

28th ITS World Congress, Los Angeles, September 18-22, 2022

Tire track identification: Application of U-net deep learning model for drivable region detection in snow occluded conditions

Parth Kadav^{1*}, Nicholas Goberville¹, Farhang Motallebiaraghi¹, Alvis Fong¹ and Zachary D. Asher¹

1. Western Michigan University, Kalamazoo, MI, parth.kadav@wmich.edu

Abstract

Advanced Driver Assistance Systems (ADAS) typically utilize cameras to provide limited automation features to improve driver safety. ADAS utilizes computer vision (CV) to extract vehicle surrounding information. However, when the vehicle is operating in bad weather (e.g., obstructed lane lines), ADAS products fail. We have developed a new technique to detect tire tracks which was evaluated in conditions of variable snow coverage and lane line occlusion. Previously we focused on using basic machine learning (ML). We expanded this to a convolutional neural network (CNN). A custom dataset was collected using an instrumented automated research vehicle. The CNN model had an intersection over union (IoU) score of 89% in detecting tire tracks and outperformed the traditional ML model on key metrics (precision, recall, and more). Overall we have demonstrated that this method works as an end-to-end pipeline to detect tire tracks and expand the operational design domain of ADAS.

Keywords:

AUTONOMOUS VEHICLES, ADAS, MACHINE LEARNING

Introduction

Advanced Driver Assistance Systems (ADAS) such as Forward Collision Warning (FCW), Automatic Emergency Braking (AEB), Lane Departure Warning (LDW), Lane-keeping Assistance (LKA), blind-spot warning assistance, and many more have the potential to prevent or mitigate approximately 40% of all passenger vehicle crashes [1]. Because human error causes the majority of road accidents, ADAS was created to automate and improve aspects of the driving experience in order to increase safety and safe driving practices. If the vehicle crosses the lane and no turn signals or steering movements are detected, lane-keeping systems detect reflective lane markings in front of the car and inform the driver via various sorts of audio, tactile, and/or visual cues [2]. From the 1853 driver injury crashes studied in [3,4], it was discovered that LDW/LKA systems were able to reduce head-on and single-vehicle crashes on roads with higher speed limits (45-75mph) and visible lane markings by 53%. The greatest benefit of such systems, according to [5], is at lower operating speeds (5-20 mph), where between 11 and 23% of drift-out-of-lane incidents and 13 to 22% of seriously to fatally wounded drivers could have been avoided if the technology was used. FCW and AEB alone cut front-to-rear collisions by nearly half [6]. By 2023, the market for ADAS is expected to be worth more than \$30 billion [7] where ADAS will not only be confined to safety but will also help increase vehicle efficiency [8-13].

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



Despite these successes of ADAS technology, there is a glaring unresolved problem: inclement weather. During 2007–16, weather-related vehicular crashes accounted for 21% (1,235,145) of all reported crashes annually resulting in 16% (5,376) of crash fatalities and 19% (418,005) of crash injuries throughout the United States [14]. Fundamentally, adverse weather conditions can cause impairment to situational awareness and inhibitions to vehicular maneuverability which can occur in a variety of ways depending on the type of adverse weather [14].

Developing techniques for the operation of ADAS in inclement weather is a current research challenge. Because there are significant ramifications for safety as outlined above, the initial goal is to recognize and classify road lanes during inclement weather in order to aid in the location of both the ego vehicle and other vehicles [15]. The challenge is that inclement weather such as heavy rain, snow, or fog lowers the maximum range and signal quality for ADAS sensors such as cameras and it occludes the high contrast lane markers [15]. This is a well-documented problem and has been demonstrated in cameras and lidars in particular [16]. A specific instance of this issue can be found in [3], where it is stated that LDW/LKA was only able to reduce head-on and single-vehicle crashes on roads with operating speeds of 45-75 mph by 53% if the roads had visible road markings and specifically "the road surface was not covered by ice or snow." New sensor technologies are getting better in these performance areas but are still far from addressing the issue of reliable ADAS operation in inclement weather [8]. For now, to achieve a feasible research scope for this paper, we will focus on only snowy weather.

There are just a few key studies that address the issue of reliable ADAS operation in snowy weather. The first study developed a custom snowy weather dataset and determined the driveable region through semantic segmentation [17]. When evaluated on a non-snow dataset, the model had a mean Intersection over Union (mIoU) of 80%, when trained on a snowy dataset it dropped to 19% and when both models were combined, it provided a mIoU of 83.3%. However, the model still must be improved and made more robust because it considers the entire road rather than just the Region of Interest (ROI), which can be computationally expensive. The second study employed a CNN model with a specified architecture and used sensor fusion between the camera, lidar, and radar [18]. The results showed that there was an increase in driveable region detection (mIoU of 81.35%) and non-driveable region detection (mIoU of 93.85%) after fusing the information from various sensors and testing it on the dataset. This is an improvement, but it comes with drawbacks, the most significant of which is that the method requires additional sensors, which increases the cost and computational power required. Additionally, this method, like the first study, examines the entire driveable region, rather than just a ROI [18]. In the third relevant study, a method to improve the detection in adverse weather conditions using "You Only Look Once" (YOLO) was developed by merging it with a CNN and the Federated Learning (FL) framework [19]. This was tested on the Canadian Adverse Driving Conditions (CADC) dataset. The method resulted in the average test accuracy of the model, gossip, and centralized approaches which are the three different methods they use in their study to be 90.4–95.2%, 82.4–88.1%, and 71.4–76.16%, respectively. The FL method, which utilizes an edge server, is the foundation for this model. After training a global YOLO CNN model on a publicly available dataset, the edge server sends the initial parameters to the AVs. These parameters are then used by the AVs to locally train the model on their own dataset. The number of AVs collecting data, the connection between the edge server and each vehicle, and the computational power in each vehicle all contribute to the FL method's training time. Furthermore, the vehicle chassis has been equipped with eight cameras, increasing the cost [19]. All the above studies provide methods for improving the detection of objects and regions in the entire driveable space and not necessarily the lane information, these studies are both computationally and

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



monetarily costly and rely on multiple sensors. None of these studies demonstrates high accuracy driveable region detection for snow-covered roads using a single camera sensor that is implementable in modern ADAS products.

To address this research gap, we are utilizing a computationally light, cost-effective, and high-accuracy method of extracting driveable region information using a single camera which is a ubiquitous automotive sensor [16,17]. ML techniques such as CNN have established themselves as a dominating methodology in modern computer vision algorithms and applications, as well as in segmentation research. Based on our previous study for detecting tire tracks in snowy weather conditions [20], the ML model required a lot of image pre-processing and feature engineering, which is addressed in this study by using a CNN. In this study, both supervised ML semantic segmentation models and CNNs were developed. These methods were then compared for detecting tire tracks in the snow. The paper addresses the following novel topics:

- 1) Custom data acquisition method for tire track data collection and labeling
- 2) Snow tire track image preprocessing and feature extraction
- 3) Tire track identification CNN architecture
- 4) CNN and ML model performance comparison for snow tire track identification

Methodology

In this section, we will first discuss the methods we used to collect and prepare the data. The data that has been processed is then used to develop models.

Data Collection

The route we chose consisted of two-lane arterial roads in Kalamazoo that had the road characteristics we were looking for. This drive cycle replicated roads that are rarely cleaned after snowfall and are maintained much less frequently than highways and other multi-lane roads. We collected the data during the 2020 winter season. The lanes had snow occlusion with distinct tire track patterns, with the tire tracks visible to show the tarmac below and the lane line markings covered in snow. Data was collected using our Energy Efficient and Autonomous Vehicles (EEAV) Lab's instrumented automation development platform shown in Fig. 1. This development platform is built upon a drive-by-wire capable 2019 Kia Niro and the relevant sensor for this study is a forward-facing ZED 2 RGB stereo camera made by Stereolabs. The ZED 2 has a 120-degree field of view wide-angle lens that captures images and videos using stereo vision, although only one of the lenses was used for this study. The camera was set to record video at a frame rate of 29 frames per second with a resolution of 1280 x 720 pixels.

The ZED 2 was connected to the in-vehicle computer and data was collected as *.mp4 files over arterial roads with visible tire tracks and occluded lane lines. From these video files, a total of 1,500 individual frames were extracted for ML training. Fig. 1 shows an overview of this data collection process. The 1500 frames of images were divided into three batches, each with 500 images. Different parameters such as exposure, resolution, and occlusion were assessed in the images. Clear tire tracks with distinct tarmac and snow boundaries were chosen from the images.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



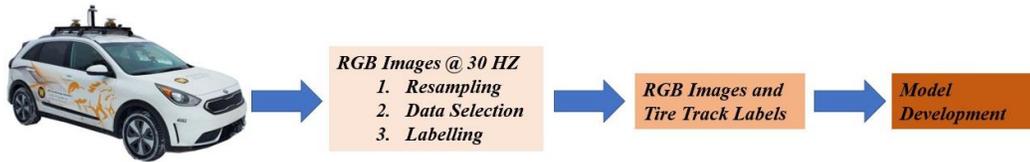


Figure 1. Flow diagram for the data collection, resampling of the data, extracting 1500 RGB Images and corresponding Tire Track Labels, and labeling of data

Data Preparation

The images that were previously segregated into different batches of frames are then used for labeling. Every frame's tire tracks were labeled by hand using an open-source, online tool known as the Computer Vision Annotation Tool (CVAT). Images were uploaded in respective batches and the labeled dataset of each batch was exported with their corresponding raw images using the format: CVAT for images 1.1. This process was again repeated for all the batches.

Each exported dataset contained the raw images and an Extensive Markup Language (XML) file which contained the attributes for the labels, such as the position of the tire-track with their corresponding pixel location on the image, image file name, and their assigned tags (tire-track, road, road-edge boundary). This process can be updated and more labels can be added according to the use case. The exported labels were then further assessed for post-processing and training the ML and CNN model. The overall data preparation pipeline is described in the next section

Model Development Pipeline

To develop the ML model we must preprocess the data and then perform feature extraction. The process of converting raw data into numerical features that the model can process while preserving information from the original data set is referred to as feature extraction. This is done because it produces significantly better results than applying machine learning to the raw dataset directly.

To improve feature detection and reduce the computational cost, images were masked with a ROI that includes just the road surface and not the entire frame. As stated in [17,18], it is seen that different methods are used to detect road surfaces with high accuracy with an array of sensors. We implemented these road surface detections by using a static ROI in which the pixels inside the ROI are the road surface and every other pixel outside the ROI is considered to be the background. Fig. 2. shows the process to extract the masked images for the ROI.

The raw images were first resized to the desired shape from their original size of 1280 x 720. In our case, we chose the images to be of shape 256 x 256. The road ROI mask was obtained from the raw image to reduce the number of pixels used for training and reduce the computational cost.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



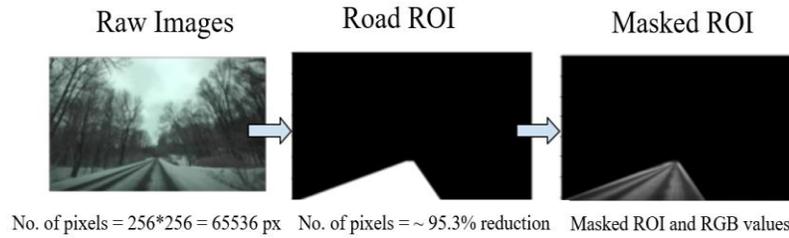


Figure 2. The feature extraction procedure, which begins by extracting only the frames within the ROI and then extracts the features from those pixels.

The Road ROI only consists of 3099 pixels which are only ~ 5% of the total pixels in the raw image. The ROI mask was then fused with the raw image to obtain all the pixels within the ROI. This will in turn be the input to the model. The different features extracted from the masked images include the red, green, blue, grayscale pixel values, and the pixel X, Y locations as done in the previous study [20].

The different feature vectors shown in Table I are grouped into different sets and are individually selected to be the final input to the model. The results from these will show the features that contribute the most to the model and yield the highest performance. The model was split into a 55 - 45% train test split. The entire model was trained using a single input array X having the shape = ((m*p), n) where m is the total number of images, p is the number of pixels in the ROI of each image (3099 pixels for the 256x256 sized images), and n is the number of feature vectors in the array. An overview of this process is shown in Figure 3.

Table I. Feature Set Properties

Feature set	Included Feature Vector	Train Array Shape (m = 1200)	Test Array Shape (m = 300)
0	Gray	(3718800,1)	(929700,1)
1	Gray X loc, Y loc	(3718800,1)	(929700,1)
2	Red, Green, Blue	(3718800,3)	(929700,3)
3	Red, Green, Blue, X loc, Y loc	(3718800,5)	(929700,5)

Machine Learning Implementation and Evaluation

As seen in our previous study [20], we trained various ML models from the input features and their respective labels. The input feature array X and label vector y were extracted from the image preprocessing and feature extraction block and then fed as inputs to the ML model. Six different models were evaluated to determine the feature set/model combination for the highest performance metrics. Models that were evaluated include K - Nearest Neighbor (KNN), Naive-Bayes, Decision Trees (Dtrees), Random Forest, Linear Regression, and Logistic Regression. These models were chosen for their characteristics and capabilities in commuting binary classification [21–23].

The outputs from the predicted model y_{pred} were compared with ground truth for evaluation. The metrics used for evaluation were the intersection over union (IoU), mIoU, pixel prediction accuracy, precision, recall, F1 score, and frame per second (FPS). These metrics were evaluated based on the ability to draw strong conclusions from the model's performance [21]. Below are the equations demonstrating these

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



calculations as well as the four corners of a confusion matrix, which define the true positives, true negatives, false positives, and false negatives.

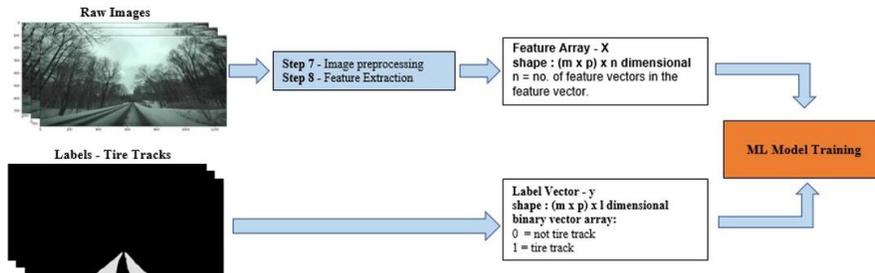


Figure 3. A flow diagram for training the ML model. The features recovered from the raw photos are stored in the input feature array X, and the label vector y contains the pixel status as either tire track (1) or non-tire track (0).

- True Positive (TP): no. of pixels classified correctly as in a tire track
- False Positive (FP): no. of pixels classified incorrectly as in a tire track
- True Negative (TN): no. of pixels classified correctly as not in a tire track
- False Negative (FN): no. of pixels classified incorrectly as not in a tire track

$$\text{Accuracy} = \frac{\text{total correct predictions}}{\text{all predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{IoU (Jaccard Index)} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2)$$

$$\text{mIoU} = 1/n * \sum_{i=1}^n \frac{\text{intersection}}{\text{union}} = 1/n * \sum_{i=1}^n \frac{TP_i}{TP_i + FP_i + FN_i}, \text{ where } n = \# \text{ of classes} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{F1 Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

Following the creation of the ML models, we discovered that this method, in our instance, necessitates a significant amount of feature engineering or image pre-processing. The raw images are cropped and turned to grayscale. Similarly, the segmentation masks are cropped to generate the ROI mask, and the X and Y pixel locations from the segmentation masks are saved to feed into the model, as explained in our image pre-processing and feature extraction sections. Furthermore, the ROI is static, which means it is fixed for each image and does not account for changing road curvature. overall this process necessitates a substantial level of effort, which CNN will address.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



Convolutional Neural Network Implementation and Evaluation

Deep learning has been shown to perform significantly better on a wide range of tasks, including image recognition, natural language processing, and speech recognition. Deep networks, when compared to traditional ML algorithms, scale effectively with data, do not require feature engineering, are adaptable and transferable, and perform better on larger datasets with unbalanced classes [24].

CNNs are a type of deep neural network whose architecture is designed to automatically conduct feature extraction thus eliminating this step [25]. CNN's create feature maps by performing convolutions to the input layers, which are then passed to the next layer. In contrast to basic ML techniques, CNNs can extract useful features from raw data, eliminating the need for manual image processing [26,27].

As previously stated, our ML model required feature engineering and did not function as an end-to-end pipeline for tire track identification. To make this process easier and to improve the overall accuracy we have implemented a CNN.

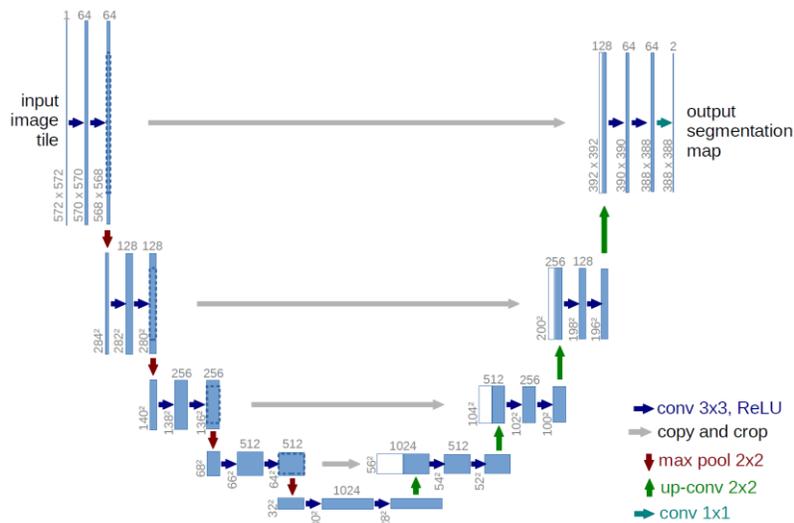


Figure 4. U-network architecture (example for 32x32 pixels in the lowest resolution) [28]. A multi-channel feature map is represented by each blue box. The number of channels is indicated on the box's top. The x-y size is indicated at the box's lower-left edge. White boxes represent feature maps that have been copied. The arrows represent the various operations.

Architecture

The U-net architecture has demonstrated excellent performance in computer vision segmentation [29]. CNN's basic premise is to learn an image's feature mapping and use it to create more sophisticated feature maps. This works well in classification problems since the image is turned into a vector, which is then classified. In image segmentation, however, we must not only transform a feature map into a vector but also reconstruct an image from this vector [29]. U-net architecture was developed specifically for this problem and was first introduced in a medical application [29]. Its structure is depicted in Figure 4.

The U-net architecture learns the image's feature maps while converting it to a vector, and the same mapping is used to convert it back to an image. The left side of the U-net architecture is known as the contracting

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



path, while the right side is known as the expansive path. The number of feature channels/filters doubles after each downsampling block to learn more complicated structures from the previous layer's output, while the image size decreases. This path consists of numerous contraction blocks. Each block takes an input and applies it to a 3×3 convolutional layer with a rectified linear unit (ReLU) activation function. The padding is set to 'same' which is followed by a 2×2 max-pooling layer for downsampling. We start off with 32 feature channels and double them with every contraction block until we reach 512 feature channels, which is when we move onto the expansive path. Each block in the expansive path (shown on the right side of the image) is composed of two 3×3 convolution layers and one 2×2 up-sampling or up-convolution layer with a ReLU activation function and padding set to 'same'. The input is appended by the feature maps of the matching contraction layer with each block in the up-convolution, which is known as concatenating and is indicated by the gray arrow between the two layers. The number of feature channels is halved with each block in this layer. A 1×1 convolution layer is applied in the final layer, with the number of feature maps equaling the number of required classes/segments. In addition, in both the expansive and contraction paths, we add a dropout layer between each convolution layer. This reduces model overfitting by randomly shutting down the necessary number of neurons in that layer [30,31].

Metrics

As mentioned in the ML section, the different metrics are shown which are used to evaluate the model's performance. From equation (1), the accuracy shows the fraction of predictions our model got right. But accuracy alone doesn't tell the complete story when working with a class-imbalanced dataset [32] In our dataset, there is a great amount of imbalance between the tire tracks and the background, which is why accuracy is not a good metric for evaluation. This means that the inaccuracy of minority classes is overshadowed by the accuracy of the majority classes when compared to pixel-wise accuracy. IoU, which is also known as Jaccard Index is substantially more suggestive of success for segmentation tasks, especially when the input data is significantly sparse. When training labels contain 80-90% background and only a tiny fraction of positive labels, a basic measure like accuracy can score up to 80-90% by categorizing everything as background. Because IoU is unconcerned about true negatives, even with extremely sparse data, this naive solution will never arise. IoU computes the overlapping region for the true and anticipated labels by comparing the similarity of finite sample sets A, B as the IoU [33]. As stated in equation (7), T stands for the true label image and P stands for the prediction of the output image. This is used as a metric, providing us with a more accurate means of measuring IoU in our model's segmentation region.

$$\text{Jaccard Index (IoU)} = \frac{|T \cap P| \text{ (Area of Overlap)}}{|T \cup P| \text{ (Area of Union)}} \quad (7)$$

Loss Function

We use two loss functions in our model. Loss functions are used to reduce loss and the number of incorrect predictions made. The loss function Binary Cross-Entropy (BCE) is used in binary classification [34] The BCE function is:

$$\text{BCE} = -t_l \log(s_l) - (1 - t_l) \log(1 - s_l) \quad (8)$$

where t_l denotes the label/segmentation mask and s_l denotes the label's predicted probability across all images. We use BCE because our model needs to predict the segmentation mask of the tire track.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



The Jaccard Loss, which is equal to the negative *Jaccard Index* from equation (7), is the second loss function used. A higher IoU value indicates that there is more overlap between the true label and the predicted label, but the loss function is concerned with minimizing IoU, which is why we use a negative Jaccard Index as the loss function to reduce loss.

Convolutional Neural Network Model Training

The model was trained using the input images and their associated segmentation masks. We used google colab pro's cloud GPU to train our model. The ML model's input feature vector array was used with feature set 2 (RGB images). The shape of the training array is $(m \times n \times p \times l) = (1300, 256, 256, 3)$ where m is the number of images in the training set, n is the image height, p is the image width and l are the number of channels in the image. In our case, we resize the images to the desired size in feature extraction (6 b.) and use feature set 2, which uses the image's RGB values. We can use the raw RGB images without any pre-processing because no image pre-processing is required.

We consider stochastic gradient descent (SGD) and Adaptive moment estimation (Adam) for our optimizers. Optimizers update the model in response to the loss function's output, attempting to minimize the loss function's output. SGD begins with a random initial value and continues to take steps with a learning rate to converge to the minima. SGDs are simple to implement and fast for problems with a large number of training examples but have a disadvantage in that they necessitate extensive parameter tuning [35] Unlike stochastic gradient descent, Adam is computationally efficient and is better suited to problems with very noisy/or sparse gradients because it computes adaptive learning rates [36] For image segmentation, Adam is thought to be a very powerful loss function [37], which is why we chose Adam as our optimizer. BCE and Jaccard loss are two different loss functions that we use which is covered in section 8.c. The batch size is set to 16 and the model is run for 25 epochs with an early callback to save the model at the best epoch for the validation loss. For testing, training, and validation, the predicted images are thresholded, so anything above 50% is saved as a correct prediction. There are 7,760,097 trainable parameters in total.

Convolutional Neural Network Model Evaluation

In contrast to our ML models, the model's predicted output was an image. The predicted segmentation masks were then assessed using a variety of metrics. We test the model for IoU, precision, recall, and F1 score, as mentioned in the metrics section. Equations (1-6) show how the confusion matrix is used to perform these calculations. Figure 5 shows the outputs from CNN.



Figure 5. CNN output (The raw image is on the left, the labeled segmentation mask is in the middle, and the predicted segmentation mask from the CNN is on the right)

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



Results

When we run the model with the loss function set to BCE and Adam as the optimizer, we see that the model's accuracy increases to ~98%. However, as discussed in the metrics section, accuracy is not a good metric for datasets with a lot of class imbalance, which is why it produces such high values. Therefore we must also test the IoU.

Figure 6a shows that the model with Jaccard loss function has an IoU score of 93% and a validation IoU of 88%. Figure 6b shows the IoU of the model with the loss function set to BCE is 89%, and the validation IoU is 84%. This means that, when compared to BCE, the Jaccard loss function does a better job of finding the intersection/overlapping region for the segmentation masks between the true and predicted. Even though this is true, BCE is still regarded as a good performer because it is only 3% less accurate. The two models have an average frame rate of nearly 350 FPS.

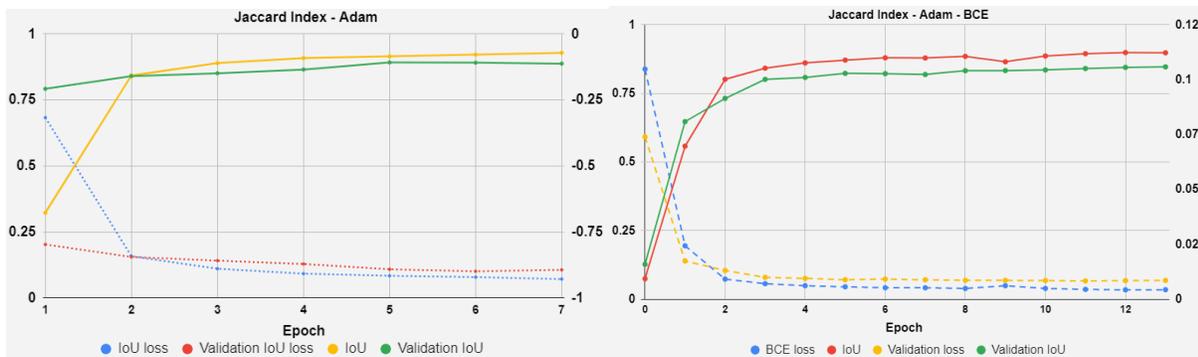


Figure 6a. Jaccard loss function, Jaccard Index (IoU) as the metric

Figure 6b. BCE loss function, Jaccard Index (IoU) as the metric

The results of the best CNN and ML models are summarized in Table II. Dtruss with feature set 1 was found to be the model with the best performance in our prior study. We compare the metrics for that model to our CNN model with feature set 2 since we don't have to perform any preprocessing in our case.

TABLE II. CNN AND ML METRICS

Model	Feature set	Accuracy	Precision	Recall	F1 Score	FPS
CNN	2	0.98	0.96	0.95	0.96	350.32
Dtruss	1	0.90	0.905	0.911	0.908	1084.1

We observe that the CNN model performs better than the ML model without any image preprocessing on metrics like accuracy, precision, recall, and F1 score, shown in Figure 7.

Limitations of this study include comparing metrics such as mIoU with the previous ML models. The ML model with Dtruss and feature set 1 obtains a mIoU of 83%, whereas the CNN achieves a mIoU of 65%. This could imply that the ML model is more accurate at predicting tire tracks, but it is not the whole story.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



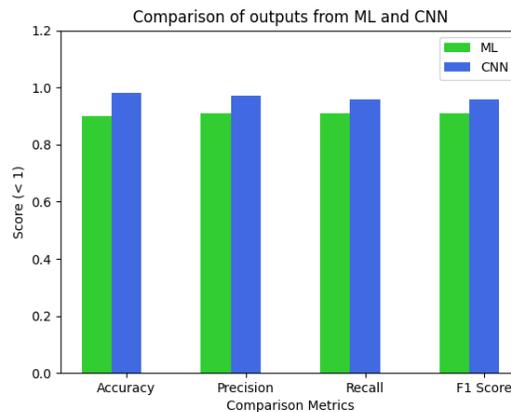


Figure 7. Precision, Accuracy, Recall and F1 score metric comparison between CNN and Dtres.

A static ROI for the ML model was employed, which means that the ML model only receives a portion of the raw image and the segmentation masks. The mIoU calculates the IoU for each class before averaging the results across all of them. Because we just feed a section of the image into the ML model rather than the complete image, it performs better at detecting these tire tracks only in that precise region, which implies the model will not do well if the road geometry shifts or if the model is tested on the entire image. The CNN, on the other hand, does not require a ROI but instead takes in the full image as input, lowering the mIoU because it is no longer simply looking at the ROI but the complete image. Another explanation for CNN's lower mIoU is the significant class imbalance (more background pixels and fewer tire track pixels), as well as the fact that deep neural networks require more training data than ML models which means to improve the mIoU we will need to train the model on larger datasets. Another way to attain a higher mIoU would be to crop the ROI for images and segmentation masks in the same way as our ML models, and then use that as the input to the CNN. However, this would necessitate preprocessing and feature engineering, which is one of the drawbacks from the ML models addressed in this paper.

Conclusion

This study addresses the research gap of driveable region detection for snow-covered roads using a single camera sensor that is implementable in modern ADAS products. We proposed a new method for extracting the drivable region for snowy road conditions when the lane lines are occluded by instead focusing on identifying tire tracks. First data was collected on our instrumented vehicle and then the data was processed by extracting the frames from the videos, segmenting them into batches, and labeling them with CVAT. We have showcased how this information was used in the model development process. Using just the raw image and no image pre-processing or feature extraction, we evaluated a U-net-based CNN for IoU, Accuracy, Feature set, Recall, F1 score, and FPS. The IoU score for the model with the Jaccard loss function was 93%. The model had an accuracy of 98%, a 95% recall, a 96% precision, and a 96% F1 score. Furthermore, we found a significant improvement in these metrics when compared to the ML model from the previous study. By feeding in the raw image and obtaining the predicted tire tracks, this method offers a full end-to-end solution for detecting drivable regions in snowy road conditions.

Overall, this study demonstrates that drivable region detection in inclement weather is feasible using current technology in a single camera. The results can be improved by improving image processing and tuning the

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



CNN. Beyond this study, there are many other research gaps in inclement weather automation that need to be addressed to combat the significant loss of life that comes from these scenarios.

References

1. Svancara, A.M., Horrey, W.J., Tefft, B., and Benson, A., “Potential Reduction in Crashes, Injuries and Deaths from Large-Scale Deployment of Advanced Driver Assistance Systems,” *AAA Foundation for Traffic Safety*, 2018.
2. Varghese, J.Z. and Boone, R.G., “Overview of autonomous vehicle sensors and systems,” *International Conference on*, 2015.
3. Sternlund, S., Strandroth, J., Rizzi, M., Lie, A., and Tingvall, C., “The effectiveness of lane departure warning systems—A reduction in real-world passenger car injury crashes,” *Traffic Inj. Prev.* 18(2):225–229, 2017.
4. Kusano, K.D., Gabler, H., and Gorman, T.I., Fleetwide Safety Benefits of Production Forward Collision and Lane Departure Warning Systems, *SAE International Journal of Passenger Cars - Mechanical Systems* 7(2):514–527, 2014, doi:10.4271/2014-01-0166.
5. Kusano, K.D. and Gabler, H.C., “Comparison of Expected Crash and Injury Reduction from Production Forward Collision and Lane Departure Warning Systems,” *Traffic Inj. Prev.* 16 Suppl 2:S109–14, 2015.
6. Real-world benefits of Crash Avoidance technologies, Insurance Institute for Highway Safety & Highway Loss Data Institute, 2020.
7. Advanced driver assistance systems: global revenue growth 2020-2023, <https://www.statista.com/statistics/442726/global-revenue-growth-trend-of-advanced-driver-assistance-systems/>, Feb. 2022.
8. Jiménez, F., Naranjo, J.E., Anaya, J.J., García, F., Ponz, A., and Armingol, J.M., “Advanced Driver Assistance System for Road Environments to Improve Safety and Efficiency,” *Transportation Research Procedia* 14:2245–2254, 2016.
9. Asher, Z.D., Tunnell, J.A., Baker, D.A., Fitzgerald, R.J., Banaei-Kashani, F., Pasricha, S., and Bradley, T.H., “Enabling Prediction for Optimal Fuel Economy Vehicle Control,” SAE Technical Paper, 2018.
10. Motallebiaraghi, F., Yao, K., Rabinowitz, A., Hoehne, C., Garikapati, V., Holden, J., Wood, E., Chen, S., Asher, Z., and Bradley, T., “Mobility energy productivity evaluation of prediction-based vehicle powertrain control combined with optimal traffic management,” 2022-01-0141, SAE Technical Paper, 2022.
11. Kadav, P. and Asher, Z.D., “Improving the Range of Electric Vehicles,” *2019 Electric Vehicles International Conference (EV)*, 1–5, 2019.
12. Rabinowitz, A., Araghi, F.M., Gaikwad, T., Asher, Z.D., and Bradley, T.H., “Development and Evaluation of Velocity Predictive Optimal Energy Management Strategies in Intelligent and Connected Hybrid Electric Vehicles,” *Energies* 14(18):5713, 2021.
13. Mahmoud, Y.H., Brown, N.E., Motallebiaraghi, F., Koelling, M., Meyer, R., Asher, Z.D., Dontchev, A., and Kolmanovsky, I., “Autonomous Eco-Driving with Traffic Light and Lead Vehicle Constraints: An Application of Best Constrained Interpolation,” *IFAC-PapersOnLine* 54(10):45–50, 2021.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



14. How do weather events impact roads?, https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm, Feb. 2022.
15. Gern, A., Moebus, R., and Franke, U., "Vision-based lane recognition under adverse weather conditions using optical flow," *Intelligent Vehicle Symposium, 2002. IEEE*, 652–657 vol.2, 2002.
16. Brandon, S., "SENSOR FUSION: A COMPARISON OF CAPABILITIES OF HUMAN HIGHLY AUTOMATED."
17. Lei, Y., Emaru, T., Ravankar, A.A., Kobayashi, Y., and Wang, S., Semantic Image Segmentation on Snow Driving Scenarios, *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2020, doi:10.1109/icma49215.2020.9233538.
18. Rawashdeh, N.A., Bos, J.P., and Abu-Alrub, N.J., "Drivable path detection using CNN sensor fusion for autonomous driving in the snow," *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2021*, SPIE: 36–45, 2021.
19. Rjoub, G., Wahab, O.A., Bentahar, J., and Bataineh, A.S., "Improving Autonomous Vehicles Safety in Snow Weather Using Federated YOLO CNN Learning," *Mobile Web and Intelligent Information Systems*, Springer International Publishing: 121–134, 2021.
20. Goberville, N.A., Kadav, P., and Asher, Z.D., "Tire Track Identification: A Method for Drivable Region Detection in Conditions of Snow-Occluded Lane Lines," SAE Technical Paper, 2022.
21. Shetty, S.H., Shetty, S., Singh, C., and Rao, A., Supervised machine learning: Algorithms and applications, *Fundamentals and Methods of Machine and Deep Learning* 1–16, 2022, doi:10.1002/9781119821908.ch1.
22. Chourasiya, S. and Jain, S., A Study Review On Supervised Machine Learning Algorithms, *International Journal of Computer Science and Engineering* 6(8):16–20, 2019, doi:10.14445/23488387/ijcse-v6i8p104.
23. Osisanwo, F.Y., Akinsola, J.E.T., Awodele, O., Hinmikaiye, J.O., Olakanmi, O., and Akinjobi, J., "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)* 48(3):128–138, 2017.
24. Seif, G., "Deep Learning vs Classical Machine Learning," <https://towardsdatascience.com/deep-learning-vs-classical-machine-learning-9a42c6d48aa>, 2018.
25. Long, J., Shelhamer, E., and Darrell, T., "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440, 2015.
26. Albawi, S., Mohammed, T.A., and Al-Zawi, S., "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, 1–6, 2017.
27. Why Convolutional Neural Networks Are The Go-To Models In Deep Learning, <https://analyticsindiamag.com/why-convolutional-neural-networks-are-the-go-to-models-in-deep-learning/>, 2018.
28. Ronneberger, O., Fischer, P., and Brox, T., U-Net: Convolutional Networks for Biomedical Image Segmentation, *arXiv [cs.CV]*, 2015.
29. Sankesara, H., "UNet," <https://towardsdatascience.com/u-net-b229b32b4a71>, 2019.
30. Baldi, P. and Sadowski, P., "Understanding Dropout."

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:



<https://proceedings.neurips.cc/paper/2013/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf>, Feb. 2022.

31. Brownlee, J., "A Gentle Introduction to Dropout for Regularizing Deep Neural Networks," <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>, 2018.
32. Classification: Accuracy, <https://developers.google.com/machine-learning/crash-course/classification/accuracy>, Feb. 2022.
33. Duque-Arias, D., Velasco-Forero, S., Deschaud, J.-E., Goulette, F., Serna, A., Decencière, E., and Marcotegui, B., "On power jaccard losses for semantic segmentation," *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, SCITEPRESS - Science and Technology Publications, ISBN 9789897584886, 2021, doi:10.5220/0010304005610568.
34. Godoy, D., "Understanding binary cross-entropy / log loss: a visual explanation," <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>, 2018.
35. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., and Ng, A.Y., On optimization methods for deep learning, 2011.
36. Kingma, D.P. and Ba, J., Adam: A Method for Stochastic Optimization, *arXiv [cs.LG]*, 2014.
37. Yaqub, M., Jinchao, F., Zia, M.S., Arshid, K., Jia, K., Rehman, Z.U., and Mehmood, A., "State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images," *Brain Sci* 10(7), 2020, doi:10.3390/brainsci10070427.

WWW.ITSAMERICAEVENTS.COM/WORLD-CONGRESS

HOSTED BY:



IN PARTNERSHIP WITH:



BUILT BY:

