A Comparison Study on Semantic Segmentation Architectures for Ground Traversability Estimation

Suzanne Ong
University of Adelaide
Australia
suzanne.ong@student.adelaide.edu.au

Dr Feras Dayoub
University of Adelaide
Australia
feras.dayoub@adelaide.edu.au

ABSTRACT

Ground traversability estimation is an important ability for robotic appliances, particularly mobile ground robots, to predict the path that is traversable. It involves the analysis of the area traversability on the ground, which can be learned from the images classified via semantic segmentation. In this paper, we review the effect of semantic segmentation approaches available in ground transversability estimation and compare them based on the performance of the model. It aims to evaluate the performance within the selected range of existing frameworks while implementing in a simulation of real-world elevation dataset.

1 INTRODUCTION

Ground traversability estimation is the ability of autonomous, mobile appliances to articulate a traversable path to reach a designated target position. Depending on the elevation, slope and obstacles on the ground, it relies on semantic segmentation approaches, such as image classification or object detection, to partition areas that are either traversable and non-traversable (i.e. when faced obstacles like walls or static objects) for the appliances to move accordingly [1]. Traditionally, it is done by utilising sensors such as LIDAR to detect and identify the areas.

However, the complexity increases when it comes to uneven ground scenarios, because there are more constraints to consider other than the partitioning of areas based on images. For instance, given an offroad robot with a limited level of operational power, it may not be able to ascend upslope even though it is the shortest path based on the plan generated by algorithm used, hence rendering it infeasible to compute. This problem is also dependent on the centre of mass of the robots, as robots with a disproportionately great height but smaller surface area may result in failure, despite having sufficient power to either ascend or descend a slope. External factors such as the ground clearance, robot orientation, zero moment point distance, force-angle stability measure, traction efficiency and distance stability measure may impact the robot optimality in traversing to an estimated path based on the image generated above ground level [11].

Recent development in semantic segmentation techniques, such as pyramid scene parsing network, has provided a research gap in exploring more optimised ways to better articulate the estimation of traversability based on the given imagery datasets. Through algorithms that categorized traversable and non-traversable areas based on the recorded footage and outliers, there have been numerous attempts to improve the performance of the process in studies over the last five years [1,6].

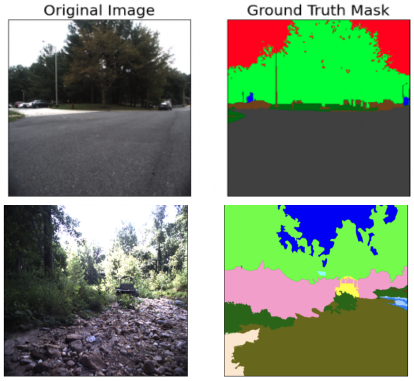


Figure 1. An illustration of outdoor scenes in RUGD off-road dataset and annotations as ground truth masks to the right, the colour code for each class can be found in the Appendix [22]

In this project, the problem statement can be stated as follows: Given a generated image representing the heightmap of a ground patch like Figure 1, identify a better approach in semantic segmentation that allows the robot to traverse from right to left with the highest accuracy. This is suggested with consideration that metrics like pixel accuracy and intersection over union, are a strong indication of an efficient algorithm, which is a favourable outcome when it is used in large datasets and limited computational resources like time, cost, and power [3,5,9]. Moreover, because each approach is built differently in terms of its structure and approach towards each pixel of the images, results may vary in performance depending on the robotic appliances used and diverse labelling methods on images from the dataset [6] and examples given in Figure 1. Hence, to observe the optimal approach from current findings, we reviewed three publicly retrievable, state-of-the-art semantic segmentation architectures to determine the technique that yields the best performance in semantic segmentation for ground transversability estimation. Based on the benchmark datasets, the configurations would be calibrated according to the implementation of images taken using the Robotic Operating System (ROS), particularly in off-road robotic applications used at a constant set of imagery dataset to address the problems faced in ground traversability problems. Added with the lack of comparison study to analyse the performance for these architectures for off-road dataset, it offers a research gap for us to estimate the ground traversability in both simulations and real-world robotic appliances [1,4,17]. Hence, for this paper, we would be focusing on finalising two concepts, which are, identifying the semantic segmentation approaches that obtains the best performance using metrics like the average mean intersection of union, pixel accuracy in detecting each label and overall loss after training the model, and secondly, observe the difference in performance for newly developed approaches with alteration in its optimisation algorithms in the form of optimizers.

2 MOTIVATIONS

The motivation for this paper is to extend the literature of available semantic segmentation techniques on ground transversability estimation, inspire future research in addressing any identified challenges, and making a comparison study for the community as an additional reference for testing these approaches based on their performance in the model.

Our research scope would be focusing primarily on testing up-to-date semantic segmentation techniques that are preferably open-sourced, configurable in previous research and can be implemented in our research environment. Although there is a boost in research implementation for semantic segmentation techniques in practical applications such as autonomous driving prediction [5, 10], up to date evaluation and analysis on traversable vegetation still holds theoretical significance in deciding the optimal technique in semantic segmentation to use for their respective fields due to the introduction of improved algorithms that helps mitigate the effects of existing limitations like outliers and roughness of the terrain [4]. The need to benchmark these approaches is further emphasised with the recent development of image classification approach suggested in the 2018 conference [1] and scene recognition approach by Matsuzaki that utilised unsupervised domain adaptation for semantic segmentation to navigate through traversable plants to navigate the robot's path based on the geometric information of the environment [9]. The development of new frameworks and usage of enhanced methods in other fields like autonomous driving that are optimized in efficiency, simplicity, and accuracy would be an interesting research aspect that may generate optimal solutions for problems faced in ground traversability estimation.

Moreover, the idea of comparing different semantic segmentation methods from published frameworks, especially in the overall computation demand, would provide us with a better insight into the runtime, quality and performance of the algorithms used in each framework. For instance, if we have an imagery dataset of 5.4GB images and environment footage, will there be any impact on the predictions generated using different semantic segmentation networks, its impact when alternate optimisers are used while training the model and overall performance? Another extension of this idea would be the effect of qualitative parameters on the quality or prediction behaviour of the algorithm, which would be useful in identifying the constraints of the selected approach on the estimation problem.

The obligation to implement a simpler, accessible setup for the given frameworks stems from the difficulty faced when generating a simulation for the architectures given due to hardware limitation. The implementation in recent frameworks, such as the frequent usage of CUDA and GPU dependent operation and workers in recent frameworks, the experimental setup may not be feasible for public usage as it is an approach niched for external GPU usage, for instance, NVIDIA. Therefore, the search for alternate workspaces and configurations for better accessibility also serves as a purpose throughout the research.

3 LITERATURE REVIEW

A review of research papers was conducted to explore the recent development of semantic segmentation techniques, the improvements, and limitations of at least three used methods on estimating ground transversability and identifying any available datasets that allow us to test and benchmark the compute budget of different semantic segmentation approaches.

Semantic segmentation is the construction of systems that process, perceive, and evaluate visual data artificially in the form of pixels and defined classes. It is widely used in various fields due to its ability to interpret visual information based on image and video data for further analysis through statistical modelling. Its purpose to translate visual data into insights readable for humans based on contextual inputs provides us with more detailed understanding to make informed decisions in business or solutions for complex real-world problems. Through accumulating a training dataset of labelled images, we provide them as input to the computer for data processing. Some of semantic segmentation applications included autonomous driving, allocation of vegetation and network security [20]. In this paper, we identified three semantic segmentation techniques, which are image classification, semantic segmentation and object detection that provided well-documented literature with open-sourced software frameworks to implement, as well as alternative methods that have not been used, but has a feasible potential to be conducted into ground traversability estimation.

Serving as a well-known approach in estimating off road navigation from visual perspective would be semantic segmentation. Given that pixel-wise predictions from footage can be conducted from models, semantic segmentation is another main approach in distinguishing the images into pixel grouping, which not only detects the land cover classes like image classification, but also classify all image pixels to determine whether if there is an object detected that would hinder the traversability on the ground. Recent studies have exploited the use of both a semantic segmentation branch for object classification and a traversability estimation branch that operated onto the pixels. In plant-rich environments, researchers managed to train the semantic segmentation branch using an unsupervised domain adaptation method and the traversability estimation branch using label images generated from the robot's traversal experience for the data acquisition phase. This resulted in the robot capability to recognise traversable plants with better accuracy than a conventional semantic segmentation with traversable and non-traversable plant classes [9]. A further extension of this approach would be the addition of Transformers with lightweight multilayer perception (MLP) decoders, which mitigates the complexity in decoders and effect of interpolation of positional codes, both that may lead to decreased algorithm performance while operating on training and testing sets. Experimented in a convoluted networks and architectures such as U-Net, or improved implementations such as SegFormer in 2021 that made used of nested transformers for a dataset compiled from urban street scenes and managed to yield a 84% increase at best in mean intersection of union (mIoU), thus suggesting that optimization using transformers improves the efficiency and compute budget of the approach [18].

In addition, by improving the image classification step in semantic segmentation, it offers potential on amplified performance during the process that involves the categorizing and labeling of land cover classes into groups of image pixels based on one or multiple desirable characteristics. By casting the problem of estimating ground traversability as a supervised image classification problem on the dataset, we can categorize the traversable areas for the robots to plan out a path with the lowest compute budget. This is backed by Chavez-Gracia's research, which compares the difference in accuracy and area under the Receiver Characteristics Operator (ROC) curve between two image classification techniques. The first option, which is the feature-based approach, extracted descriptive features such as the average terrain steepness for each heightmap patch based on the motion direction of the robot and the maximum height of any steps in the patch using the Histogram of Gradients (HOG). This is followed by the application of random forest classification with 10 trees onto the HOG computed. Alternatively, the second option involved using Convolutional Neural Networks (CNNs), another architecture that utilises the networks to categorize the data when images are inputted. In the CNN-based approach, it is expected that the network autonomously learns meaningful, problem-specific features; because the input shape is high-dimensional and no prior knowledge of the problem is provided to the model, this approach requires more training data. Using software like Keras to build the networks, Adadelta optimizer to reduce the cross-entropy loss through training for 50 epochs and Tensorflow to operate the frontend integration, the researchers were able to implement a connected layer with two output neurons [1]. Although it is being constrained the fact that it was only experimented in V-REP simulator using heightmaps, and its lack of consideration of other factors, such as the compactness, friction and robot dynamics, there is a research gap available for us to implement these approach in robot-centric perceptions to provide a detailed research literature over the implication of image classification approaches onto robotic appliances, and determine the validity of Chavez-Garcia's result in ground transversability estimation based on the underlying algorithm performance. As a result, through evaluating the performance metrics, the research suggested that CNN approach has a better performance than the feature-based approach, which allows additional room for subsequent research to compare both approaches based on the compute budget that is not specified in the paper and determine whether CNN approach is computationally feasible despite its better algorithm performance than the feature-based approach [1].

Recent improvement in semi-supervised frameworks, such as ResNet and its updated version known as VuNet [7] , has been experimented and documented for ground transversability estimation. By enhancing deep generative model, known as Generative Adversarial Network (GAN) that assist in categorizing images taken from a fish-eye camera based on their transversability, GoNet developers provided images that depicted traversable areas as positive examples for robot navigation to train GAN, with a disproportionately low number of non-transversal areas compared to the positive examples. Such discrepancy in the size of both positive and negative examples in data allows GoNet's underlying learning algorithm to exceed both supervised and unsupervised baselines in terms of its practical robustness [6]. Demonstrated by capturing images from fisheye cameras attached in a robot, it significantly improves the accuracy of robot navigation through predicting risky paths, such as collisions with an immobile object like a wall based on the negative examples learned and minimizes the probability of robots being damaged due to collisions [6]. Hence, it is suggested that GoNet framework offers the opportunity to be improvised and used in a communal setting for healthcare systems, network security and the establishment of a system to issue warnings when an obstacle is detected for the refinement in ground traversability estimation for robot navigation. The experimental method of GoNet's research also presents an opportunity to compare its overall accuracy in compute budget on outdoor door settings, as it was only limited to indoor campus environments where the impact of external obstacles like plants and vegetation on ground is negligible. This approach is further improvised with the update of VuNet framework, which handles scene view synthesis problems by making predictions on future images using RGB images for static and dynamic environments [7].

Besides reviewing on recently developed semantic segmentation architectures, there are two notable mentions in semantic segmentation that may be useful in resolving the ground traversability estimation problem, which are object detection. Object detection algorithms, such as Faster R-CNN developed in 2015, provided a well-tested, open-source improvement on CNN approach, which allows a separate network to predict the region proposals and followed by predicting the offset values through reshaping of the pooling layer. This significantly improves the traditional selective search approach, and unlike the YOLO, or You Only Look Once algorithm [9,13], since it uses regions to localize the object, it is not constrained by small obstacles detected in the image. Through leveraging statistical techniques like cluster analysis [3], it is able to estimate the ground transversability in accordance with the class probabilities generated with greater efficiency.

Such research correlates with our goal to focus on the accuracy of visual perception, the leverage over the performance and efficiency for robot navigation. Given that proprioceptive-based methods are implemented to gather frequency domain vibration information by the robot sensors while generating off-road dataset, this can serve as an alternate benchmark for the problem. However, these methods require the robot to navigate through the region to collect data and assume that a mobile robot can navigate over the entire terrain, which resulted in a low number of compatible datasets to benchmark upon when it dealt with off-road vegetation or terrain. The geometric-based methods [19], [22] generally use Lidar and stereo cameras to gather 3D point cloud and depth information of the environment. This information can be used to detect the elevation, slope, and roughness of the environment as well as obstacles in the surrounding area. than other CNN-based methods but are computationally expensive. Therefore, there has been increased discussion of more efficient design based on transformers for fast inference time and lower computational cost, including. While most of these architectures work well on structured datasets like Cityscapes and PASCAL, they do not work well for off-road datasets due to ill-defined boundaries and confusing class features.

4 METHODOLOGY

A review of research papers was conducted to explore the recent development of semantic segmentation techniques, the improvements, and limitations of at least three used methods on estimating ground transversability based on two varied optimizers and identifying any available off-road dataset that allow us to test and benchmark the performance of different semantic segmentation approaches. For the start, we will discuss the implementation of three used architectures, followed by the definition and impact of both optimizers towards the experiment stated in Table 1.

A. Pyramid Scene Parsing

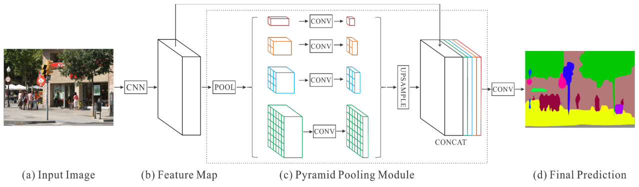


Figure 2: A basic implementation of PSPNet on semantic segmentation [26]

As a network built with the focus to improve scene parsing for unrestricted open vocabulary and diverse scenes, the Pyramid Scene Parsing, or PSPNet, has . Tried successfully by Zhao's research team in 2016, it exploited the capability of global context information via different-region-based context aggregation. The network proceeds to maximize the implementation of the pyramid pooling module for context information gathering, followed by the concatenation of the original feature map and generation of the prediction map in accordance with Figure 2. Upon closer inspection on cited research and external integration of PSPNet, it had produced good quality results on the scene parsing task, while PSPNet provides an 80.2% mean IoU for pixel-level prediction tasks when it is benchmarked in Cityscape [15] , thus suggesting a significant improvement compared to 68% from other CNN approaches like Faster-SCNN [13].

B. Deep Lab V3

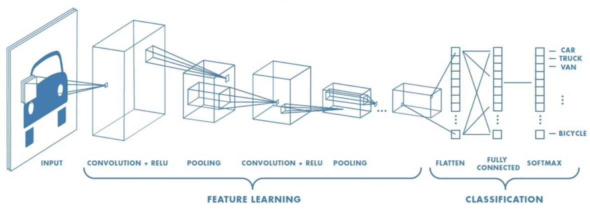


Figure 3: A basic implementation of Deep Lab V3 on semantic segmentation [8]

In Deep Lab V3's architecture, convolutions with up sampled filters are used for tasks that involve dense prediction. Segmentation of objects at multiple scales is done via atrous spatial pyramid pooling. Finally, DCNNs are used to improve the localization of object boundaries. Atrous convolution is achieved by up sampling the filters through the insertion of zeros or sparse sampling of input feature maps. Usually, classification DCNNs have four main operations, which are convolutions, activation function, pooling, and fully connected layers. By passing an image through a series of these operations outputs a feature vector containing the probabilities for each class label. Notice that by pointing out that in this setup, the network categorizes an image as a whole and assigns a single label to an entire image pixel-wise. This model could output a probability tensor with shape [W, H, C], where W and H represent the Width and Height. And C the number of class labels. Applying the argmax function (on the third axis) gives us a tensor shape of [W, H,1]. After, we compute the cross-entropy loss between each pixel of the ground-truth images and our predictions. In the end, we average that value and train the network using back prop. For better computation time and memory allocation, instead of having the three main components, which are convolutions, down sampling, and up sampling layers, DeeplabV3 offers an architecture for controlling signal decimation and learning multi-scale contextual features. It uses an ImageNet pre-trained ResNet as its main feature extractor network and proposes a new Residual block for multi-scale feature learning. Hence, instead of regular convolutions, the last ResNet block uses atrous convolutions as shown in Figure 3. Also, each convolution uses different dilation rates to capture multi-scale context, in which on top of this new block, it uses Atrous Spatial Pyramid Pooling (ASPP) to get dilated convolutions with different rates in order to classify regions of an arbitrary scale.

C. Fully Convoluted Network

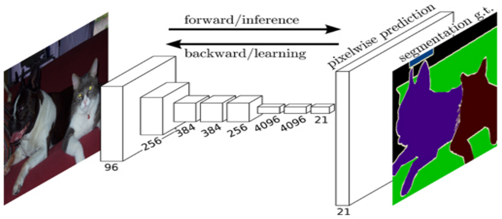


Figure 4: A basic implementation of FCN on semantic segmentation [12]

Convolutional networks are powerful visual models that yield hierarchies of features. Presented as trained models for end-to-end, pixels-to-pixels, the research in 2015 improved on the previous best result in semantic segmentation [8] with optimized networks. Taking input of arbitrary size, it produces correspondingly sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. Adapting existing networks, which in this experiment, would be VGGNet, into fully convolutional networks and transfer their learned representations by fine-tuning to the segmentation task. Illustrated in Figure 4, we then define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations.

D. Optimisers

Over the recent years, optimization algorithms have been improved and widely used in CNN-based models by utilizing gradient descent to lower the error rate of the training process and to reform the hyperparameters. The information generated guides the error function to alter the downward to the local minimum. The orthodox batch gradient descent technique computes a gradient of the training data, which resulted in slow computation in training. Therefore, the usage of optimization algorithms in the form of optimizers are highly sought after.

We utilize the usage of two optimizers, which are the Adaptive Momentum (Adam) optimiser and the Stochastic Gradient Descent (SGD) with a momentum of 0.9 for the given CNN-based architectures. Adam has gained much popularity in the literature of semantic segmentation [17, 21], and based on existing implementation for off road navigation dataset, there is potential to research on the usage of different optimizer algorithms simultaneously for future references to improve on the existing architectures.

For context, in Yaqub's recent comparison on optimizers, Adam generated the least computational cost, requires less memory for implementation, and is invariant to diagonal rescaling of the gradients. This takes care of the issues such as but not limited to huge data sets, hyperparameters, noisy data, inadequate gradients, and non-stationary problems that require small tuning [23, 25]. Upon further observation on semantic segmentation's development, Adam has an increased growth in popularity in training models, particularly in off road navigational datasets like RELLIS-3D and RUGD [5, 22, 24]

Another approach we have considered is the Stochastic Gradient Descent, also known as SGD and is another implementation of optimizer. By using defined steps, inputs are taken to provide the exact results while training. This would reduce the computational work according to [19] as it allows the user to best optimize the training using linear regression utilizing gradient descent. Added with additional improvisation in the optimizer by allocating momentum, it can be useful for datasets that consist of noisy gradients [25]. Momentum provides an updated rule which is inspired by the physical perspective of optimization. Imagine a ball in the mountainous area trying to reach the deepest valley, it passes through slight hills when the slope is very high, and the ball gains a lot of momentum. The speed of the ball depends on the momentum of the ball, and momentum provides a boost to speed up learning that changes very little to SGD and velocity to make the updates that store velocity for the parameters. The adapted function for SGD uses the momentum updated rule. However, while momentum is very high, the goal is very close, and we do not know how to slow down the speed. At the beginning, the oscillate minima do not reach the goal. GD has extra cure surfaces in one direction but not in the other direction. It also reduces the oscillation. For updating the weights, it takes the gradient of the current and previous time steps which move faster towards convergence. Convergence is faster when we apply the momentum optimizer to surfaces with curves.

5 EXPERIMENTAL SETUPS

A. Initial plan and design

During the weeks ranging from 1 to 3, we structured the initial layout based on the conference and rubric requirements, with additional revision on the available semantic segmentation techniques used and conducted thorough research on existing papers. A variety of benchmarked datasets available are reviewed, as well as a quick glance through over the implications of the software package Robotic Operating System (ROS) over off-road dataset and as a stretch, robotic applications. Added with the provision of time for the construction of the design document draft, Week 4 provides time to address additional feedback and evaluation over the feasibility of the project.

With these preparations on hand, week 5 primarily focused on downloading the elevation dataset and constructing the semantic segmentation approaches and training the models used in code using Python, PyTorch and Google Colab. A buffer period in Week 6 was scheduled to update the progress into the final report, address the lesson learned or arising problems throughout, and discuss the need to implement additional approaches or evaluation metric depending on the progress. We also evaluated the results obtained, recorded any findings or additional resources, and finally reviewed the feasibility of implementing any stretch goals identified between Week 7 and Week 11. The project is finalised with the construction of the final report, uploading of trained models and the delivery of the presentation on Week 12 and 13.

The completion of the project was defined by the successful construction of the final report and compliable algorithms stored in a GitHub repository in Week 12, followed by constructive feedback sessions from the supervisor and members of the public selected from University of Adelaide through a finalised presentation in Week 13. However, this project presents us with three challenges, which are the possibility of negative results, lack of on-hands experience in ROS or robotic appliances and the vast size of the dataset, that may hinder progress. As we have yet to conduct much experiment or usage on the proposed frameworks, there may be

Approach	Architecture	Optimizer
1	Res 50 with parallel stacking layer (ResNext 50) + PSPNet	Adam
2	Res 50 with parallel stacking layer (ResNext 50) + PSPNet	SGD
3	Res 50 + DeepLabV3	Adam
4	Res 50 + DeepLabV3	SGD
5	Res 50 + Fully Convoluted Network (FCN)	Adam
6	Res 50 + Fully Convoluted Network (FCN)	SGD

Table 1: Compared architectures for semantic segmentation

indications that a negative result may be obtained regardless of the techniques' performance. The lack of insights of the ROS software and large imagery dataset may also require additional learning time and storage. Therefore, it is necessary to implement contingency planning for all challenges faced, which is known to be a good practice for any project design. By implementation, in a scenario where a negative result is obtained, it should not be disregarded and should be presented as a valid conclusion that provides room for additional research in the future. The practice of cherry-picking results in the project should not be practised as well to maintain the level of accuracy and consistency of the project. A buffer of at least one week in the timeline for adequate time allocation, and an external hard-disk plug-in is used as an alternative source for storage to address the challenges.

B. Task Structure

We use the dataset taken from RUGD, which consisted of 24 colour labelled classes alongside pixel-wise annotations for ground truth comparison [22]. Optimization is done using both Adam and stochastic gradient descent optimizer with a learning rate of 0.0001 and a decay of 0.001. We adopt the polynomial learning rate policy with a power of 0.9. We augment the data with horizontal random flip, vertical flip, and random crop. Implementing a batch size of 20 and a crop size of 304 × 304, the training and validation sets are used to generate an optimized model through the proposed architecture, followed by obtaining the performance metric by training the model on a separate test set. The total number of frames consists of 7,453 images, each attached with densely pixelated annotations to be implemented as the growth truth. Assigning three different models with a pretrained Res50 Net backbone, an encoder that is convolutional neural network that is 50 layers deep, the main purpose of this research is to compare the mean pixel accuracy, mean IoU and F1 score of each model with two different optimizers.

During training, due to the time intensively and limited RAM usage in Google Colab, only a range of 1 to 10 epochs are tested on the architectures for all allocated images from the dataset. Limitation in GPU RAM and training time due to Google Colab's existing user policy has minimized the use of larger batch sizes and epoch iteration for further testing, which is considered as a limitation and may be addressed in future research. We also avoid using a number beyond 50 for epochs to avoid the risk of overfitting the models, which may cause an undesirable increase in test error and accuracy as the model begins to memorise the training set if the learning rate and model hyperparameters are too small. This will undermine the model's functionality to predict and segmentate the image pixels based on the classes defined. Hence, to mitigate such risk, we regularise the model by passing the model on the allocated test dataset that are absent during the training, and implement the early stopping technique, which is defined by conducting a learning rate scheduler that stops training once the test accuracy stopped improving. These methods allow us to monitor the test accuracy during training, providing us with better accuracy in performance evaluations throughout the research. Utilizing Google Colab Pro's inbuilt NVIDIA GPU for model training and environment setup, we generate the results based on the following procedures:

1. Install the required libraries for framework compilation and download the required dataset.
2. Pre-process dataset:
    a. Initialise dataset and for each pixel, patchify masks(annotations) into one hot encoder format.
    b. Resize each image and split datasets to train, valid and test groups by 60%, 20% and 30% of the dataset respectively.
    c. Normalize each image on the dataset.
    d. Define labels and the RGB values of each class.
3. Initialise data loaders for train, valid and test datasets.
4. Establish and defining model architectures with pretrained weights that are publicly available using PyTorch and compiled in Jupyter Notebooks.
5. Establish Optimizers, Learning Rate Scheduler, and focal loss as the loss function.
6. Train the model using the train and valid data loaders in 5 epochs
7. Validate the model for accuracy on the test data loader that consists of unseen images
8. Evaluate the performance based on 3 main performance metrics: mean IoU, pixel accuracy and F1 score. Other performance metrics such as precision and recall may be considered as a stretch.
9. Record and present the overall finding, success or unsuccessful result in report, codebase, and presentation.

To mask the labels identified in the pixel-wise annotations provided, the experiment requires the classification of image by following an encoder/decoder structure where we down sample the spatial resolution of the input, developing lower-resolution feature mappings which are learned to be highly efficient at discriminating between classes, and the up sample the feature. representations into a full-resolution segmentation map. Throughout the experiment, it is ensured that the model, input and labels from raw frames and annotations are sent to the same device in the assigned runtime, and flexibility to use CPU, GPU or TPU are accepted and easily configured depending on the type detected from the user's local configuration. Upon clearing the gradients in the optimizer before the forward or backward propagation for optimized memory allocation on each iteration, we first train the training dataset, followed by the valid dataset to define the checkpoints and models. For inference, we set the model to evaluate the network adjustment before passing through the test dataset to check on variables like batch norm and dropouts to mitigate the risk of affecting the network weights.

The scope is to focus on implementing known semantic segmentation approaches for ground and terrain traversability estimation, which is semantic segmentation using both naive CNN approaches and its easier implementation using in-built libraries like TensorFlow segmentation models to conduct tests on the given dataset. However, given the complexity of the ground traversability problem, it offers numerous rooms of extension on the research. As part of the contingency planning in dataset selection and a stretch goal for upcoming implementation, the retrieval of GO Stanford 3 indoor and RELLIS-3D datasets may be applied for its previous usage in robotic appliances to test the relevance of the original results obtained. Although not fully utilized in this experiment, ROS was used to ease compatibility and the connection of the given robotic appliances and its specific sensors, which in turn offers an alternate approach to improvise the availability of off-road dataset when used in equipment like Lidar sensors.

C. Performance Evaluation

To justify the overall performance of the six architecture combinations, we benchmarked the models based on three standard segmentation metrics, which are mean pixel accuracy, mean Intersection of Union and using focal loss as the loss function, to benchmark and compare between different architectures and optimizer usage for off-road dataset.

Given that the recognition of each pixel of the image of a single class impacts the provision of accuracy of the model for classification, we detected the usage of the mean pixel accuracy as another means to evaluate performance based on the fundamental calculations, where pixel accuracy is equivalent to the percentage of pixels that are correctly classified. However, the illustration of high pixel accuracy can be affected by class imbalance, which is a recurring issue in the dataset due to the unequal presence of classes such as "sky" or "grass" compared to other labelled classes. Therefore, the performance evaluation on the models is further assessed using alternate metrics like the intersection over union, or IoU.

Positively correlated to the Dice coefficient, the mean IoU provides the ability to quantify the overall percentage of the image that overlaps between the masks from annotations and the predicted output. It is calculated according to the number of pixels that share commodity between the ground truth and predicted masks of the image divided by the total number of pixels that are present in both masks. Averaged over all classes, we can obtain the mean IoU score for each model.

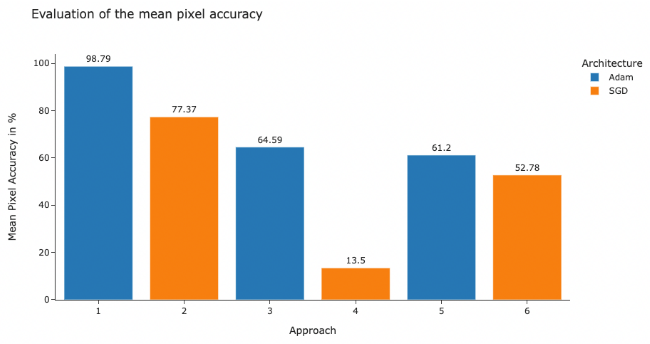
Depending on the precision and recall of the images, F1 score is also considered to be an ideal metric to evaluate the connectedness of the pixels based on the true and predicted values. It combines and balances both the precision and recall of the ground truth and predicted images to help evaluate any imbalance data, and a score of 1 can be regarded as the optimal for F1 score as it indicates that the model achieved perfect accuracy.

To address the class imbalance issue faced, as well as classifying more accurate positives and negatives within the images, we implemented the focal loss as the loss function of the models. As an extension of cross-entropy loss function, it first scaled based on the scaling factors decaying at zero as the confidence level in each classes increases, down-weight identified pixels that are easier to be identified and focus on training the hard negatives, which can be a better predictor in multi-class semantic segmentation in off-road implementation, where detection on certain classes like "bicycle" are more difficult than the remaining classes due to its rarity compared to other classes [5].

6 RESULTS

A. Analysis

In this section, we focus primarily on comparing the six architectures mentioned with design and configuration like GaNav experimental setup, with alteration in image size and epoch iteration to correlate with Google Colab's environment. The mean pixel accuracy of the study are presented in Figure 5, followed by the mean IoU and Focal loss in Figure 6 and 7 respectively.



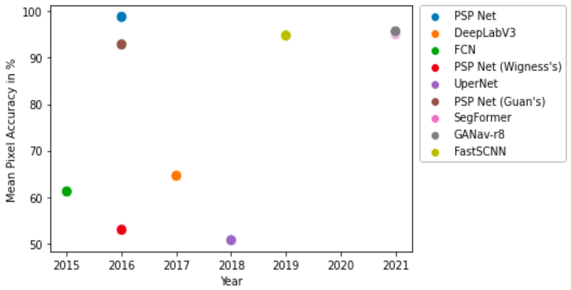


Figure 5: Mean pixel accuracy for tested approaches, followed by a scatter plot to compare our results from Adam optimizers with benchmark results from Guan's and Wigness's research shown in Table 2 in the Appendix [4,22]. The approaches are numbered based on the architectures stated in Table 1, with odd and even-indexed approaches to be classified as architectures with Adam and SGD respectively. The detailed values of the mean pixel accuracy can be found in the Appendix

Upon observation, PSPNet architecture with Adam conveyed the highest percentage of mean pixel accuracy 98.79% and by comparing the mean pixel accuracy between Adam and SGD, it is suggested that Adam optimizers have outperformed its SGD's counterpart in semantic segmentation, which conforms to the conclusion taken in Yaqub's research [25]. Alternatively, in comparing the pixel accuracy with other research, PSP Net and SegFormer have outperformed the architectures in mean pixel accuracy.

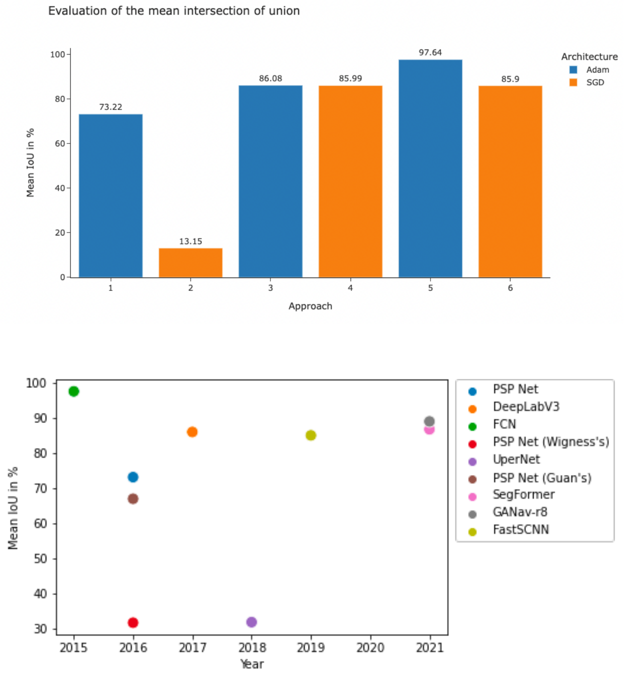


Figure 6: Mean IoU for tested approaches, followed by a scatter plot to compare with benchmark results from Guan's and Wigness's research shown in Table 2 in the Appendix [4,22].

Through comparison, it is identified that FCN architecture using Adam optimizers convey the highest percentage of mean IoU at 97.64%. In addition, by comparing the mean IoU between Adam and SGD optimizers, it is suggested that Adam optimizers have outperformed SGD in semantic segmentation, which again, supports the conclusion taken in Yaqub's research [25]. However, in comparing the mean IoU with other research, the use of FCN has outperformed all architectures in mean pixel accuracy, which suggested that FCN may serve as a better approach in segmentation of off-road datasets compared to recently developed networks that utilises transformers or group-wise classification, which correlates with the accurate presentation in predicted images in Figure 9 of the Appendix.

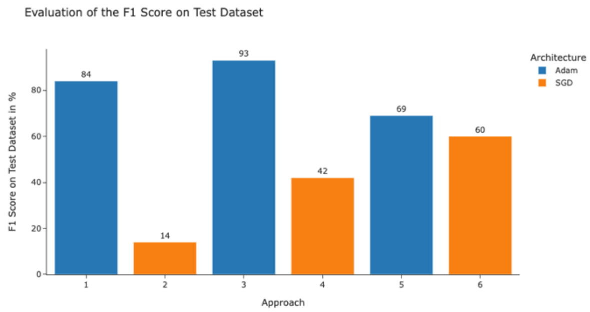


Figure 7: F1 Score on testing datasets. We can observe that the PSPNet attached with a SGD optimizer and Res50 backbone consists of a considerably highest loss, while DeepLabV3 with Adam's approach has the best model given that it provides a score of 93%

Given that RUGD dataset provides us with an unbalanced percentage of occurrence for each class [22], the DeepLabV3 approach with Adam offers the minimal loss among the six architectures at a F1 score of 93% despite the sparsity of classes such as "bicycle" compared to more common classes like "tree", hence making it a critical consideration as we evaluate the F1 scores. There is also a significant reduction in F1 score between Adam and SGD implementations, which strongly suggested Adam's higher feasibility in off-road semantic segmentation, but it may also indicate a potential bias in the experiment in which further testing is needed to ensure the accuracy of SGD's weak performance.

B. Code implementation

The following architectures are released in GitHub based on Google Colab. ipynb formatted notebooks, along with trained models to be used as checkpoints. The results is updated in the attached link below, with codes for visualisations and stretch goal implementations to be released in the near future:

https://github.com/SuzanneOngCodes/Semantic-segmentation.git

7 CONCLUSIONS

We have proposed a comparison study for semantic segmentation models based on off road navigation dataset taken from RUGD research [22]. For this research, a total of six combinations based on the difference in both architecture and optimizers are evaluated on the given dataset, with the complete list stated in Table 1.

Overall, upon closer inspection of the results, we can conclude that according to benchmarked results, there are significant differences between our research outcome and the results obtained from other researchers, hence prompting the need for additional research. Moreover, based on performance discrepancy between Adam and SGD with 0.9 momentum optimizers, with lower mean pixel accuracy, IoU and F1 score compared to the Adam's counterpart, there are strong indications that Adam optimizer has outperforms SGD in multiclass semantic segmentation, hence further supported the conclusion stated from Yaqub's research [25].

Taken by inspiration via the exponential growth in research over numerous approaches such as GaNav and MemSeg, further work can be reviewed using different datasets, such as RELLIS-3D and potentially test run in simulations. For a start, there are four ways in which the research can progress further, which are, i) the implementation of other existing offroad dataset with the use of transformers ii) The collection of additional datasets using ROS software and LIDAR sensored annotations [24] for benchmarking and iii) Additional testing on different encoders and architectures for improved solutions once published and iv) Potential improvement on architectures in configured breakpoints.

REFERENCES

[1] Chavez-Garcia, RO, Guzzi, J, Gambardella, LM & Giusti, A 2017, 'Image classification for ground traversability estimation in robotics', in International Conference on Advanced Concepts for Intelligent Vision Systems, Springer, pp. 325-336.

[2] Eriksson, D & Harström, J 2019, 'Object detection by cluster analysis on 3D-points from a LiDAR sensor'.

[3] Guan, T, He, Z, Manocha, D & Zhang, L 2021, 'TTM: Terrain traversability mapping for autonomous excavator navigation in unstructured environments', arXiv preprint arXiv:2109.06250.

[4] Guan, T, Kothandaraman, D, Chandra, R, Sathyamoorthy, AJ & Manocha, D 2021, 'Ganav: Group-wise attention network for classifying navigable regions in unstructured outdoor environments', arXiv preprint arXiv:2103.04233.

[5] Guastella, DC & Muscato, G 2020, 'Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review', Sensors, vol. 21, no. 1, p. 73.

[6] Hirose, N, Sadeghian, A, Vázquez, M, Goebel, P & Savarese, S 2018, 'Gonet: A semi-supervised deep learning approach for traversability estimation', in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 3044-3051.

[7] Hirose, N, Sadeghian, A, Xia, F, Martín-Martín, R & Savarese, S 2019, 'Vunet: Dynamic scene view synthesis for traversability estimation using an rgb camera', IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 2062-2069.

[8] Jin, Y, Han, D & Ko, H 2021, 'Trseg: Transformer for semantic segmentation', Pattern Recognition Letters, vol. 148, pp. 29-35.

[9] Jin, Y, Han, D & Ko, H 2021, 'Memory-based Semantic Segmentation for Off-road Unstructured Natural Environments', in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 24-31.

[10] Kiran, PSR, Kumar, A & Mohan, R 2019, 'Aerial-Ground Robotic system for Terrain estimation and Navigation', in 2019 Fifth Indian Control Conference (ICC), IEEE, pp. 101-106.

[11] Matsuzaki, S, Masuzawa, H & Miura, J 2022, 'Image-based scene recognition for robot navigation considering traversable plants and its manual annotation-free training', IEEE Access.

[12] Mou, L, Hua, Y & Zhu, XX 2019, 'A relation-augmented fully convolutional network for semantic segmentation in aerial scenes', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12416-12425.

[13] Onozuka, Y, Matsumi, R & Shino, M 2021, 'Weakly-supervised recommended traversable area segmentation using automatically labeled images for autonomous driving in a pedestrian environment with no edges', Sensors, vol. 21, no. 2, p. 437.

[14] Papadakis, P 2013, 'Terrain traversability analysis methods for unmanned ground vehicles: A survey', Engineering Applications of Artificial Intelligence, vol. 26, no. 4, pp. 1373-1385.

[15] Prágr, M, Čížek, P & Faigl, J 2018, 'Incremental learning of traversability cost for aerial reconnaissance support to ground units', in International Conference on Modelling and Simulation for Autonomous Systems, Springer, pp. 412-421.

[16] Ren, S, He, K, Girshick, R & Sun, J 2015, 'Faster r-cnn: Towards real-time object detection with region proposal networks', Advances in Neural Information Processing Systems, vol. 28.

[17] Ross, PJ 2016, 'Vision-based traversability estimation in field environments', Queensland University of Technology.

[18] Sevastopoulos, C & Konstantopoulos, S 2021, 'A Simulated Environment for Traversability Estimation Experiments in Field Robotics Applications', in The 14th PErvasive Technologies Related to Assistive Environments Conference, pp. 256-257.

[19] Shan, T, Wang, J, Englot, B & Doherty, K 2018, 'Bayesian generalized kernel inference for terrain traversability mapping', in Conference on Robot Learning, PMLR, pp. 829-838.

[20] Sun, P, Kretzschmar, H, Dotiwalla, X, Chouard, A, Patnaik, V, Tsui, P, Guo, J, Zhou, Y, Chai, Y & Caine, B 2020, 'Scalability in perception for autonomous driving: Waymo open dataset', in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2446-2454.

[21] Viswanath, K, Singh, K, Jiang, P, Sujit, P & Saripalli, S 2021, 'Offseg: A semantic segmentation framework for off-road driving', in 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), IEEE, pp. 354-359.

[22] Wigness, M, Eum, S, Rogers, JG, Han, D & Kwon, H 2019, 'A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments', in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 5000-5007.

[23] Wu, Z, Shen, C & Van Den Hengel, A 2019, 'Wider or deeper: Revisiting the resnet model for visual recognition', Pattern Recognition, vol. 90, pp. 119-133.

[24] Xie, E, Wang, W, Yu, Z, Anandkumar, A, Alvarez, JM & Luo, P 2021, 'SegFormer: Simple and efficient design for semantic segmentation with transformers', Advances in Neural Information Processing Systems, vol. 34.

[25] Yaqub, M, Feng, J, Zia, MS, Arshid, K, Jia, K, Rehman, ZU & Mehmood, A 2020, 'State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images', Brain Sciences, vol. 10, no. 7, p. 427.

[26] Zhao, H, Shi, J, Qi, X, Wang, X & Jia, J 2017, 'Pyramid scene parsing network', in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881-2890.

[27] Zhao, J, Masood, R & Seneviratne, S 2021, 'A review of computer vision methods in network security', IEEE Communications Surveys & Tutorials.

APPENDIX

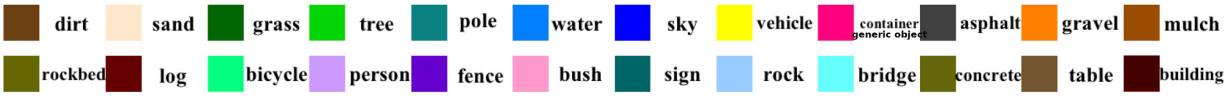


Figure 8: Names of all 24 classes defined in RUGD dataset [22]

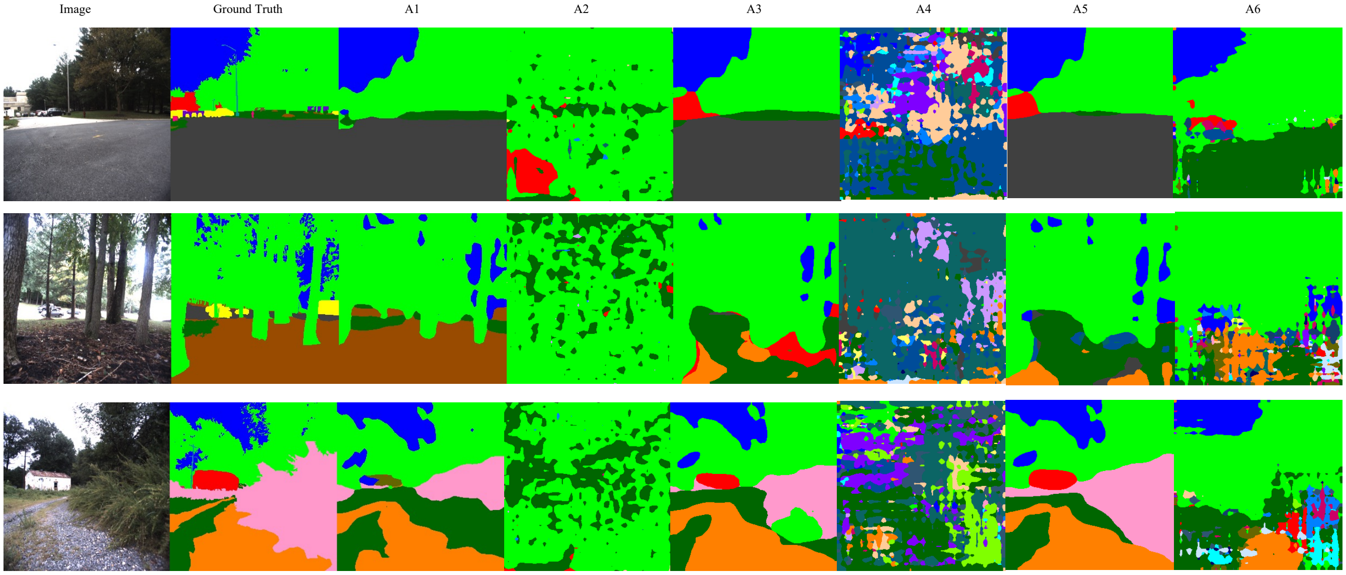


Figure 9: Predicted images for the six architectures. Upon inspection, the architectures that used Adam (A1, A3, A5) tend to have a high accuracy in detecting classes compared to SGD (A2, A4, A6), thus conforming to the lower mean pixel accuracy, IoU and F1 score.

Architectures (Res50 backbone with Adam optimizers)	**Year**	**Mean IoU in %**	**Mean Pixel Accuracy in %**
PSPNet	2016	73.22	98.79
DeepLabV3	2017	86.08	64.59
Fully Convoluted Network (FCN)	2015	97.64	61.20
PSPNet (Wigness's)	2016	31.78	52.96
UperNet	2018	31.95	50.72
PSPNet (Guan's)	2016	67.06	92.85
SegFormer	2021	86.83	95.17
GANav-r8	2021	89.08	95.66
FastSCNN	2019	85.11	94.77

Table 2: List of architectures and benchmarked networks from Guan's and Wigness's research