0.1 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.

0.1.1 Experiment Setup

Hardware

We run all the experiment on a Ubuntu 18.10 work station equipped with a Ryzen 2700x, a powerful CPU with 8 cores and 16 threads, and a NVIDIA 1080 GPU with 8GB of dedicated RAM.

Dataset

For the classification task, we select a threshold of 0.2m on a time window, Δt, 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. This processed is explained in detail in the previous chapter. We minimise the binary Cross Entropy.

While, for regression, we did not label the patch and directly regress on the advancement minimising the Mean Square Error (MSE).

Initially, to train the models in booth cases we first use Standard Gradient Descent with momentum set to 0.95 and weight decay to $1e-4$ with an initial learning rate of $1e-3$ as was originally proposed to train residual network **?**. However, we later utilize Leslie Smith's 1cycle policy **?** that allows us to trian the network faster and with an higher accuracy.

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

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.

0.1.2 Quantitative Results

Model selection

We compared two different *micro-resnet* and the *vanilla* cnn from the previous Chapter. We evaluate those models using a time window of two second, a threshold of 20cm and the data augmentation techniques described before. We run five experiments per architecture and we select the best performing network, the results are showed in the following table.

		Vanilla	MicroResnetSE	
			3 × 3 stride 1	7 × 7 stride 2
AUC	Top	0.892	0.888	**0.896**
	Mean	**0.890**	0.883	0.888
Params		974,351	313,642	314,282

Table 1. Model comparison on the test set.

Luca told me is better to split the models like Model1 and Model2 etc

Based on this data We select *micro-resnet* with squeeze and excitation and a starting convolution's kernel size of 7 × 7 with stride of 2. This model has one third of the parameter of the origal model proposed by Chavez-Garcia et all **?**.

As proof of work, we also train the best network architecture, MicroResnetSE with a first convolution's kernel size of 7 × 7 and stride= 2, with and without the Squeeze and Excitation operator.

	MicroResnet7 × 7	MicroResnet7 × 7-SE	Improvement
Top	0.875	**0.896**	+0.021
Mean	0.867	**0.888**	+0.021

Table 2. AUC top value and mean value for MicroResnet with a fist convolution of 7 × 7 and stride = 2 with and without the SE module. The improvement is the same.

0.1.3 Final results

The following table shows in deep the score of the best network for each dataset.

I have actually never talk about surf rocks

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.

add result with and without data agu

Adding dropout increases the results.

table with results

With dropout plus coarse dropout.

table with results

Dataset			micro-resnet		Size	Resolution(cm/px)
Type	Name	Samples	ACC	AUC		
Synthetic	Training	429312	-	-		2
	Valdiation	44032	95.2 %	0.961		2
	Arc Rocks	37273	85.5 %	0.888		2
Real evaluation	Quarry	36224	88.2 %	0.896		2
	foo	TODO				
	baaa	TODO				

Table 3. Final results on different datasets.

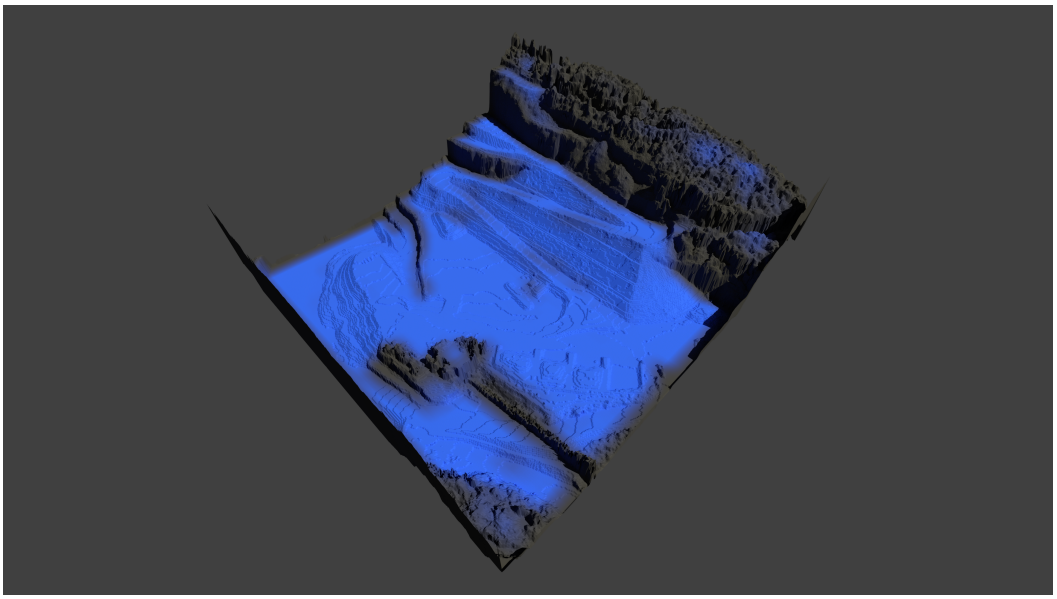
0.1.4 Qualitative results

THIS PLOT IS OLD!

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