

1 Results

In the section we show and evaluate the models results. We will start by presenting to the reader the networks score on each metric, then we will use the best models to predict the traversability of real world terrain. Finally, we will use handcrafted patches, for example a wall of a certain height in a specific position, to test the robustness of the network by trying to highlighting its behaviour.

1.0.1 Experiment Setup

We run all the experiment on a work station with Ubuntu 18.10 operating system. The machine is equipped with a Ryzen 2700x, a powerful CPU with 8 cores and 16 threads, and a NVIDIA 1080 GPU with 8GB of dedicated RAM. For the classification task, we select a threshold of $0.2m$ on a time window of two seconds to label the patches, meaning that a patch with an advancement less than 20 centimeters is labeled as *no traversable* and viceversa. We minimise the binary Cross Entropy. On the other hand, for regression, we did not label the patch and directly regress on the advancement for a given time window while minimising the Mean Square Error (MSE). To train the network we follow the best practice on residual network [he2015deep] using Standard Gradient Descent with momentum set to 0.95 and weight decay to $1e - 4$ with an initial learning rate of $1e - 3$. We fix the maximum number of train epochs to 30 and reduce the learning rage on platuo by a factor of 0.1 with a patience of 4. We used early stopping to stop the training if the validation accuracy does not increase in 6 epochs.

fix this typo

1.0.2 Experimental validation

We select as *validation* ten percent of the training data. We remain to the reader that we store each run of *Krock* as a *.csv* file. So, to avoid any biases, we used completely different dataframes, meaning that train and validation sets are composed by non overlapping data from the simulations.

add more maps if we add them to the test set

1.0.3 Metrics

Classification: To evaluate the model's classification performance we used two metrics: *accuracy* and *AUC-ROC Curve*. Accuracy scores the number of correct predictions made by the network while AUC-ROC Curve represents degree or measure of separability, informally it tells how much model is capable of distinguishing between classes. For each experiment, we select the model with the higher AUC-ROC Curve during training to be evaluated on the test set.

Regression: We used the Mean Square Error to evaluate the model's performance.

1.1 Quantitative Results

The following table shows the final results on various test dataset made by using real-world heightmaps.

Dataset			ResNet		Size	Resolution(cm/px)
Type	Name	Samples	ACC	AUC		
Synthetic	Training	TODO	-	-		2
	Evaluation	TODO				2
Real evaluation	Quarry	TODO				2
	Quarry	TODO				
	Quarry	TODO				

Table 1: Todo

Moreover, we would like to also show the different steps we made to reach this result. The following table shows the metric's score without any data-augmentation. Adding dropout increases the results.

table with results

With dropout plus coarse dropout.

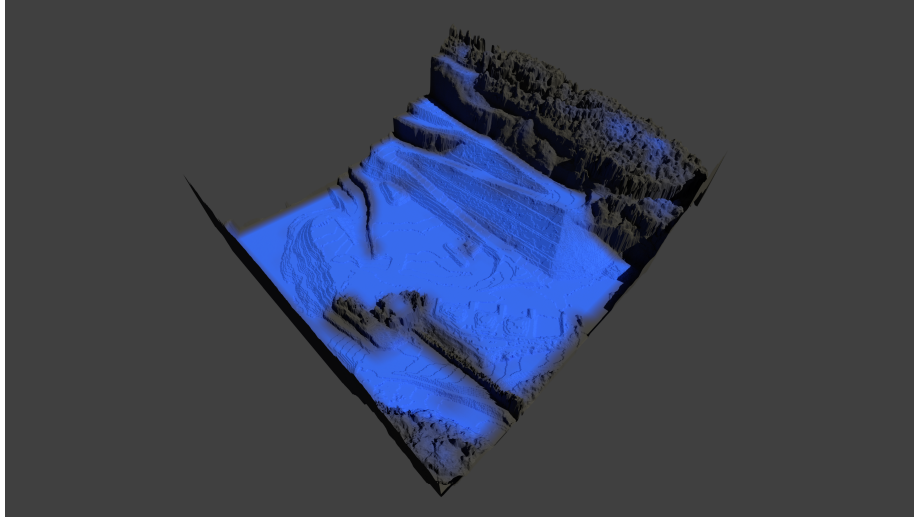
table with results

1.2 Qualitative results

We qualitative evaluate the models in real world scenarios by computing the traversability probability for each map with different rotation. Specifically, we used a sliding window to extract the patches from the heightmaps and colour by blue the relative region with the corresponding traversability probability. A brighter colour yields an higher probability. For each map we show the traversability from bottom to top, top to bottom, left to right and right to left since those are the most human understandable. We will start by showing the traversability probability on the *Quarry* assuming *Krock* is walking from bottom to top.

add quarry textures from bottom to top

Thanks its special locomotion, *Krock* can traverse the big slopes in the top part of them map while obliously it is stock by big bumps near the bottom as show in the next figures.



add figure of krock traversing the big slopes and getting stop near the bottom

To convince the reader that those slopes can be traverse, we run *Krock* on them directly from the simulator.

image of one extracted patch from quarry and one run on the simulator