaptive confidence thresholds, where the model dynamically adjusts detection acceptance criteria based on uncertainty estimates. Instead of a hard threshold, we define a probabilistic threshold as given in Eq. (3), where $\bar{c}$ is the mean confidence score of the model, σ is the estimated uncertainty (standard deviation of predictions), and k is a tunable parameter that determines the confidence level required to accept a detection. This ensures that predictions with low confidence and high uncertainty are automatically discarded, improving detection robustness.

$$\theta = \bar{c} - k \cdot \sigma \quad (3)$$

The mean confidence is computed by running the model T times on the same input image and averaging the confidence scores of the predicted bounding boxes. It can be calculated using Eq. (4), where c_t represents the confidence score of the detection at the t -th forward pass, and T is the total number of forward passes.

$$\bar{c} = \frac{1}{T} \sum_{t=1}^T c_t \quad (4)$$

4 Results and Discussion

In this section, results obtained using the most recent object detection and semantic segmentation algorithm, YOLOv8 with SimCLR-based self-supervised pretraining, are discussed and presented.

4.1 Evaluating the Impact of SimCLR Pretraining

The effectiveness of SimCLR-based self-supervised pretraining was validated through an ablation study, comparing YOLOv8 trained without pre-training and YOLOv8 pre-trained with SimCLR before fine-tuning on the labeled dataset. Table 2 presents the evaluation metrics for both settings. The results demonstrate that SimCLR pre-training significantly improves the feature extraction, which leads to a higher precision and mAP scores. The improvements are particularly evident in low-contrast areas, where the model struggled to differentiate road boundaries from surrounding snow without pretraining. Fig. 3 validates that self-supervised pretraining enables the model to learn meaningful snow textures and road structures even without manual labels, and improves generalization.

Table 2: Performance of YOLOv8 model with and without pre-training

Metric	YOLOv8 (No pre-training)	YOLOv8 (Pre-training)
Precision	0.91	0.97
Recall	0.89	1
F1-Score	0.90	0.98
mAP@50	0.89	0.99
mAP@50–95	0.80	0.92



Figure 3: Detected path using YOLO pre-trained model

4.2 Analysis of False Positives and False Negatives

The result of uncertainty-aware detection was assessed by analyzing the impact of Bayesian YOLO and Monte Carlo Dropout (MC-Dropout) on false positives and false negatives. Traditional YOLO-based models

uses fixed confidence thresholds, which fail in dynamic snow conditions, resulting in the misclassification of high-glare snow as drivable road surfaces. In contrast, the proposed Bayesian YOLO approach dynamically adjusts detection confidence based on uncertainty estimation, thereby filtering out low-confidence detections that are often caused by snow glare and occlusions. Fig. 4 illustrates the improvements in false positive and false negative rates after integrating Bayesian YOLO and Monte Carlo Dropout. The false positive rate was reduced by 55.3%, and the false negative rate dropped by 50.7%. Additionally, $mAP@50-95$ saw a 9.0% increase, confirming improved robustness in extreme snowy conditions. These the results demonstrated the importance of uncertainty-aware detection in Arctic environments, leading to a more reliable segmentation of drivable paths.

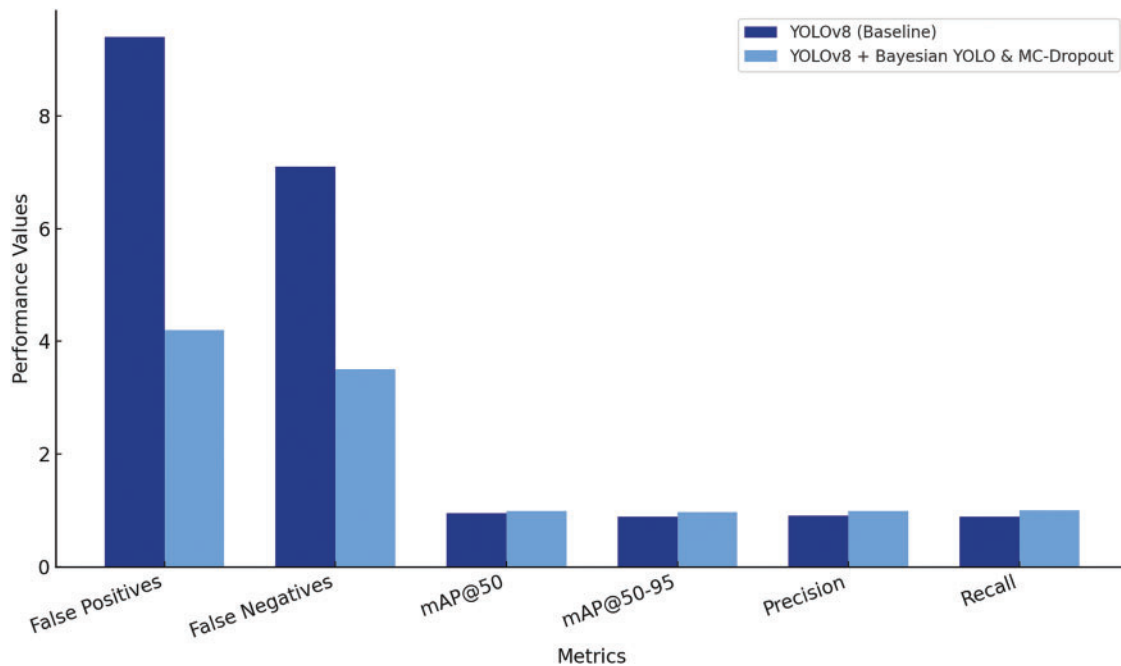


Figure 4: Impact of Bayesian YOLO and Monte Carlo Dropout (MC-Dropout)

4.3 Epochs

The model was trained using different numbers of epochs to analyze how training duration impacts the performance. Initially, it was trained for 50 epochs. Then, the number of epochs was gradually reduced to assess whether similar performance could be achieved with fewer epochs. The results on the training data across different epochs are shown in Fig. 5. As demonstrated in Fig. 5, increasing the number of epochs from 10 to 20 significantly improves the model's performance in terms of recall, F1 score, $mAP@50$, and $mAP@50-95$. A further increase of epoch to 30, 40, and 50 show a continuous improvement in results, but with minor improvements, particularly for $mAP@50$, which begins to stabilize. This indicates that with each additional epoch, the model gains a better understanding of the data and updates its weights more effectively. However, the improvements become marginal after a certain point, showing that the model has reached a stable learning phase. It is important to note that evaluating model performance solely on training data is insufficient. A model that performs well on training data might overfit—that is, because it memorizes the training examples but fails to generalize to new and unseen data. To address this, the model was evaluated on a validation set comprising 20% of the dataset, as described in the data description section. The results are presented in Fig. 6. As shown in Fig. 6, increasing the number of epochs from 10 to 20 also enhances

the model's performance on the validation data, as indicated by metrics such as precision, recall, F1 score, mAP@50, and mAP@50–95. This demonstrates that the model is learning relevant data patterns and does not require additional training beyond this point. To ensure the model's generalization capability, it was also evaluated on a separate test set, comprising 10% of the entire dataset. The results are illustrated in Fig. 7. As shown in Fig. 7, the model achieves 100% precision on the test data, along with strong performance in recall, F1 score, and mAP metrics. However, as the number of epochs increases beyond a certain point, the performance metrics such as precision, mAP@50–95, and F1 score start to decline, indicating overfitting. This can be mitigated using techniques such as early stopping, regularization, or reducing model complexity. Based on these observations, stopping training at 10 epochs appears to be an optimal point to balance performance and generalization.

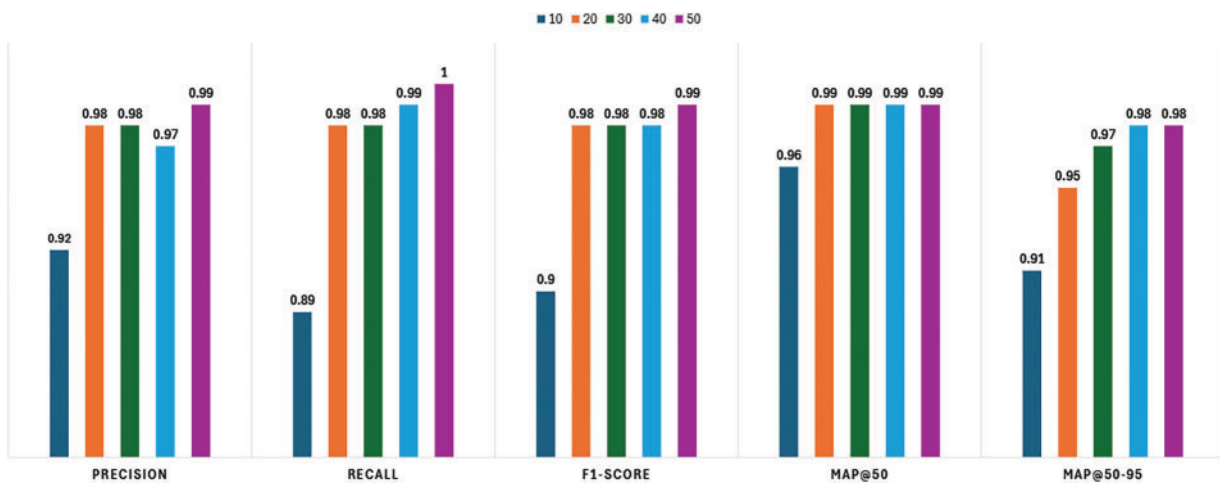


Figure 5: Performance metrics of the model across training data with varying epoch

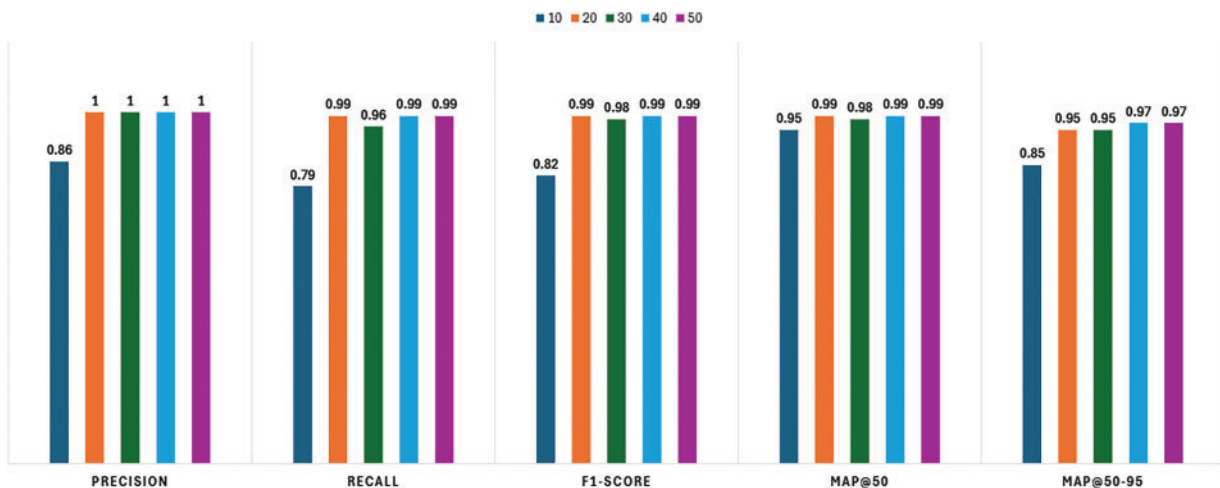


Figure 6: Performance metrics of the model across validation data with varying epoch

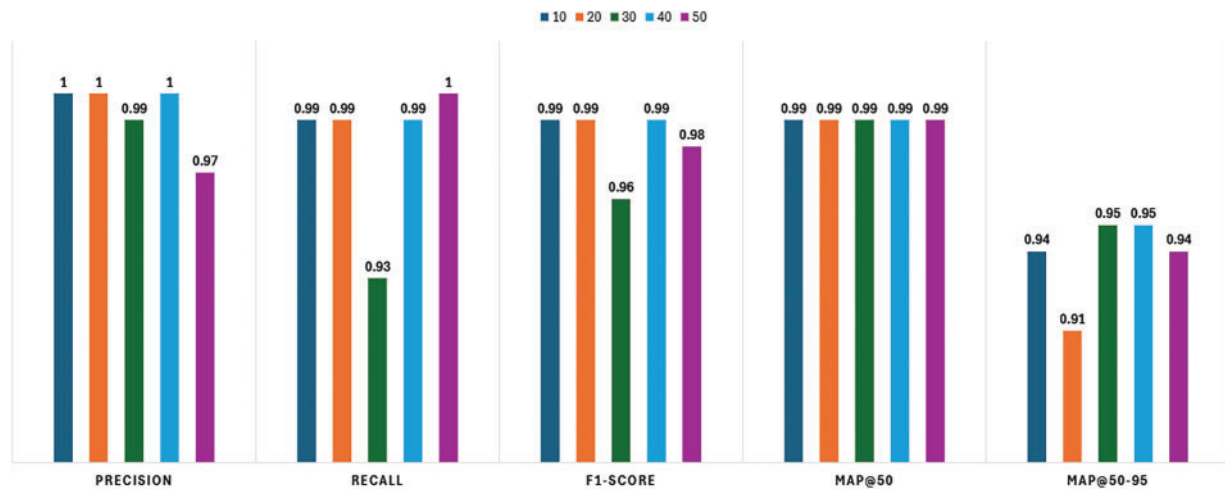


Figure 7: Performance metrics of the model across testing data with varying epochs

4.4 Batch Size

To evaluate the effectiveness of the proposed model, various batch sizes were also tested, including batch size of 8, 16, 32, 64, and 128. Table 3 presents the results for different batch sizes. These results indicate that the model performs best with a smaller batch size, particularly with a batch size of 8, achieving a mAP@50–95 of 97%. This can be attributed to the behavior of the gradient descent algorithm during training. When the batch size is small, fewer samples are used in each forward pass, resulting in more frequent weight updates based on a smaller subset of data. This enables the model to respond quickly to changes in the error signal, thereby reducing the risk of overfitting and enhancing generalization. Conversely, with larger batch sizes, more samples are processed before each update, and the model's weight adjustments are influenced by the average error over a broader dataset. While this can stabilize training, it may also increase the likelihood of the model being misguided if the combined signal from a large batch does not reflect the actual gradient direction. As a result, the optimization process may converge more slowly or settle into suboptimal solutions.

Table 3: Performance metrics of the model across training data with varying batch sizes

Batch size	Epochs	Precision	Recall	F1-Score	mAP@50	mAP@50–95
8	10	0.97	0.93	0.95	0.95	0.86
	20	0.93	0.98	0.96	0.98	0.95
	30	0.98	0.93	0.95	0.98	0.91
	40	1	0.99	0.99	0.99	0.97
	50	1	0.93	0.96	0.99	0.96
16	10	1	0.9	0.99	0.99	0.94
	20	1	0.99	0.99	0.99	0.91
	30	0.99	0.93	0.96	0.99	0.95
	40	1	0.99	0.99	0.99	0.95
	50	0.97	1	0.98	0.99	0.94
32	10	1	0.39	0.56	0.92	0.81
	20	0.99	0.93	0.96	0.97	0.95
	30	0.97	0.93	0.95	0.96	0.91

(Continued)

Table 3 (continued)

Batch size	Epochs	Precision	Recall	F1-Score	mAP@50	mAP@50–95
64	40	0.93	1	0.96	0.99	0.96
	50	0.99	0.93	0.96	0.99	0.95
	10	0.003	1	0.006	0.73	0.65
	20	0.94	0.73	0.82	0.94	0.85
	30	0.87	1	0.93	0.98	0.96
	40	0.99	1	0.99	0.99	0.96
	50	0.99	1	0.99	0.99	0.97
	10	0.003	1	0.006	0.63	0.47
128	20	0.003	1	0.006	0.90	0.83
	30	0.003	1	0.006	0.94	0.91
	40	1	0.93	0.96	0.99	0.95
	50	0.003	1	0.006	0.96	0.91

4.5 Comparison with Other State-of-the-Art Models

The performance of the proposed framework for road detection in a snowy environment was compared with various other innovative methods. Karaa et al. [27] annotated a dataset of snow-covered road images using unsupervised methods. Because researchers had previously focused on images from rural areas, they also considered annotating images from urban areas. They addressed the class imbalance problem by using a customized convolutional neural network model. While their model performed reasonably well, with a precision of 0.79 and mAP@50–95 of 0.71, it struggled with the fine-grained segmentation of road boundaries, especially in low-contrast, snow-covered areas. Jiang et al. [28] combined the Whale Optimizer Algorithm and Back Propagation models to create a new model known as WOA-BP. They used inputs such as temperature, voltage variation, and water film thickness to predict road conditions. This method aimed to improve feature selection and classification accuracy. However, it demonstrated limitations in real-time inference and generalization across varied snow textures, yielding an F1-score of 0.70 and mAP@50–95 of 0.80. A PVDM-YOLOv8l model was proposed by Tahir et al. [29] for detecting vehicles and pedestrians in adverse conditions. They have included methods such as the convolutional block attention module (CBAM) for feature refinement and the swim transformer for global feature extraction. While their model improved pedestrian and vehicle detection in adverse conditions, its mAP@50 was 0.78, which is lower than the proposed approach, indicating potential limitations in handling extreme lighting variations caused by snow glare. Feng et al. [30] proposed the ADWNet model, which improves upon the YOLOv8 model for object detection in autonomous driving. Their model incorporated SLOU loss and multi-scale attention mechanisms, resulting in improved precision (0.89) and recall (0.88). However, it had a lower mAP@50–95 (0.80) than the proposed model, indicating reduced performance in the fine-grained segmentation of drivable areas. The results of these state-of-the-art models, compared with those of the proposed model, are summarized in Table 4.

4.6 Comparison with Other YOLO Variants

The performance of the proposed framework for path detection on snowy roads was compared with that of YOLOv5, YOLOv6, and YOLOv7 on the same dataset. It was observed that YOLOv5 did not yield a satisfactory result in the case of a road with small snow patches, as it has limitations in detecting small objects.

It also struggled to differentiate the small contrast changes on the road due to poor exposure adoption. In the case of YOLOv6, it yielded improved results for a road with small snow patches; however, the localization error remained. It also improved in differentiating small contrast changes on the road, but it still lacked fine granularity. In the case of YOLOv7, the model was significantly improved in terms of edge detection, but it generated a large number of false positives. It also classified the high-glare snow, making the results unreliable. However, in the case of the proposed YOLOv8 model, it yielded promising results in low-contrast areas, and feature extraction was significantly improved compared to other YOLO variants. It performs well in feature extraction due to the use of SimCLR, the Bidirectional Feature Pyramid Network (BiFPN), and an improved Non-Maximum Suppression (NMS) algorithm. Due to Uncertainty Estimation (Monte Carlo Dropout and Bayesian YOLO), the misclassification rate was also very low, making it a reliable solution for detecting snowy paths on the road. Table 5 presents a comparison of the results from other YOLO variants with those of the proposed YOLOv8 framework.

Table 4: Comparison with other state-of-the-art methods

Ref.	Model	Precision	Recall	F1 Score	mAP@50	mAP@50–95
[27]	CNN	0.79	0.72	0.74	0.80	0.71
[28]	WOA-BP	0.69	0.74	0.70	0.82	0.80
[29]	PVDM-OLOv8	0.86	0.81	0.79	0.78	0.71
[30]	ADWNet	0.89	0.88	0.90	0.85	0.80
	Our Model	0.99	1	0.99	0.99	0.97

Table 5: Comparison with YOLO variants

Model	Precision	Recall	F1 Score	mAP@50	mAP@50–95
YOLOv5	0.76	0.86	0.80	0.83	0.80
YOLOv6	0.83	0.80	0.82	0.81	0.74
YOLOv7	0.92	0.87	0.90	0.85	0.80
Our model	0.99	1	0.99	0.99	0.97

4.7 Discussion

The model's performance significantly improved due to the use of a hyperparameter optimization approach. The most important key parameters that were optimized include batch size and epochs. The learning rate was adjusted to strike a balance between speed and stability in achieving convergence, thereby avoiding overfitting during training. The batch size was set to optimize GPU memory usage, promoting generalization. Additionally, the NMS IoU threshold was adjusted to reduce the number of false positives. These modifications were evaluated using mean Average Precision (mAP) across intervals of IoU's mAP@50 and mAP@50–95. There was a significant increase in precision and recall compared to the model's baseline. These adjustments enhanced the model's accuracy in handling specific variations of road conditions covered with snow, thereby increasing the model's reliability.

The proposed model can be integrated into an autonomous driving system by connecting it with the vehicle control system, sensor fusion modules, and decision-making frameworks. Initially, the trained model can be stored in lightweight formats, such as pickle or TensorRT, and deployed on edge devices like NVIDIA

Jetson. Multiple cameras can be integrated into autonomous vehicles to enable real-time road detection, collecting data, and achieving high accuracy. The data from these cameras can be merged using sensor fusion techniques. The data obtained can be forwarded to the trained model for road segmentation and route planning, and the outputs can be sent through a real-time data pipeline for trajectory planning. In a real-world scenario, the roads detected by the proposed model's output serve as input for the vehicle's trajectory planning and control systems. The model generates bounding boxes or segmentation masks that designate the drivable area. These are subsequently fused through a sensor fusion module that utilizes LiDAR, Radar, GPS, and IMU to enhance accuracy and robustness. Incorporating machine-detected road features with real-time localization and mapping enables the system to intelligently adapt its navigation strategy in constantly changing conditions. In addition, smoothing techniques such as Kalman or particle filters can mitigate snow blockage, generated occlusions, or other transient errors in predictions. The interaction with control systems involves the vehicle's steering, acceleration, and braking, which can be commanded and must comply with the detected road boundaries for safety.

5 Conclusion and Future Work

This study collected data from snow-covered roads in Tromsø, Norway, by recording videos in diverse environments, including the city center, residential areas, and highways. A custom Python script was used to extract frames from these videos at one frame per second, generating a dataset of snowy road conditions with varying textures, lighting, and occlusions. The dataset was preprocessed using SimCLR to enhance detection capabilities. This allowed the model to learn robust feature representations from unlabeled arctic road images before fine-tuning them with labeled data. Image annotations were created for semantic segmentation to detect drivable paths in snow-covered regions. During training, the YOLOv8 model was optimized with uncertainty-aware detection techniques, incorporating Monte Carlo Dropout and Bayesian YOLO to estimate prediction confidence and reduce false positives caused by snow glare and occlusions. Performance evaluation was conducted using precision, recall, F1-score, mAP@50, and mAP@50–95, as well as analysis of false positive reduction and uncertainty quantification.

Additionally, the impact of hyperparameter tuning (batch size and epochs) was assessed to maximize detection efficiency. The proposed Self-Supervised and Uncertainty-Aware YOLOv8 model achieved state-of-the-art performance, with a precision of 0.99, recall of 1.0, F1-score of 0.99, mAP@50 of 0.99, and mAP@50–95 of 0.97. These results revealed that self-supervised pretraining improved the generalization. At the same time, uncertainty estimation improves robustness against adverse weather conditions by significantly reducing reliance on labeled datasets and improving real-world detection accuracy. This study advances autonomous road detection in Arctic environments, contributing to safer navigation for self-driving vehicles in extreme weather conditions.

In the future, the plan is to collect more data in snowy environments across multiple regions. The data collection for this study was limited to Norway. Future efforts aim to collect data in other countries, including Finland, Sweden, and Denmark, to create a more comprehensive dataset. Efforts will also focus on the improvement of computational power of the proposed model by using techniques such as minimizing the number of layers and weights. Lower computational costs for the models will allow the deployment on edge devices. Techniques such as knowledge distillation, quantization, and pruning will be explored to improve model performance for embedded devices. This will allow recording road video in real-time, sending it to a computer for path detection, and displaying the results live on a car's screen. Such advancements aim to enhance vehicle autonomy in poor weather conditions.

Acknowledgement: Not applicable.

Funding Statement: It is funded by the UiT, Project number: 310239152.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Aqsa Rahim and Fuqing Yuan; methodology, Aqsa Rahim; software, Aqsa Rahim; validation, Aqsa Rahim, Fuqing Yuan, and Javad Barabady; formal analysis, Aqsa Rahim; investigation, Aqsa Rahim; resources, Fuqing Yuan and Javad Barabady; data curation, Aqsa Rahim; writing—original draft preparation, Aqsa Rahim; writing—review and editing, Aqsa Rahim, Fuqing Yuan and Javad Barabady; visualization, Aqsa Rahim; supervision, Fuqing Yuan and Javad Barabady; project administration, Fuqing Yuan and Javad Barabady; funding acquisition, Fuqing Yuan and Javad Barabady. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Vachmanus S, Ravankar AA, Emaru T, Kobayashi Y. An evaluation of rgb-thermal image segmentation⁵⁴⁶ for snowy road environment. In: 2021 IEEE International Conference on Mechatronics and Automation (ICMA); 2021 Aug 8–11; Takamatsu, Japan. p. 224–30.
2. Lei Y, Emaru T, Ravankar AA, Kobayashi Y, Wang S. Semantic image segmentation on snow driving scenarios. In: 2020 IEEE International Conference on Mechatronics and Automation (ICMA); 2020 Oct 13–16; Beijing, China. p. 1094–100.
3. Vachmanus S, Ravankar AA, Emaru T, Kobayashi Y. Semantic segmentation for road surface detection in snowy environment. In: 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE); 2020 Sep 23–26; Chiang Mai, Thailand. p. 1381–6.
4. Zang S, Ding M, Smith D, Tyler P, Rakotoarivelo T, Kaafar MA. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Veh Technol Mag.* 2019;14(2):103–11. doi:10.1109/MVT.2019.2892497.
5. Vachmanus S, Ravankar AA, Emaru T, Kobayashi Y. Multi-modal sensor fusion-based semantic segmentation for snow driving scenarios. *IEEE Sens J.* 2021;21(15):16839–51. doi:10.1109/JSEN.2021.3077029.
6. Aldibaja M, Sukanuma N, Yoneda K. Improving localization accuracy for autonomous driving in snow-rain environments. In: 2016 IEEE/SICE International Symposium on System Integration (SII); 2016 Dec 13–15; Sapporo, Japan. p. 212–7.
7. Aldibaja M, Sukanuma N, Yoneda K. Robust intensity-based localization method for autonomous driving on snow-wet road surface. *IEEE Trans Ind Inform.* 2017;13(5):2369–78. doi:10.1109/tii.2017.2713836.
8. Sharma T, Debaque B, Duclos N, Chehri A, Kinder B, Fortier P. Deep learning-based object detection and scene perception under bad weather conditions. *Electronics.* 2022;11(4):563. doi:10.3390/electronics11040563.
9. Yinka AO, Ngwira SM, Tranos Z, Sengar PS. Performance of drivable path detection system of autonomous robots in rain and snow scenario. In: International Conference on Signal Processing and Integrated Networks (SPIN); 2014 Feb 20–21; 2014; Noida, India. p. 679–84.
10. Zhang M, Zhang Y, Zhang L, Liu C, DeepRoad Khurshid S. GAN-based metamorphic testing and input validation framework for autonomous driving systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering; 2018 Sep 3–7; Montpellier, France. p. 132–42.
11. Ryan M. The future of transportation: ethical, legal, social and economic impacts of self-driving vehicles in the year 2025. *Sci Eng Ethics.* 2020;26(3):1185–208. doi:10.1007/s11948-019-00130-2.
12. Hopkins D, Schwanen T. Talking about automated vehicles: what do levels of automation do? *Technol Soc.* 2021;64:101488. doi:10.1016/j.techsoc.2020.101488.

13. Tian Y, Gelernter J, Wang X, Chen W, Gao J, Zhang Y, et al. Lane marking detection via deep convolutional neural network. *Neurocomputing*. 2018;280:46–55. doi:10.1016/j.neucom.2017.09.098.
14. Gupta A, Anpalagan A, Guan L, Khwaja AS. Deep learning for object detection and scene perception in self-driving cars: survey, challenges, and open issues. *Array*. 2021;10:100057. doi:10.1016/j.array.2021.100057.
15. De La Torre G, Rad P, Choo KKR. Driverless vehicle security: challenges and future research opportunities. *Future Gener Comput Syst*. 2020;108:1092–111. doi:10.1016/j.future.2017.12.041.
16. Yu L, Shao X, Wei Y, Zhou K. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning. *Sensors*. 2018;18(9):2905. doi:10.3390/s18092905.
17. Parmar Y, Natarajan S, Sobha G. Deeprange: deep-learning-based object detection and ranging in autonomous driving. *IET Intelligent Transp Syst*. 2019;13(8):1256–64. doi:10.1049/iet-its.2018.5144.
18. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput*. 2019;31(7):1235–70. doi:10.1162/neco_a_01199.
19. Fu J, Liu J, Li Y, Bao Y, Yan W, Fang Z, et al. Contextual deconvolution network for semantic segmentation. *Pattern Recognit*. 2020;101:107152. doi:10.1016/j.patcog.2019.107152.
20. Herrmann A, Brenner W, Stadler R. *Autonomous driving: how the driverless revolution will change the world*. Bingley, UK: Emerald Publishing Limited; 2018.
21. Khosravian A, Amirkhani A, Masih-Tehrani M, Yazdaniyoo A. Multi-domain autonomous driving dataset: towards enhancing the generalization of the convolutional neural networks in new environments. *IET Image Process*. 2023;17(4):1253–66. doi:10.1049/ipr2.12710.
22. Vachmanus S, Emaru T, Ravankar AA, Kobayashi Y. Road detection in snowy forest environment using RGB camera. *arXiv:221208511*. 2022.
23. Broggi A, Cantoni V, Vallone U, Fascioli A. Snowcat track detection in snowy environments. In: *Proceedings of the 2001 IEEE Intelligent Vehicle Symposium; 2001 Aug 25–29; Oakland, CA, USA*. p. 19–24.
24. Ushma A, Scholar M, Shanavas P. Object detection in image processing using edge detection techniques. *IOSR J Eng*. 2014;4:10–3. doi:10.9790/3021-04311013.
25. Florinabel DJ. Real-time image processing method to implement object detection and classification for remote sensing images. *Earth Sci Inform*. 2020;13(4):1065–77. doi:10.1007/s12145-020-00486-1.
26. Balaji S, Karthikeyan S. A survey on moving object tracking using image processing. In: *2017 11th International Conference on Intelligent Systems and Control (ISCO); 2017 Jan 5–6; Coimbatore, India*. p. 469–74.
27. Karaa M, Ghazzai H, Sboui L. A dataset annotation system for snowy weather road surface classification. *IEEE Open J Syst Eng*. 2024;2:71–82. doi:10.1109/OJSE.2024.3391326.
28. Jiang J, Xu G, Wang H, Yang Z, Sun B, Guan C, et al. High-accuracy road surface condition detection through multi-sensor information fusion based on WOA-BP neural network. *Sens Actuators A Phys*. 2024;378:115829. doi:10.1016/j.sna.2024.115829.
29. Tahir NUA, Zhang Z, Asim M, Iftikhar S, El-Latif AAA. PVDM-YOLOv8l: a solution for reliable pedestrian and vehicle detection in autonomous vehicles under adverse weather conditions. *Multimed Tools Appl*. 2024;2024:1–26. doi:10.1007/s11042-024-20219-6.
30. Feng X, Peng T, Qiao N, Li H, Chen Q, Zhang R, et al. ADWNet: an improved detector based on YOLOv8 for application in adverse weather for autonomous driving. *IET Intell Transp Syst*. 2024;18(10):1962–79. doi:10.1049/itr2.12566.