2022-01-0083 Published 29 Mar 2022

Tire Track Identification: A Method for Drivable Region Detection in Conditions of Snow-Occluded Lane Lines

Nicholas A. Goberville, Parth Kadav, and Zachary D. Asher Western Michigan University

Citation: 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 2022-01-0083, 2022, doi:10.4271/2022-01-0083.

Received: 10 Nov 2021 Revised: 06 Dec 2021 Accepted: 17 Dec 2021

Abstract

Today's Advanced Driver Assistance Systems (ADAS) predominantly utilize cameras to increase driver and passenger safety. Computer vision, as the enabler of this technology, extracts two key environmental features: the drivable region and surrounding objects (e.g., vehicles, pedestrians, bicycles). Lane lines are the most common characteristic extracted for drivable region detection, which is the core perception task enabling ADAS features such as lane departure warnings, lane-keeping assistance, and lane-centering. However, when subject to adverse weather conditions (e.g., occluded lane lines) the lane line detection algorithms are no longer operational. This prevents the ADAS feature from providing the benefit of increased safety to the driver. The performance of one of the leading computer vision system providers was tested in conditions of variable snow coverage and lane line occlusion during the 2020-2021 winter in Kalamazoo, Michigan. The results show that this computer vision system was only able to provide high confidence detections in less than 1% of all frames recorded. This is an alarming result, as 21% of all crashes in the U.S. are weather-related. To increase the capabilities of ADAS when snow-occlusions are present, a tire track identification system was developed by comparing various supervised machine learning models. A custom dataset was collected using the Energy Efficient and Autonomous Vehicles lab's research platform from Western Michigan University. A data preparation pipeline was implemented to label tire tracks and train the machine learning models. The best model achieved high confidence detections of tire tracks in 83% of all frames of which tire tracks were present, an 82% increase in detections than the leading computer vision system provider.

Introduction

Advanced Driver Assistance Systems (ADAS) features such as lane centering (LC), lane-keeping assist (LKA), and lane departure warning (LDW) have proven to reduce sideswiping collisions by 11% and lane-change crashes by 21% [1, 2, 3]. The ultimate vision for ADAS is to incrementally increase the performance of these features from the current state (SAE level 2, [4]) to higher automation states (SAE levels 4/5) to reach potential collision mitigation of 94% of all accidents [5]. However, due to the weather-limited operational design domain (ODD), the path to level 4 ADAS features is not clear. The perception system's ODD of the ADAS stack needs to include weather if this vision is to be realized [6, 7, 8, 9].

Nearly 1 in 5 accidents in the U.S. can be attributed to adverse weather [10]. Of these weather incidents, 24% occur on snowy or icy pavement and 15% occur during heavy snowfall [11]. This causes challenges to detect the drivable region (due to occluded lane lines) for LC, LKA, or LDW and to detect objects (due to heavy precipitation) for adaptive cruise control [12]. The latter challenge is not as concerning due to the capabilities of radar in heavy precipitation [13,14]. However, due to the reliance on lane line visibility for LC, LDW, and LKA, these systems fail when lane lines are occluded by snow. The leading computer vision system in the automotive industry was only able to detect the left and right lane lines in 21.8% of driving done in the winter [15]. Only 0.9% of the 21.8% were high enough confidence to use for safety-critical ADAS features. Low-confident drivable region detections pose a bottleneck for advancing solutions to SAE level 3 as well as limiting the safety benefits ADAS can provide.

The lacking performance of current computer vision algorithms can be attributed to designing these algorithms for clear weather, where lane lines are completely visible. As over 70% of the U.S. becomes affected by snowfall annually, these systems must be capable of localizing even if snow is present [11]. The scope of computer vision system development must begin to include more weather-related data in order to expand the ODD. The lack of adverse weather data in developing Artificial Intelligence/Machine Learning (AI/ML) models for cameras has previously been shown to reduce the accuracy in detections when these conditions are present [16]. A previous study was done to compare the effects of snow on camera data in snow-covered road conditions, and a high correlation was demonstrated for the different levels of snow coverage [15]. This indicates an opportunity to develop different feature

extraction models depending on the level of snow coverage. To address the issue of inoperable algorithms in snow, new methodology pipelines for training more improved AI/ML models using the camera must be developed.

Our scope for this study is to introduce a new method for detecting the drivable region for automated driving localization in snow-covered roads. There have been 3 key studies found with a similar focus. Lei, Emaru, Ravankar, et. al. introduced a custom snowy weather dataset and computed drivable region semantic segmentation to determine the location of the drivable space [17]. A model trained on non-snow data showed an 80% mean Intersection of Union (mIoU) when tested on more non-snow data. However, when using the same model for testing on snow, the mIoU dropped to 19%. Their final model utilized both real-world and synthesized data, resulting in a mIoU of 83.8%. There is certainly more development necessary to achieve a robust model. Rawashdeh, Bos, and Abu-Alrub have also investigated drivable path detection in the snow [18]. A Convolutional Neural Network (CNN) deep learning model was presented that utilizes multi-modal sensor fusion. Their model uses LiDAR, camera, and radar to compute the drivable space. The results show a mIoU of 81.35% and 93.58% for the drivable space and non-drivable space, respectively. While this is a great improvement in the mIoU over the previously mentioned study, there are trade-offs with this being a more expensive solution, requiring high computational power as well as more expensive sensors. The final similar study conducted by Cai, Li, Zhou, and Mou estimates the drivable region using map-fusion images [19]. Their methodology was to compare their method of map-fusion images with non-map-fusion images for training an FCN-VGG16 CNN deep learning model. The map-fusion image model resulted in 91.7% pixel accuracy in estimating the drivable region. Further metrics such as mIoU or F1 score would be better to demonstrate the accuracy of the segmentation model; however, these metrics were not provided. None of these three studies provide precise lane information necessary for localization. Instead, they all conduct entire road detection for the drivable space. A method is needed for detecting more precise road features that can be used for localization.

To address this significant research need, we have developed and compared 24 supervised machine learning semantic segmentation models for detecting tire tracks in the snow. Inspired by the biomimicry of humans, by inferring what the human mind does during high snow coverage conditions, it was clear that there is a reliance on tire tracks for localization. Having a perception model for tire track identification would allow for confident localization during heavy snow coverage conditions. Within this paper, the key novel aspects of which we are contributing include:

- The implementation of a custom data acquisition pipeline including data collection and data labeling for tire tracks
- The explanation of the model development pipeline: feature extraction, model training, and model testing
- The analysis of model performances using the mean Intersection of Union (mIoU), FPS (frames per second), and pixel classification accuracy metrics
- A conclusive result showing the highest performing model and feature set pair for tire track identification

Methodology

The full methodology pipeline for developing & comparing models for tire track identification required seven total steps (see Figure 1). The two sub-pipelines necessary were data preparation and model development. For data preparation, the tasks were to collect, select, and label data. Next, for model development, the tasks were to extract features from the data, train the models, evaluate the models, and then use model results for analysis. The specific explanation for how these tasks were completed is demonstrated in this methodology section.

Data Preparation Pipeline

Three main steps were performed to prepare the data for the development and evaluation of the ML models. Figure 2 illustrates the three main steps of the data preparation pipeline.

Data Collection Data collection was done using the Energy Efficient and Autonomous Vehicles (EEAV) lab's research vehicle platform. This platform is built upon a 2019 Kia Niro and consists of a forward-facing RGB camera, 360° LiDAR, forward-facing radar, vehicle CAN bus interface and GNSS antennas. The forward-facing RGB camera was the only sensor used for the purposes of this study. Figure 3 shows the flow of data from the time it was collected on the vehicle to the time it was used for ML model training.

The camera provided RGB images at a 120° field of view, a resolution of 720 x 1280 pixels, and a frame rate of 30 frames per second (FPS). The recorded frames were then resampled from 30 FPS to 5 FPS to lessen the number of images. Next, a total of 1500 images were selected for use in this study based on the presence of visible tire tracks. These 1500 images were separated into 3 batches, each containing 500 images which allowed for streamlining the labeling process.

Data Labeling The program used to annotate the images was the open-sourced, web-based Computer Vision

FIGURE 1 Full methodology pipeline used for data preparation and model development, numbering the seven total steps completed, in order.

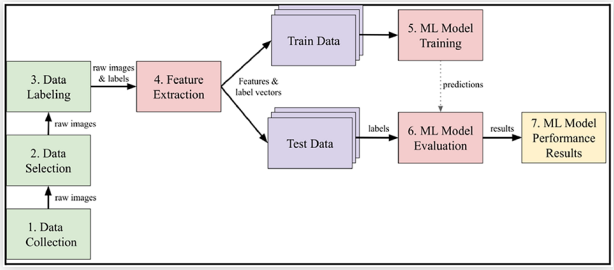


FIGURE 2 Flow diagram of the three key steps completed for data preparation.

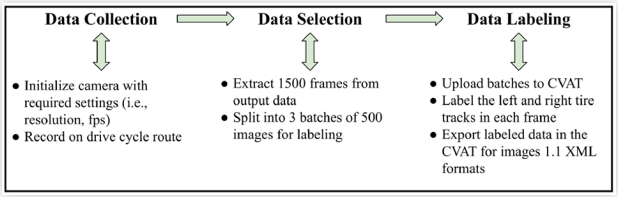
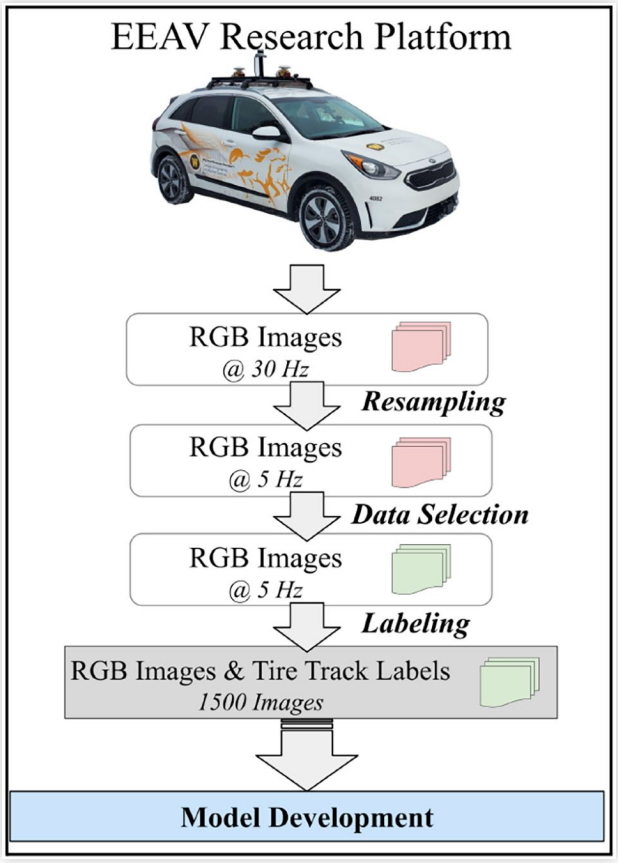


FIGURE 3 Flow diagram for data collection, resampling of the data from 30 Hz to 5 Hz, selecting 1,500 images from the dataset, and labeling of the data.



Annotation Tool (CVAT) software. CVAT allows annotations for image classification, image segmentation, and object detection. The images were uploaded to CVAT in the respective batches (as organized in the data selection process). These batches were then scanned image-by-image for annotation. The overall approach was to annotate each image for the left and right tire tracks within the lane using polygon segmentations. These annotations create a mask of the tire tracks used as the pixel-level labels of either a tire track or not a tire track, shown as white or black, respectively, in Figure 4.

CVAT uses a custom XML format for storing pixel segmentation labels called "Images for CVAT 1.1". These XML files contained the attributes such as the position of pixels and assigned tags (tire track, road, road edge boundary) for the labels used in the model development.

Model Development Pipeline

Feature Extraction Prior to feature extraction, the images were masked with a region of interest that includes the entirety of the road surface. As seen in [17, 18, 19], methods for detecting the road surface exist with high accuracy. Additional examples using more than just a camera are based on sensor fusion or LiDAR [20,21]. We represent the road surface detections seen in those examples by implementing a static Region of Interest (ROI) in which all pixels within the ROI are considered the road surface and the pixels outside the ROI are the background. This was implemented by creating an ROI mask and only using the pixels within the ROI as input to the model. The features extracted from the images were the red, green, blue, grayscale pixel values, and the pixel x, y locations, i.e. X loc and Y loc (see Figure 5 for extraction flow diagram).

These six different feature vectors were then grouped into 4 separate feature sets that represent the final input to the model. This was done to identify the combinations of the most important features that yield the highest performance results. With resizing the images from 720 x 1280 pixels to 256 x 256 pixels, these feature arrays were small enough to train the entire model without batching. The entire model was trained with the singular input array X having the shape of $X\_shape = ((m * p), n)$. Where m is the total number of images,

FIGURE 4 Example usage of CVAT for annotating the tire tracks with polygons (left) and the output mask of the labeling process (right).

FIGURE 5 The process used for feature extraction, beginning with extracting only the frames within the region of interest, then extracting the features from those pixel values.

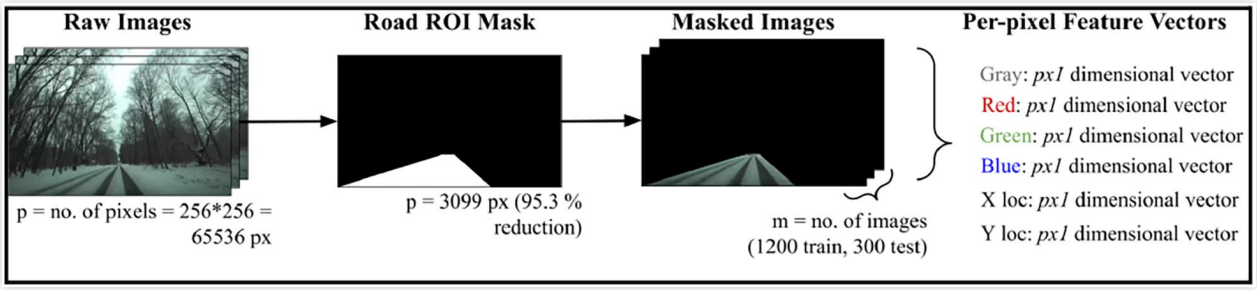


TABLE 1 Included feature sets used in model development along with their array shapes.

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, 3)	(929700, 3)
2	Red, Green, Blue	(3718800, 3)	(929700, 3)
3	Red, Green, Blue, X loc, Y loc	(3718800, 5)	(929700, 5)

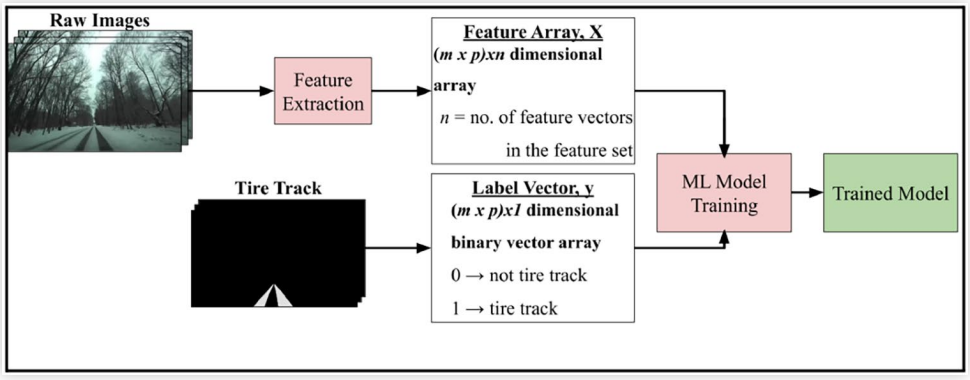
p is the total number of pixels in the ROI per image (3099 pixels for 256 x 256 dimensions), and n is the number of feature vectors in the array. These values are tabulated in Table 1 to show the total size of inputs required for both training and testing.

Model Training The input feature array X and label vector y were extracted from the images and labels coming from the feature extraction block of the pipeline. These inputs were then fed into the ML model for training (see Figure 6 for flow diagram). Six different ML algorithms were used to determine the algorithm/feature set pair with the greatest performance metrics. The ML algorithms evaluated were K-Nearest Neighbor (KNN), Naive-Bayes, Decision Trees, Random Forest, Linear Regression, Logistic Regression, and Support Vector Machines (SVM) due to the capabilities of computing binary classification [22, 23, 24, 25]. SVM ended up not being included as the training time for this algorithm was extremely high (~1 minute per image) due to the high dimensionality of the input array X. Upon eliminating SVM, the final algorithms used for training were KNN, Naive-Bayes, Decision Trees, Random Forest, Linear Regression, and Logistic Regression.

Model Evaluation The predicted outputs of the models y_{pred}were compared with the ground truth y for evaluation. The metrics used were evaluated based on the ability to draw significant conclusions on the model performance [26]. These metrics were the mIoU, pixel prediction accuracy, precision, recall, F1 score, and FPS. Equations 1-5 show how these calculations are made using the four corners of the confusion matrix: true positives, false positives, true negatives, and false negatives.

- 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

FIGURE 6 Flow diagram for ML model training. The input array X contains the features extracted from raw images and the label vector y contains the pixel status as either tire track (1) or not tire track (0).



$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. (1)}$$

$$\text{mIoU} = 1/n * \sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i + FN_i}, \text{where } n = \#\text{of classes} \quad \text{Eq. (2)}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Eq. (3)}$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{Eq. (4)}$$

$$\text{F1 Score} = 2 * \frac{precision * recall}{precision + recall} \quad \text{Eq. (5)}$$

Results

The overall results are shown in Table 2. The average pixel classification accuracy and mIoU of all models were 85.3% and 77.5%, respectively. To identify the model with the greatest mIoU, each was compared to the average mIoU of all models. The best performing model and feature set pair were the random forest model with feature set 1 as the input, containing grayscale pixel values and the pixel X, Y locations as the feature set input. All models containing pixel locations in the feature set outperform the models without pixel locations. Additionally, the results from Figure 7 indicate that grayscale pixel values produce higher mIoU than RGB values, as the top 3 performing models all use grayscale. Random forest is the best performing model for every single feature set input.

Looking at FPS performance in Figure 8, the random forest model is the slowest, other than KNN. The model that results in high mIoU, as well as great FPS performance, is decision trees with feature set 1. This model performs almost as well as random forest in mIoU, however, the frame rate is 95.94 times faster.

Conclusions

As we seek to improve autonomous vehicle functionalities beyond level 2 ADAS features, we must be prepared to include adverse weather conditions within the ODD. This study introduces a novel concept to expanding the ODD in snowy road conditions by utilizing a feature on the road surface which is commonly extracted by human drivers. We have demonstrated the methodology used for the data preparation pipeline by recording data on the EEAV research platform and labeling the data using the Computer Vision Annotation Tool (CVAT).

TABLE 2 Resulting mIoU, accuracy, FPS, precision, recall, and F1 score for all models, indicating the performance of the tire tracks identification model

Method	Feature Set	mIoU	Accuracy	FPS	Precision	Recall	F1 Score
knn	0	0.742	0.8314	0.4	0.803	0.9078	0.852
knn	1	0.832	0.9007	6.9	0.896	0.9215	0.908
knn	2	0.738	0.8341	2.9	0.828	0.8713	0.849
knn	3	0.831	0.9002	4.8	0.899	0.9164	0.908
naive	0	0.741	0.8203	7675.5	0.763	0.9624	0.851
naive	1	0.740	0.8191	819.6	0.762	0.9628	0.851
naive	2	0.732	0.8102	2099.2	0.750	0.9677	0.845
naive	3	0.733	0.8112	1813.1	0.751	0.9674	0.846
rforest	0	0.757	0.8367	30.5	0.788	0.9510	0.862
rforest	1	**0.834**	**0.9023**	11.3	0.899	0.9205	**0.910**
rforest	2	0.771	0.8578	13.4	0.848	0.8942	0.871
rforest	3	0.831	0.9003	10.5	0.899	0.9162	0.908
dtrees	0	0.757	0.8367	3899.5	0.788	0.9510	0.862
dtrees	1	0.832	0.9017	1084.1	**0.905**	0.9117	0.908
dtrees	2	0.771	0.8579	1335.3	0.849	0.8934	0.871
dtrees	3	0.801	0.8806	1186.5	0.882	0.8968	0.889
logistic	0	0.752	0.8313	32142.7	0.779	0.9551	0.858
logistic	1	0.765	0.8410	19076.3	0.785	**0.9683**	0.867
logistic	2	0.770	0.8508	36966.2	0.814	0.9342	0.870
logistic	3	0.796	0.8701	10746.7	0.831	0.9504	0.887
linear	0	0.752	0.8313	27585.0	0.779	0.9551	0.858
linear	1	0.765	0.8410	10542.4	0.785	**0.9683**	0.867
linear	2	0.770	0.8508	**37729.9**	0.814	0.9342	0.870
linear	3	0.796	0.8701	5377.6	0.831	0.9504	0.887

FIGURE 7 Difference mIoU from the mean mIoU of all trained models trained with each feature set.

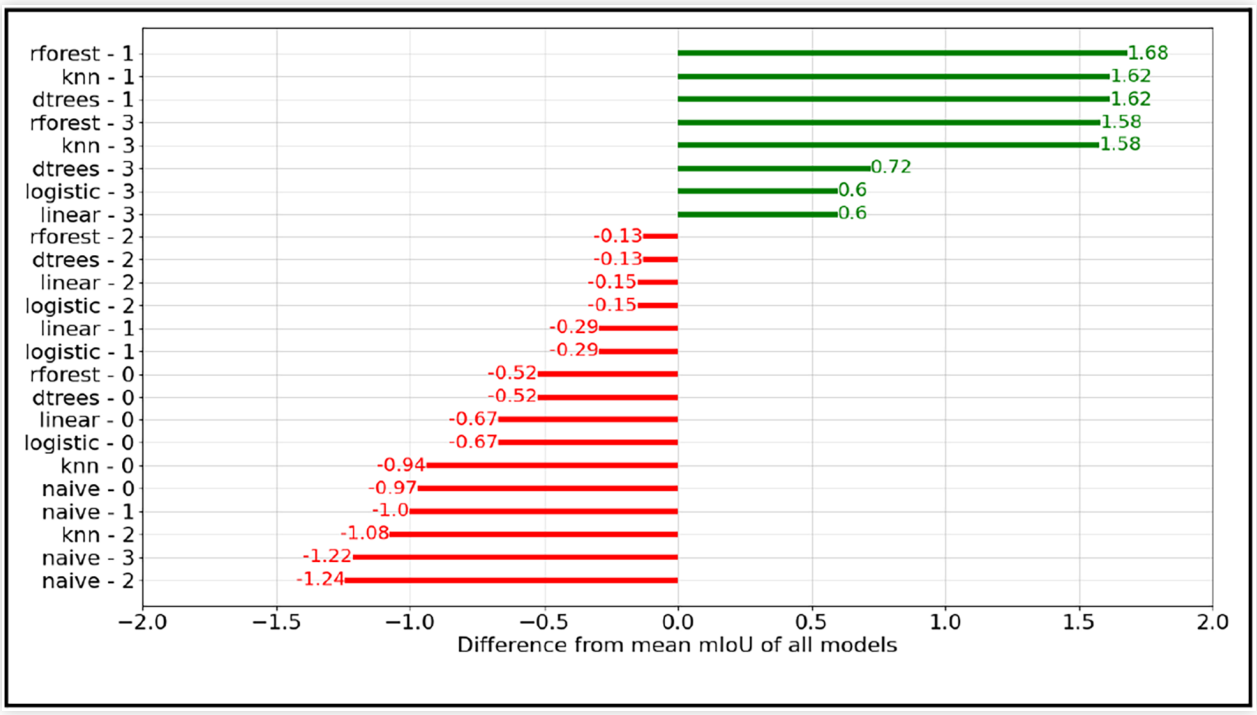
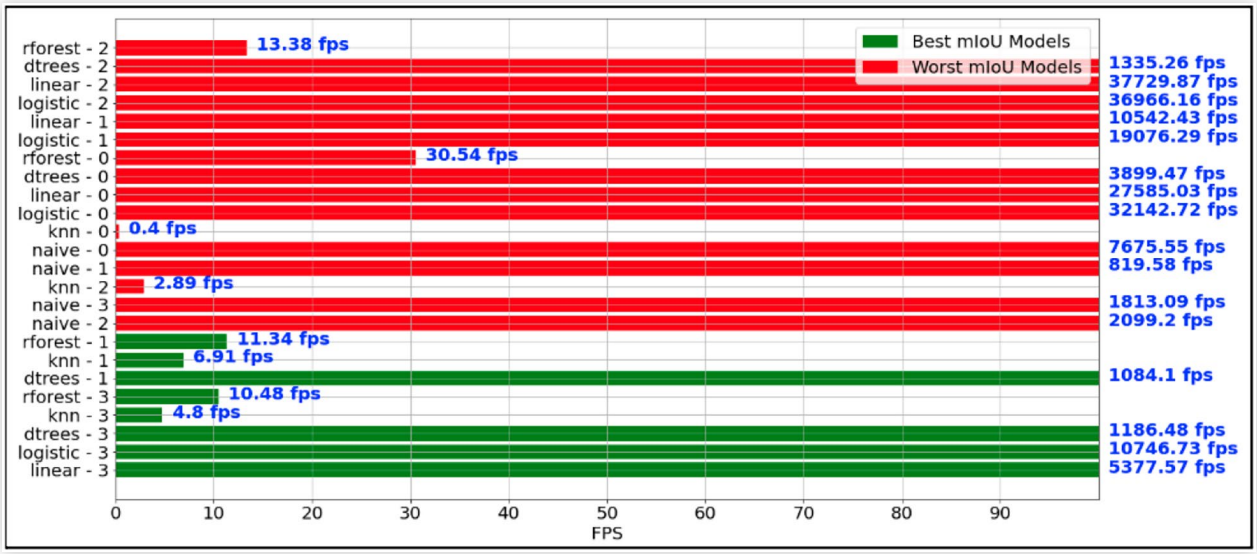


FIGURE 8 The operating frame rate of the models tested on an 8th Gen Intel i7 processor.



We also demonstrated how this data was used in the model development pipeline and showed the results comparing 6 different machine learning methods (decision trees, random forest, k-nearest neighbor, logistic regression, linear regression, and Naive-Bayes) trained with 4 different feature sets consisting of a variety of grayscale & RGB values and including/excluding pixel locations. The results from this comparison showed the random forest classifier had the highest mIoU, however, due to its low frame rate of 11.3 FPS, the decision trees classifier was identified as the model with the greatest applicability on a vehicle today. This decision trees model was trained using grayscale pixel values and the pixel X and Y locations. This model resulted in a mIoU of 83.2%, accuracy of 90.2%, a frame rate of 1084.1 FPS, a precision of 90.5%, recall of 91.2%, and an F1 score of 90.8%.

Upon expansion of this work, higher resulting metrics can be achieved by scaling data collection to include a larger dataset, implementing new feature extraction methods to

include post-processed features and/or leverage neural networks, and including more diverse data to further verify the capabilities of the model. It is targeted to include this model in a hierarchical system consisting of methods for perception in snowy weather conditions as well as clear weather conditions to achieve overall improvements in the perception system during snowy conditions.

References

1. Real-World Benefits of Crash Avoidance Technologies, Insurance Institute for Highway Safety & Highway Loss Data Institute, 2020.
2. Hearst Autos Research, "ADAS: Everything You Need to Know," Car and Driver, https://www.caranddriver.com/research/a31880412/adas/, 2020.
3. 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).
4. Society of Automotive Engineers, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," J3016_202104, 2021.
5. Sohrabi, S., Khreis, H., and Lord, D., "Impacts of Autonomous Vehicles on Public Health: A Conceptual Model and Policy Recommendations," *Sustainable Cities and Society,* 63:102457, 2020, doi:10.1016/j.scs.2020.102457.
6. Walker, C.L., Boyce, B., Albrecht, C.P., and Siems-Anderson, A., "Will Weather Dampen Self-Driving Vehicles?," *Bull. Am. Meteorol. Soc.,* 101(11):E1914-E1923, 2020, doi:10.1175/bams-d-19-0035.1.
7. Vargas, J., Alsweiss, S., Toker, O., Razdan, R. et al., "An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions," *Sensors,* 21(16), 2021, doi:10.3390/s21165397.
8. Kukkala, V.K., Tunnell, J., and Pasricha, S., "Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles," *IEEE Consumer* (2018).
9. Hnewa, M. and Radha, H., "Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-Of-The-Art and Emerging Techniques," *IEEE Signal Process. Mag.,* 38(1):53-67, 2021, doi:10.1109/MSP.2020.2984801.
10. How Do Weather Events Impact Roads?, https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm, Jul. 2021.
11. Snow & Ice - FHWA Road Weather Management, https://ops.fhwa.dot.gov/weather/weather_events/snow_ice.htm, Feb. 2021.
12. Bhadoriya, A.S., Vegamoor, V.K., and Rathinam, S., "Object Detection and Tracking for Autonomous Vehicles in Adverse Weather Conditions," SAE Technical Paper 2021-04-0079 (2021). https://doi.org/10.4271/2021-01-0079.
13. Sharma, V. and Sergeyev, S., "Range Detection Assessment of Photonic Radar under Adverse Weather Perceptions," *Opt. Commun.,* 472:125891, 2020, doi:10.1016/j.optcom.2020.125891.
14. Sharma, A., Chaudhary, S., Malhotra, J., Saadi, M. et al., "A Cost-Effective Photonic Radar under Adverse Weather Conditions for Autonomous Vehicles by Incorporating a Frequency-Modulated Direct Detection Scheme," *Frontiers in Physics,* 9:467, 2021, doi:10.3389/fphy.2021.747598.
15. Goberville, N.A., Walker, C.L., Amanda, S., and Asher, Z.D., "Snow Coverage Estimation Using Camera Data for Automated Driving Applications," *ASME Journal of Autonomous Vehicles and Systems*, 2022, doi:10.13140/RG.2.2.28134.55369/1.
16. Goberville, N., El-Yabroudi, M., Omwanas, M., Rojas, J. et al., "Analysis of LiDAR and Camera Data in Real-World Weather Conditions for Autonomous Vehicle Operations," *SAE International Journal of Advances and Current Practices in Mobility,* V129-99EJ, 2020, doi:10.4271/2020-01-0093.
17. Lei, Y., Emaru, T., Ravankar, A.A., Kobayashi, Y. et al., "Semantic Image Segmentation on Snow Driving Scenarios," *IEEE International Conference on Mechatronics and Automation (ICMA)*, 1094-1100, 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 Ground, Air, Sea and Space Vehicles and Infrastructure 2021*, unknown: 5, 2021, doi:10.1117/12.2587993.
19. Cai, Y., Li, D., Zhou, X., and Mou, X., "Robust Drivable Road Region Detection for Fixed-Route Autonomous Vehicles using Map-Fusion Images," *Sensors,* 18(12), 2018, doi:10.3390/s18124158.
20. Tao, J., "3D LiDAR based Drivable Road Region Detection for Autonomous Vehicles," diva-portal.org, 2020.
21. Zhao, G. and Yuan, J., "Curb Detection and Tracking using 3D-LIDAR Scanner," in *19th IEEE International Conference on Image Processing*, 2012, ieeexplore.ieee.org: 437-440, doi:10.1109/ICIP.2012.6466890.
22. Osisanwo, F.Y., Akinsola, J.E.T., Awodele, O., Hinmikaiye, J.O. et al., "Supervised Machine Learning Algorithms: Classification and Comparison," *International Journal of Computer Trends and Technology (IJCTT)* 48, no. 3 (2017): 128-138.
23. Singh, A., Thakur, N., and Sharma, A., "A review of supervised machine learning algorithms," in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016 , ieeexplore.ieee.org: 1310-1315.
24. Kotsiantis, S.B., Zaharakis, I., Pintelas, P., and Others, "Supervised Machine Learning: A Review of Classification Techniques," *Emerging Artificial Intelligence Applications in Computer Engineering* 160, no. 1 (2007): 3-24.
25. Sen, P.C., Hajra, M., and Ghosh, M., "Supervised Classification Algorithms in Machine Learning: A Survey and Review," in: , *Emerging Technology in Modelling and Graphics*, (Singapore: Springer, 2020), 99-111, doi:10.1007/978-981-13-7403-6_11.
26. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V. et al., A Review on Deep Learning Techniques Applied to Semantic Segmentation, *arXiv [cs.CV]*, 2017.

Contact Information

Nicholas A. Goberville
Mechanical & Aerospace Engineering Dept.
1903 W Michigan Ave.
Kalamazoo, MI 49008-5314 USA
nicholas.a.goberville@wmich.edu

Zachary D. Asher, Ph.D.
Mechanical & Aerospace Engineering Dept.
1903 W Michigan Ave.
Kalamazoo, MI 49008-5314 USA
zach.asher@wmich.edu

Definitions/Abbreviations

ADAS - Advanced Driver Assistance Systems

EEAV - Energy Efficient and Autonomous Vehicles - Research lab at Western Michigan University

LC - Lane-Centering

LDW - Lane Departure Warning

LKA - Lane-Keeping Assist

ODD - Operational Design Domain - domain an autonomous system is designed to operate within

AI/ML - Artificial Intelligence / Machine Learning

CNN - Convolutional Neural Network

FCN-VGG16 - Deep learning feature extraction model for computer vision

FPS - Frames Per Second

CAN - Controller Area Network

RGB - Red, Green, Blue

GNSS - Global Navigation Satellite System

CVAT - Computer Vision Annotation Tool

ROI - Region Of Interest

ISSN 0148-7191