**ROADSIDE VEGETATION DETECTION THROUGH DASHCAM**

# ABSTRACT

In the ever-evolving landscape of autonomous driving technology, understanding and interpreting the surrounding environment is of paramount importance. Among the myriad elements a self-driving vehicle must identify, roadside vegetation detection poses a unique challenge due to the variability in shape, size, color, and density of vegetation. This paper presents a deep learning-based approach to detect and segment roadside vegetation through dashcam images using a fine-tuned ResNet-50 model. Our method exploits the power of transfer learning, employing a pre-trained ResNet-50 model that is further fine-tuned on the MIT DriveSeg Dataset, a comprehensive dataset specifically designed for dynamic driving scene segmentation. This dataset comprises 5,000 images accompanied by masks, containing a total of 12 distinct labeled classes. The focus of our study is class 9, vegetation.

The paper describes our approach towards enhancing the model's ability to perceive and segment roadside vegetation from various other elements in the driving scene. We also evaluate our model's performance, shedding light on its effectiveness in accurately detecting vegetation, and discuss the potential impacts of this research on improving the safety and efficiency of autonomous driving systems. The findings of this study may pave the way for more sophisticated and nuanced environmental perception capabilities in self-driving vehicles, thus contributing to the broader field of autonomous driving technology.



**Figure 1. Example images from Driving Scene Vegetation detection**

# INTRODUCTION:

The proliferation of autonomous driving technology necessitates advanced models capable of identifying and interpreting diverse elements within their environment accurately. Among these elements, roadside vegetation presents a unique challenge due to its inherent variability in terms of size, shape, color, and density. This research paper introduces a deep learning-based method for the detection and segmentation of roadside vegetation using dashcam images, leveraging a fine-tuned ResNet-50 model.

The Research community has discussed the usage of semantic segmentation in various ways. For example, Taha, Abdel Aziz, Allan Hanbury et. Al (1) discussed the usage of bias metrics and reducing the overall bias for semantic segmentation in medical imaging. Monteiro et.al (2) and Wang et.al (3) provided detailed performance evaluation methods for image segmentation covering region based, boundary based and pixel-based methods to cater to the needs of the task. Joshi et.al. (4) discusses the many techniques and algorithms to optimize the performance of CNNs to model and analyze the forest features which helped us get an idea of how the neural networks can be used to detect vegetation after extracting relevant features of the forest. Ding et.al (5,6) discuss the usage of the dataset that we used which is the DriveSeg(Manual) that focuses on the manual cars and Ding et. al. (7,8) discuss about the DriveSeg(Semi-automatic) dataset. All these papers discuss the importance of classifying the regions to detect vegetation and other obstacles to generate a dynamic driving scene semantic segmentation. They also provide inputs on how the existing deep learning models and the dynamic driving scene generated can be improved for predictive analysis of car crashes. Ding et.al (9) discusses the data dealing with temporal dynamics i.e., taking both the driver and response into consideration while training. This allows the temporal-dynamic driving scene segmentation providing complex, accurate and robust results in analysis. Koonce et. al (10), discusses the detailed approach and usage of the resNet50 model which we have used to train the model upon the DriveSeg dataset in this project.

The ResNet-50 model, part of the Residual Network family, is renowned for its capacity to train deep neural networks by utilizing skip connections or shortcuts to jump over some layers. This helps combat the problem of vanishing gradients often faced when training very deep networks. Pre-trained on ImageNet, a large-scale dataset for visual object recognition, the ResNet-50 model has already learned a rich hierarchy of features, which we repurpose for the task of vegetation detection in driving scenes.

# DATA PROCESSING

The MIT DriveSeg Dataset is a rich, detailed dataset designed specifically for dynamic driving scene segmentation. It comprises 5,000 high-resolution images like the ones shown in Figures 2.1 and 2.2, each with dimensions of 1080x1920 pixels, extracted from a video stream recorded from a car dashcam. Each image in the dataset is accompanied by a labeled mask where every pixel is assigned an ID corresponding to one of the twelve classes identified in the configuration file. The granularity of this labeling system ensures a comprehensive categorization of each scene, facilitating a thorough analysis of each constituent element.



**Figure 2.1 Sample frame 1**



**Figure 2.2 Sample frame 2**

For the purposes of our study, we focused on the vegetation category, denoted by the ID 9. We preprocessed the data by converting all pixels with ID 9 to 255 (white) and all other pixels to 0 (black), thus creating binary masks where the vegetation stands out prominently against the background as shown in Figures 3 and 4. This preprocessing step enabled us to isolate the vegetation and refine our model's focus on detecting it accurately.

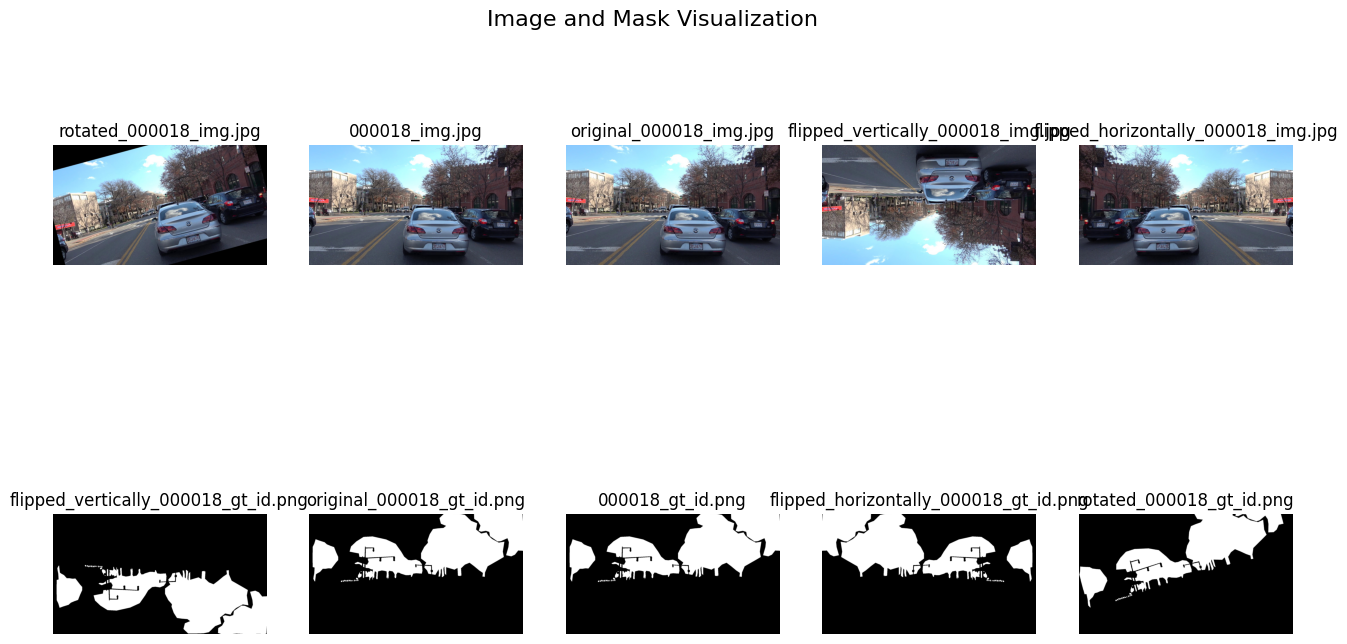


**Figure 3. Unmasked frame**

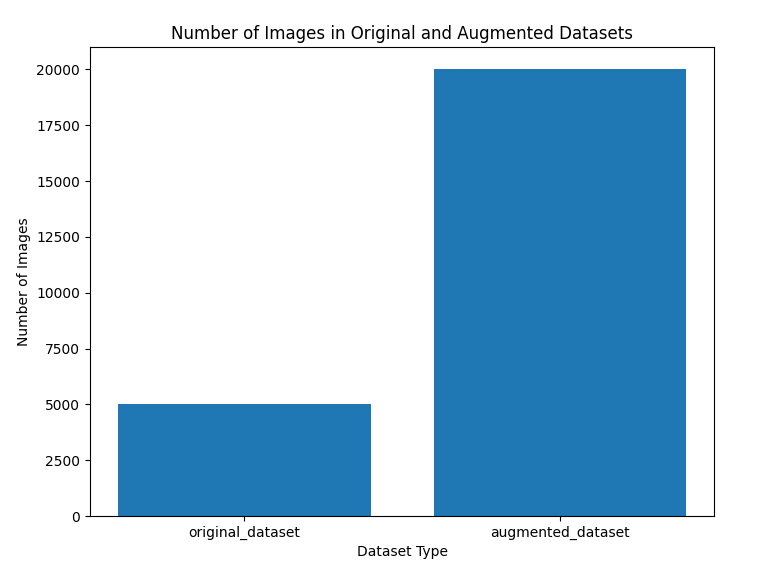


**Figure 4. Binary masked frame**

To optimize memory usage and provide a broader base for the model to learn from, we implemented a custom data augmentation function. This function processes 100 images per batch, flipping each image horizontally and vertically and rotating it by 15 degrees. These augmentations effectively quadruple the dataset size, enhancing the model's exposure to different instances of vegetation and boosting its ability to generalize.



**Figure 5. Performing Visualization operations on unmasked and masked images**



**Figure 6. Original vs Augmented dataset images**

The augmented data was then split into a training set of 12,000 images, a validation set of 3,000 images, and a test set of 5,000 images. This division ensures a robust evaluation of the model's performance, allowing us to fine-tune it based on the validation feedback and finally test it on unseen data.

# EXPERIMENT/ OUR APPROACH

We utilized torchvision's transformation functionality to pre-process the data before feeding it to the ResNet-50 model. The ResNet-50 model is a deep residual network, renowned for its ability to learn complex patterns without succumbing to the vanishing gradient problem. It is a pre-trained model with 50 layers, including residual blocks that allow the direct transmission of input features to deeper layers. By fine-tuning this model on our specific task of vegetation detection, we aim to leverage its deep learning capabilities for our specific task, building on its already solid foundation.

**A diagram of a network

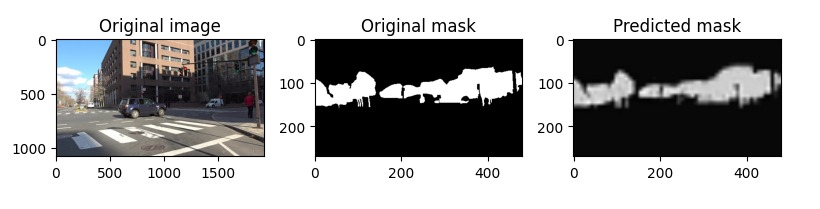
Description automatically generated with medium confidence**

**Figure 7. ResNet50 Architecture**

Our research employed the fine-tuning approach on a pre-trained ResNet-50 model using the augmented data prepared from the MIT DriveSeg Dataset. The choice of fine-tuning as opposed to training a model from scratch was driven by the fact that pre-trained models like ResNet-50 have already learned a rich set of features from a large dataset (such as ImageNet).

Our training was implemented over 10 epochs, an epoch being a complete pass through the entire training dataset. At each epoch, we used the validation set to evaluate the model's performance by calculating the loss. This validation loss gives a measure of how well the model is performing on data it has not been trained on, thus providing an indicator of the model's generalizability.

Moreover, to track the model's learning progression, we plotted the output at every epoch. This visualization enabled us to monitor the model's performance in real-time, offering insights into its learning process and aiding in the early detection of any potential issues such as overfitting or underfitting.



**Figure 8. Training progression of the model**

Interestingly, we observed a notable improvement in the model's performance after just three epochs. This rapid enhancement highlights the efficacy of transfer learning in our context and the utility of the ResNet-50 model's pre-trained weights for our task.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Epoch | Training Loss | Validation Loss |
| ResNet50 | 1 | 7904.90 | 7436.95 |

**Table 1. Model Training and validation loss metrics Epoch=1**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Epoch | Training Loss | Validation Loss |
| ResNet50 | 10 | 1297.61 | 1588.15 |

**Table 2. Model Training and validation loss metrics Epoch=10**

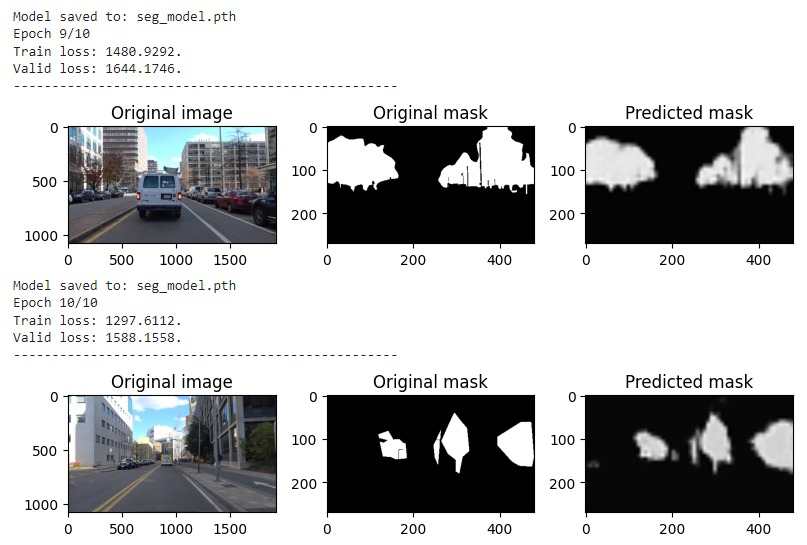
We observed the Training and validation loss decrease drastically after 10 epochs by more than 75% with reference to the tables 1 and 2 as shown above.

A screenshot of a computer

Description automatically generated with low confidence

**Figure 9. Training Progression of the model with evaluation at first two epochs**

Each epoch took approximately 10 minutes on an A100 GPU, a high-performance graphics processing unit known for its efficient handling of machine learning workloads. We carried out our computations on Google Colab, which provided the requisite high RAM for our intensive deep learning computations. The training progression of the first and last two epochs are shown in figures 9,10.



**Figure 10. Training progression of the model at the last two epochs**

# EVALUATION:

Given the nature of our task, traditional performance metrics such as accuracy, precision, and recall are not readily applicable in their standard form. Our model's output is an image, specifically a mask that segments vegetation from the rest of the scene. As such, we need a way to compare this output image with the ground truth image on a pixel-by-pixel basis.

To accommodate this requirement, we devised custom functions to calculate pixel-wise accuracy, precision, and recall. Each pixel in the image is treated as a separate entity, and the predicted mask is compared with the original mask pixel by pixel. We consider a prediction to be accurate if the predicted value for a pixel matches the actual value in the original mask.

To calculate the precision, recall and accuracy of the predicted mask, the following functions were used to evaluate the performance of the segmentation model on the dataset:

First, the function percentage\_precision(mask, predicted\_mask) was used to calculate the percentage of pixels that were correctly classified as positive (true positive) or negative (true negative), as well as the percentage of pixels that were incorrectly classified as positive (false positive) or negative (false negative). This function takes as input the ground truth segmentation mask and the predicted segmentation mask, both of which are matrices of size 256x256 containing binary values indicating whether each pixel belongs to the foreground (1) or the background (0).

The function creates a matrix of size (256, 256) with all elements as 255, which is used to calculate the total number of pixels in the image. It then calculates the false positive and false negative percentages by subtracting the ground truth mask from the predicted mask and vice versa, and setting all negative or positive values to 0, respectively. The true positive percentage is calculated by adding the predicted mask and the ground truth mask, setting all values less than 255 to 0, and all values greater than 255 to 255. Finally, the precision is calculated as the true positive divided by the sum of true positive and false positive, and the function returns the precision value.

Second, the function percentage\_recall(mask, predicted\_mask) was used to calculate the percentage of pixels that were correctly classified as positive (true positive) or negative (true negative), as well as the percentage of pixels that were incorrectly classified as positive (false positive) or negative (false negative). This function also takes as input the ground truth segmentation mask and the predicted segmentation mask. The false positive and false negative percentages are calculated in the same way as in the percentage\_precision function, and the true positive percentage is calculated by adding the predicted mask and the ground truth mask, setting all values less than 255 to 0, and all values greater than 255 to 255. The recall is then calculated as the true positive divided by the sum of true positive and false negative, and the function returns the recall value.

These functions were used to evaluate the precision and recall of the segmentation model, which are important metrics for measuring the accuracy of a segmentation model. Precision measures the proportion of positive predictions that are true positives, while the recall measures the proportion of true positives that are correctly identified by the model. We also use an accuracy function to calculate the accuracy. The percentage\_accuracy function calculates the accuracy between a ground truth mask and a predicted mask. It first creates a matrix to determine the total number of pixels in the image. Then it calculates the absolute difference between the two masks and the percentage of pixels that are different. Finally, it subtracts the percentage difference from 100 to obtain the accuracy and returns the value.

# RESULTS

The effectiveness of our approach in roadside vegetation detection through dashcam images was evident in the results of our experiment. We observed promising performance of the fine-tuned ResNet-50 model on the test set, demonstrating the model's robustness and accuracy in segmenting vegetation from dynamic driving scenes.

The performance metrics for the test set are presented in the following Table 3:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall |
| ResNet50 | 83.76% | 84.87% | 91.54% |

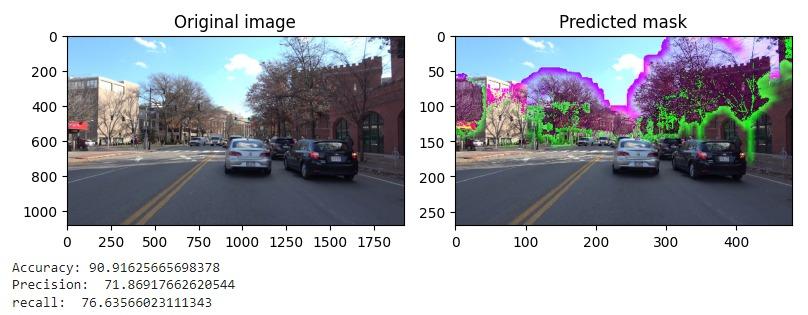
**Table 3. Model Performance metrics**

Our test results in reference to Table 3, demonstrated a robust performance with a pixel-wise accuracy of approximately 83.76%. This indicates that our model was able to accurately predict the class of each pixel (vegetation or non-vegetation) for more than 83% of the total pixels across all test images. This high degree of accuracy underscores the model's ability to effectively discern and categorize vegetation from the other elements in the images.

Moreover, the model exhibited a commendable precision of approximately 84.88%. Precision is a measure of the model's ability to correctly identify positive instances, i.e., correctly identifying vegetation pixels as vegetation. A high precision suggests that the model has a low false-positive rate, thereby minimizing the risk of misclassifying non-vegetation pixels as vegetation.

Furthermore, the model achieved a recall of approximately 91.54%, which is a measure of the model's ability to identify all relevant instances, in this case, all vegetation pixels. A high recall rate implies that the model was successful in detecting most of the vegetation pixels present in the images, thus reducing the likelihood of false negatives.

The following visuals Image (11, 12, 13) provide a glimpse into the results yielded by our model. The image on the left represents the original, while the one on the right displays a resized counterpart with an overlaid mask. The accuracy, precision, and recall metrics are computed based on the pixel-wise discrepancy between the original and the predicted mask, specifically for this sample.



**Figure 11. Result example frame 1 with Predicted mask and performance metrics**

A picture containing text, screenshot, graphics software, multimedia software

Description automatically generated

**Figure 12. Result example frame 2 with Predicted mask and performance metrics**



**Figure 13. Result example frame 3 with Predicted mask and performance metrics**

These results shown in Figures 11,12 and 13 collectively suggest that the fine-tuned ResNet-50 model demonstrates a strong performance in roadside vegetation detection. The relatively high accuracy, precision, and recall rates indicate a balanced performance, with the model effectively minimizing both false positives and false negatives.

# REFERENCES

1. *Taha, Abdel Aziz, Allan Hanbury, and Oscar A. Jimenez del Toro. "A formal method for selecting evaluation metrics for image segmentation." 2014 IEEE international conference on image processing (ICIP). IEEE, 2014.*
2. *Monteiro, Fernando C., and Aurélio C. Campilho. "Performance evaluation of image segmentation." Image Analysis and Recognition: Third International Conference, ICIAR 2006, Póvoa de Varzim, Portugal, September 18-20, 2006, Proceedings, Part I 3. Springer Berlin Heidelberg, 2006.*
3. *Wang, Zhaobin, E. Wang, and Ying Zhu. "Image segmentation evaluation: a survey of methods." Artificial Intelligence Review 53 (2020): 5637-5674.*
4. *Joshi, Durga, and Chandi Witharana. "Roadside Forest Modeling Using Dashcam Videos and Convolutional Neural Nets." The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 46 (2022): 135-140.*
5. *Ding, L., Terwilliger, J., Sherony, R., Reimer, B. & Fridman, L. (2020). MIT DriveSeg (Manual) Dataset for Dynamic Driving Scene Segmentation. Massachusetts Institute of Technology AgeLab Technical Report 2020-1, Cambridge, MA. (*[*pdf*](https://ieee-dataport.s3.amazonaws.com/docs/25625/MIT_DriveSeg_Manual.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJOHYI4KJCE6Q7MIQ%2F20220118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220118T160943Z&X-Amz-SignedHeaders=Host&X-Amz-Expires=86400&X-Amz-Signature=df36110567849dbee0c52a4c6d064b8364452ca28c26576a94a0bd6f9b23bd0d)*)*
6. *Ding, L., Terwilliger, J., Sherony, R., Reimer, B. & Fridman, L. (2020). MIT DriveSeg (Manual) Dataset. IEEE Dataport. DOI:* [*10.21227/mmke-dv03*](http://dx.doi.org/10.21227/mmke-dv03)*.*
7. *Ding, L., Glazer, M., Terwilliger, J., Reimer, B. & Fridman, L. (2020). MIT DriveSeg (Semi-auto) Dataset: Large-scale Semi-automated Annotation of Semantic Driving Scenes. Massachusetts Institute of Technology AgeLab Technical Report 2020-2, Cambridge, MA. (*[*pdf*](https://ieee-dataport.s3.amazonaws.com/docs/25911/MIT_DriveSeg_Semiauto.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJOHYI4KJCE6Q7MIQ%2F20220118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220118T160915Z&X-Amz-SignedHeaders=Host&X-Amz-Expires=86400&X-Amz-Signature=15a2566fca6beb2660bbafc839a2b90c70e013f0b082bc85e6ac1c805ac58837)*)*
8. *Ding, L., Glazer, M., Terwilliger, J., Reimer, B. & Fridman, L. (2020). MIT DriveSeg (Semi-auto) Dataset. IEEE Dataport. DOI:* [*10.21227/nb3n-kk46*](http://dx.doi.org/10.21227/nb3n-kk46)*.*
9. *Ding, L., Terwilliger, J., Sherony, R., Reimer. B. & Fridman, L. (2019). Value of Temporal Dynamics Information in Driving Scene Segmentation. arXiv preprint arXiv:1904.00758. (*[*link*](https://arxiv.org/abs/1904.00758)*)*
10. *Koonce, Brett, and Brett Koonce. "ResNet 50." Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization (2021): 63-72.*