TERRAIN TRAVERSABILITY ESTIMATION USING FULLY CONVOLUTIONAL NETWORKS

DEEKSHA SETHI (DEESETHI@SEAS.UPENN.EDU), KEYUSH SHAH (KEYUSH06@SEAS.UPENN.EDU), TANAYA GUPTE (TANAYAG@SEAS.UPENN.EDU),

ABSTRACT. In this project, we propose implementing a Supervised Fully Convolutional Network with Attention to image segmentation and traversability estimation. The model aims to proficiently categorize terrains into six driveable and non-driveable classes. Our custom Attention-based FCN demonstrates improved prediction of traversable regions, achieving a 2% increase in mean IoU for the training data and a 10% improvement in the test data specifically for 'smooth trail'. We have also introduced innovative metrics for assessing the traversability of a region. Future endeavors will focus on deploying the trained model for efficient path planning, thereby enhancing the overall navigation capabilities of autonomous systems in off-road scenarios.

1. INTRODUCTION

1.1. **Overview.** In the current dynamic landscape of robotics research, the crucial factors of environmental adaptability and awareness have gained prominence. For robots to navigate effectively, it is essential to possess a precise comprehension of the traversability of the terrain [1]. The safe navigation of robotics and autonomous vehicles on off-road terrains poses a significant challenge, necessitating effective path-planning strategies. To address this challenge, a critical step involves the classification of terrain into drivable and non-drivable regions. In addition to identifying traversable terrain, mitigating the occurrence of "phantom braking" is imperative, as it can lead to unnecessary braking in autonomous vehicles even in the absence of obstacles. Traversability estimation stands as a foundational principle in off-road navigation, particularly in the context of autonomous vehicles and advanced driver-assistance systems (ADAS). This estimation entails evaluating the feasibility of navigating through specific terrains or paths, aiming to ascertain whether a vehicle can traverse an area successfully without encountering obstacles or becoming immobilized.

1.2. **Contributions.** In this study, we implement a Baseline FCN model, drawing inspiration from [2]. We suggest enhancing the baseline model by incorporating attention gates, demonstrating improved performance, particularly for the smooth trail and sky classes. Contextualization plays an important role in image segmentation and we build upon that idea by introducing attention gates to the baseline model The Attention UNet, a well-known model in image segmentation, serves as a reference for our exploration. We introduce the concept of traversability estimation into asymmetric FCNs with attention, supported by experimental results presented in this work.

1.3. **Related Work and Background.** In the work presented in [3], a custom CNN, based on UNet, is employed for semantic segmentation using a novel off-road imagery dataset. Another approach, discussed in [4], involves generating labels from past trajectories, considering regions traversed by the vehicle as traversable. A neural network, employing PSPNet with ResNet50, is trained using these self-supervised labels to classify terrains based on traversability. Similarly, in the work in [5], a self-supervised Visual Transformer model is utilized to assess traversability in challenging environments like forests and grasslands. The study in [6] introduces a complete navigation algorithm leveraging self-supervised data collected from the real world. The approach involves a network with convolutions, fully-connected layers, and a recurrent LSTM unit. Contrary to past attempts relying solely on geometric methods, which fall short in distinguishing the quality of terrain and identifying traversable regions accurately, our approach, inspired in [2], employs custom convolutional neural networks to extract traversable regions from images. This includes an Attention with FCN. Subsequently, we segment the traversable part from the entire region, indicating the direction in which the robot should move and the potential path it could take.

2. APPROACH

Our task is to estimate terrain traversability from a camera-captured image. Initially, we identify traversable and non-traversable areas through semantic segmentation (see section 2.1). We then analyze traversability and suggest an optimal path, detailed in section 2.2. The integrated system is depicted in Figure 1.

2.1. **Semantic Segmentation.** Our model segments distinct regions within the field of view (FOV) into appropriate classes using an input image (RGB or BGR) captured from a mobile robot/vehicle camera. We achieve this semantic segmentation using Fully Convolutional Networks (FCNs). An FCN is built using convolution, pooling, and transposed convolutions. These networks follow a *downsampler-upsampler* layout. The downsampler consists of consecutive convolutions used to extract the contextualized features in the image. The upsampler contains consecutive transposed convolutions, which construct the

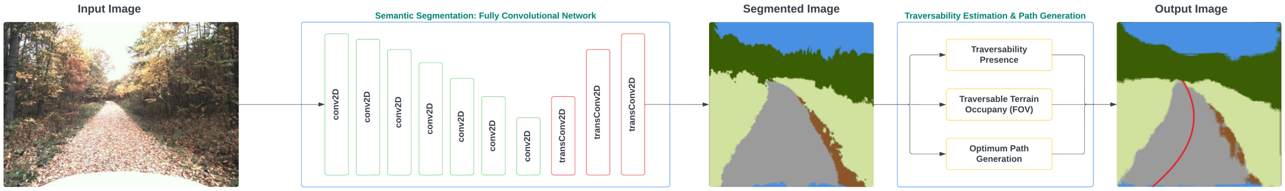


FIGURE 1. Flowchart demonstrating our approach to estimating traversability from a provided input image.

segmented image by localizing the context obtained from the downsampler. FCN also takes advantage of skip connections between the downsampling and upsampling convolutions to account for the loss of information while downsampling.

In our implementation, we experiment with three variants of FCN. All the models have been written from scratch using Python3 and PyTorch.

2.1.1. *Baseline FCN.* Our baseline FCN is inspired by the work in [2]. The architectural details are provided in table 1.

- *Downsampler*: 2D convolutions are performed to extract contextual information from the input images and make the network translation equivariant. Pooling layers are employed to induce translation invariance in the network. Network-In-Network (`nin`) layers are 1 x 1 convolutions. These layers are designed to ensure that all inputs in the skip connection have the same dimensionality. Layers `conv fc6` and `conv fc7` are convolutional layers followed by a dropout layer.
- *Upsampler*: 2D transpose convolutions are performed on the downsampled images to construct the segmented image. The upsampling of the image uses skip connections. We perform an element-wise summation between the outputs of the downsampler and the upsampler to form fusion layers. Three fusion layers in our architecture are: (`nin crop conv5` $\oplus$ `up1`), (`crop nin pool1` $\oplus$ `up2`) and (`crop nin conv1` $\oplus$ `up3`). Fusion layers are used to account for the lost spatial information while downsampling.

Besides, we use fusion layers as our skip connections to make up for the spatial information lost. This helps in preserving fine-grained details that might be lost during downsampling operations like pooling. We intend to understand both the local details as well as the global context and these layers bridge this semantic gap by concatenating features at different resolutions.

Layer	˜Kernel Size	**Input Shape**	**Output Shape**
conv1	7	3 x 300 x 300	96 x 227 x 227
pool1	3	96 x 227 x 227	96 x 75 x 75
conv2	2	96 x 75 x 75	256 x 71 x 71
pool2	2	256 x 71 x 71	256 x 35 x 35
conv3	3	256 x 35 x 35	512 x 35 x 35
conv4	3	512 x 35 x 35	512 x 35 x 35
conv5	3	512 x 35 x 35	512 x 35 x 35
pool5	3	512 x 35 x 35	512 x 11 x 11
conv fc6	6	512 x 11 x 11	4096 x 10 x 10
conv fc7	1	4096 x 10 x 10	4096 x 10 x 10
up1	4	8 x 10 x 10	8 x 20 x 20
crop conv5	-	512 x 35 x 35	512 x 20 x 20
nin crop conv5	1	512 x 20 x 20	8 x 20 x 20
up2	4	8 x 20 x 20	8 x 40 x 40
nin pool1	8	96 x 75 x 75	8 x 75 x 75
crop nin pool1	8	8 x 75 x 75	8 x 40 x 40
up3	5	8 x 40 x 40	8 x 120 x 120
nin conv1	1	96 x 227 x 227	8 x 227 x 227
crop nin conv1	-	8 x 227 x 227	8 x 120 x 120
nin6	1	8 x 120 x 120	6 x 120 x 120
conv6	2	6 x 120 x 120	6 x 240 x 240
conv7	16	6 x 240 x 240	6 x 300 x 300

TABLE 1. Architecture details for Baseline FCN. The shapes for the input and output for each layer are in the format: (Number of Channels, Height, Width)

2.1.2. *Attention with FCN.* In this modified version of FCN, we have incorporated three attention gates into the upsampler component of the baseline FCN [9]. The rationale behind introducing these attention gates is to generate attention coefficients, which serve to amplify regions with more contextual information and diminish the influence of regions with less context.

To illustrate the functionality of these attention gates, let's consider an example where the attention gate is applied before the fusion layer (`nin crop conv5` $\oplus$ `up1`). In this scenario, `up1` acts as the gating signal, and `nin crop conv5` serves as the input feature map. The gating signal is added to the input feature map and passed through a rectified linear unit (ReLU) activation function, constituting the intermediate stage of our attention block. The resultant intermediate output is then passed through a sigmoid activation function to obtain a normalized range of attention coefficients ranging from 0 to 1.

These attention coefficients, having dimensions (H x W) corresponding to the input feature map, undergo a Hadamard product operation with each layer of the input feature map. Notably, the input feature map in the attention gate corresponds to the output of the respective layer in the downsampler. As a result, these attention coefficients essentially elevate the significance of the most prominent features in the convolved images and scale their magnitudes, thereby enhancing contextualization in the fusion layer.

It is crucial to note that the gating signal consistently originates from the output of the corresponding layer in the upsampler. This ensures that only those features deemed significant during upsampling in the respective transpose convolution layer are validated and amplified in the input feature map.

2.2. **Traversability Estimation.** Our FCN models give us segmented images that show the traversable regions of the area covered by the image. Nevertheless, we have not yet quantified traversability, meaning we lack information on which areas or paths are more traversable than others. To address this, we formally establish traversability based on three specific criteria.

(1) Traversability Presence: If the segmented image includes either of the two classes, namely smooth trail or rough trail, we affirm the presence and possibility of traversability within the given field of view or image.
(2) Traversable Terrain Occupancy: In this context, we measure the extent of the field of view or image that is encompassed by a traversable terrain determined by the first criterion.
(3) Optimum Path Generation: Upon confirming the presence of traversable areas in the image, we suggest an optimal path for the subject to navigate. This path is determined by fitting a polynomial function to the contour values of the identified traversable regions in the segmented image.

3. Experimental Results

3.1. **Dataset.** We used two datasets for training and testing all our models- the Yamaha-CMU Off-Road Dataset and the Off-road Autonomous Driving Segmentation Dataset. The Yamaha CMU Dataset is a labeled dataset containing 1076 images. These have seven classes- sky, rough trail, smooth trail, traversable grass, high vegetation, non-traversable low vegetation, and obstacle. The Off-road Autonomous Driving Segmentation Dataset is another labeled dataset combined by researchers at CAVS (Center for Advanced Vehicular Systems). It contains 1700 images. The images here are labeled using six classes- smooth trail, rough trail, small vegetation, forest, sky, and obstacles.

We decided to use these datasets because they were made specifically for Off-road Autonomous Driving. They contained a diverse set of images with varying lighting conditions. Our final training and testing for all models was done on the Yamaha CMU Dataset to maintain uniformity.

3.2. **Pre-Processing.** We use Cross Entropy Loss as our criterion for computing the loss between our predictions and labels. Cross entropy requires our targets to be one-hot encoded. The labels we had in our dataset were images of shapes 300x300x3. We integer encoded these to 300x300 with each element in the array representing the class for that particular pixel coordinate. to accomplish this we first converted our labeled images to HSV format from BGR format. We do this because HSV colorspace is more intuitive than BGR. We had to differentiate between colors that would look quite similar to the human eye, e.g. light green corresponding to small vegetation and dark green corresponding to large vegetation. A single shade of one color corresponds to a larger range in BGR than it does in HSV. Thus HSV ensures the accuracy of the encoding. We created individual masks for all classes corresponding to a particular range of HSV values. We applied these masks on all images and checked which mask generates the highest value for every pixel (suggesting the pixel belongs to that class). The integer corresponding to that mask is thus assigned to the pixel.

3.3. **Loss.** As detailed in Section 5.2, our chosen criterion is the Cross-Entropy Loss. This particular loss function is widely employed in segmentation tasks, given that segmentation essentially involves solving a multi-class classification problem. The softmax function, commonly used in conjunction with Cross Entropy Loss, assigns probabilities to each pixel, indicating its likelihood of belonging to a specific class. Moreover, it effectively handles class imbalance, where certain classes may have more pixels in an image than others, by penalizing misclassifications in sparser classes.

Given that our segmentation task is focused on extracting the traversable parts of the terrain, our primary concern lies in accurately segmenting the classes “smooth terrain” and “rough terrain.” To address this, we apply weights to the Cross-Entropy Loss, assigning a higher weight to these two classes. Specifically, we assigned weights as follows: *Sky: 1.0, Rough Trail: 1.5, Smooth Trail: 1.5, High Vegetation: 1.0, Low Vegetation: 1.0, Obstacle: 1.0.* Notably, this adjustment led to a significant improvement in the models’ learning process. Initially, the Pyramid Pooling with FCN was predominantly learning a single class that occurred most frequently in the images (“sky” in most cases). However, upon introducing the weights, the model started learning other classes as well.

3.4. **Performance Metrics.**

3.4.1. *IoU (Intersection over Union).* We use IoU as a metric to evaluate the performance of our models. This is a very well-known evaluation metric for tasks such as segmentation, object detection, and tracking. It is used for evaluation in object detection and segmentation challenges such as the popular PASCAL VOC challenge. This metric calculates the overlap of the predicted and target images for all classes. The amount of overlap gives intuition on how accurate the segmentation is. The formula for IoU is given below-

$$\frac{Predicted \cap Target}{Predicted \cup Target}$$

3.4.2. *Heading Error.* We have evaluated the quality of the segmented image in giving us traversable and non-traversable regions using IoU. However, we have not yet quantified how accurate the path, or specifically the heading, that the segmented image suggests is. To do this, we first find the vector that represents the heading angle of the vehicle. We obtain a mask to segment out the smooth terrains in the region (since this preferred region for traversability). By using the contours of this segmented region, we find the ”moment” of the region. This will give us a point that we can call the ”centroid” which loosely corresponds to the ”center of mass” of the image. Since a path is generally trapezoidal shaped, joining the centroid to the center of origin of the path gives us the vehicle’s orientation within the traversable terrain. Figure 2 shows the vectors for two different paths.

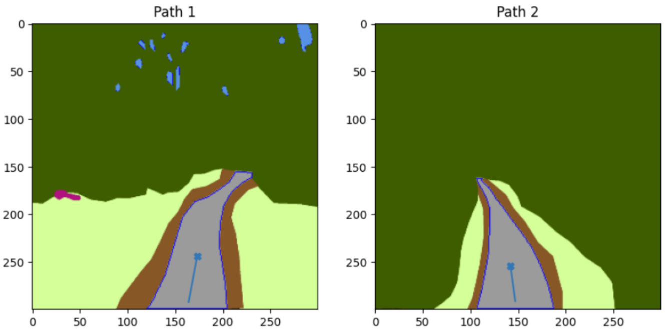


FIGURE 2. The error between the heading for these two paths is 18.315 degrees.The purple line shows the contours.

3.5. **Results.** Our experiments were conducted over 200 epochs using the Yamaha-CMU Off-Road training data, specifically limited to 931 images. This section exclusively presents the outcomes of the Baseline FCN and the Attention with FCN (Attention FCN).

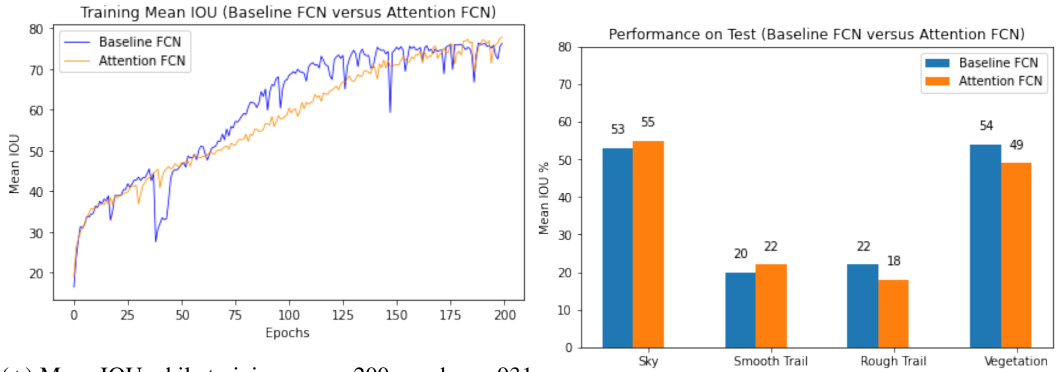
The training dataset lacked a sufficient number of images featuring obstacles. Consequently, for the sake of simplicity, the classes “obstacle,” “low vegetation,” and “high vegetation” have been consolidated into a single category labeled “Vegetation.”

Figure 3 presents a comparison between the attention FCN model and the baseline FCN. As evident from the plots, the training progression of the attention FCN exhibits relatively greater consistency than that of the baseline FCN. Additionally, a slight increment in the mean Intersection over Union (mIOU) for the attention FCN is noticeable after 200 epochs.

The marginal improvement in mIOU with the Attention FCN can be attributed to the heightened contextualization in the fusion system compared to the baseline FCN. Given the critical role of context in image segmentation, attention proves effective in addressing contextual deficiencies, resulting in a 2% improvement in the training mIOU by the 200th epoch after introducing attention. However, owing to the limited training data, this improvement remains marginal. With a more extensive dataset, the performance could potentially increase more consistently. Furthermore, an extended training duration would likely lead to a more substantial performance gain for the Attention with FCN model over the baseline FCN.

Turning to the testing data results, an elevated mIOU is observed for the classes ”Sky” and ”Smooth Trail.” While the overall mIOU for the dataset is lower compared to the original paper [2] that inspired this work, it is important to note that our network was trained on approximately one-fourth of the dataset used in the original paper. Despite the limited training data, the model demonstrated improved performance for two classes, including the highly relevant ”smooth trail” class. Had the Attention FCN been trained on a larger dataset, as in the original paper, the results might have been more favorable. Notably, the classes ”Vegetation” and ”Rough Trail” exhibited suboptimal performance after the introduction of attention. The observed reduction in performance can be attributed to the limited hyperparameter tuning conducted on the model, yet it has the potential for improvement with more effective tuning.

While there is room for fine-tuning the parameters of the Attention FCN model to enhance its performance, the introduction of attention to an FCN holds the potential to improve the overall network performance. Our study has demonstrated this improvement on a small dataset, and it is anticipated that the performance gains would scale proportionally with a larger dataset. Attention gates predominantly excel in capturing objects characterized by substantial shape variability. In our specific scenario, trails exhibit considerable diversity in shape. Notably, the training times for both the Baseline FCN and Attention FCN were comparable. Hence, employing attention is expected to yield better results within the same training duration.



(A) Mean IOU while training across 200 epochs on 931 images.

(B) Mean IOU on the test set of 145 images.

FIGURE 3. Mean IOU of the Baseline FCN and Attention with FCN while training and testing on the Yamaha-CMU dataset.

”Phantom Braking” refers to the unnecessary braking/stopping of an autonomous vehicle when it predicts an obstacle although it does not exist. A reason for this can be shadows. An autonomous system can mistake shadows for obstacles. Our model can distinguish between shadows and obstacles well. This can be seen in the figure 4

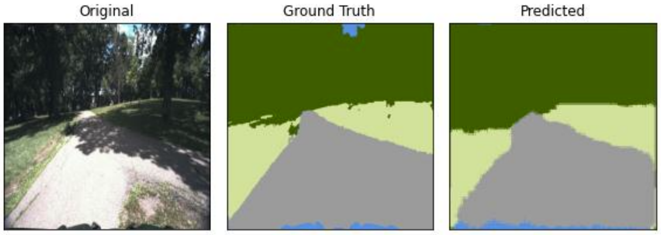


FIGURE 4. The traversable region is correctly segmented without mistaking the shadow as an obstacle or any other class.

4. DISCUSSION

We extended the baseline Fully Convolutional Network (FCN) by integrating Pyramid Pooling. This approach is thoroughly explained in the Appendix. Though we were unable to obtain noteworthy results within the allocated time frame.

Contemporary experiments in semantic segmentation demand extensive training data for optimal performance. Despite the scarcity of quality datasets and the need for highly annotated data, there is a growing interest in constructing models from limited examples. Inexpensive learning, while enticing, comes with the trade-off that deep models may not always generalize well with a small number of samples. Few Shot Learning (FSL) is a well-discussed area in AI, involving training with a limited number (k) of labeled examples. The utilization of pre-trained models is crucial in FSL.

Stochastic Resonance (SR) is a technique involving the addition of white or Gaussian noise to enhance a signal. We plan to employ SR in conjunction with attention to enhance the performance of our models, inspired by our reading of this technique in several papers.

Up-weighing relevant classes proved beneficial in improving our models’ performance, as demonstrated through both rough and smooth trials. This approach is particularly useful for addressing imbalanced datasets, as observed in our case.

The Generalized Intersection over Union (GIoU) metric, an extension of IoU, addresses the limitation of IoU by considering predictions close to the true value that do not intersect. While IoU gives a score of 0 for a region with no overlap, even if it is close to the true region, GIoU incorporates a ”closeness” parameter, providing a more nuanced performance metric.

The quality of a traversable region is determined by the ”length” and ”curvature” of the optimal path. Paths with multiple curves or bends, indicative of avoiding obstacles, may not be desirable due to the increased effort and time required for traversal.

REFERENCES

[1] Sevastopoulos, C., & Konstantopoulos, S. (2022). A survey of traversability estimation for mobile robots. IEEE Access, 10, 96331-96347.

[2] Maturana, D., Chou, P. W., Uenoyama, M., & Scherer, S. (2018). Real-time semantic mapping for autonomous off-road navigation. In Field and Service Robotics: Results of the 11th International Conference (pp. 335-350). Springer International Publishing.

[3] Field and Service Robotics, 2018, Volume 5, ISBN : 978-3-319-67360-8, Daniel Maturana, Po-Wei Chou, Masashi Uenoyama.

[4] Seo, Junwon & Sim, Sungdae & Shim, Inwook. (2023). Learning Off-Road Terrain Traversability with Self-Supervisions Only.

[5] Jonas Frey, & Matias Mattamala, & Nived Chebrolu, & Cesar Cadena, & Maurice Fallon, & Marco Hutter. (2023). Fast Traversability Estimation for Wild Visual Navigation

[6] G. Kahn, P. Abbeel and S. Levine, ”BADGR: An Autonomous Self-Supervised Learning-Based Navigation System,” in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1312-1319, April 2021, doi: 10.1109/LRA.2021.3057023.

[7] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[8] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv:1802.06955.

[9] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

[10] Gu, R., Wang, G., Song, T., Huang, R., Aertsen, M., Deprest, J., ... & Zhang, S. (2020). CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. IEEE transactions on medical imaging, 40(2), 699-711.

APPENDIX

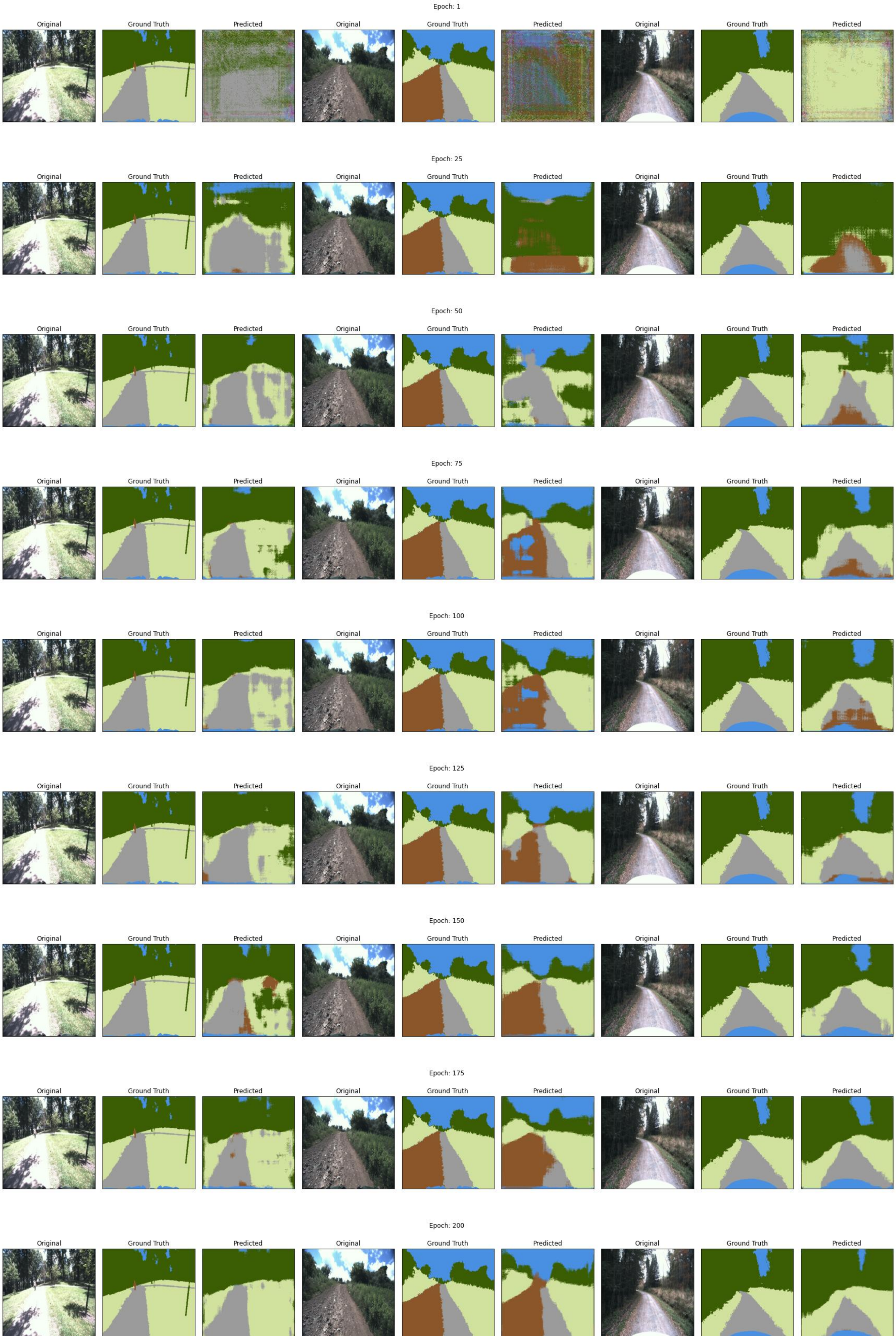
4.1. **GitHub Respository.** The implementation has been added to Github. here to visit the repository. All results are reproducible through the publicly available code.

4.2. **PSPNet Implementation.** The architecture details for Pyramid Pooling with FCN networks are provided below.
PSPNet is a state-of-the-art semantic segmentation model. It introduces the concept of Spatial Pyramid Pooling. The SPP layer performs feature pooling, creating consistent-length outputs. These outputs are subsequently directed into fully connected layers or alternative classifiers. We replace fully connected layers with transpose convolution layers in our model. Essentially, we aggregate information at a deeper level in the network hierarchy, situated between convolutional layers and fully connected layers. This approach helps us circumvent the necessity for cropping or warping at the initial stages. It aggregates coarse features at the initial stages and finer features at the latter stages, and concatenates them to create a vector of features.
Our architecture starts with multiple convolution layers, followed by the Pyramid Pooling Layer that creates Max Pooling outputs using four sizes of kernels (1, 3, 5, and 7). These are concatenated and act as input to the upsampler network that up-samples the images into the required shapes and reconstructs them to get a segmented image of size 300x300 and 6 channels corresponding to the one-hot encoding of every class.

Layer	Kernel Size	Input Shape	Output Shape
conv1	7	3x300x300	96x227x227
conv2	7	96x227x227	256x221x221
conv3	5	512x219x219	256x221x221
conv4	5	256x221x221	512x217x217
conv5	3	512x217x217	1024x217x217
spp	1, 3, 5, 7	1024x217x217	5120x217x217
up1	3	5120x217x217	256x217x217
up2	5	256x217x217	96x219x219
up3	9	96x219x219	61x251x251
up4	8	61x251x251	6x300x300

TABLE 2. Architecture details for Pyramid Pooling with FCN. The shapes for the input and output for each layer are in the format: (Number of Channels, Height, Width)

4.3. **Epoch-Wise Model Improvement.** Improvements in the training can be observed epoch-wise in the images 5.


Epoch: 1
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 25
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 50
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 75
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 100
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 125
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 150
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 175
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Epoch: 200
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted
Original
Ground Truth
Predicted