Chapter 1

Results

In the section we show and evaluate the model's results. We will start by presenting to the reader the networks score on each metric, then we will use the best model to predict the traversability in real world terrains.

1.1 Experiment setup

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.

1.2 Dataset

To perform classification, 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. The This processed is explained in detail in the previous chapter. While for the regression, we did not label the patch and directly regress on the advancement.

Initially, to train the models 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 [**he2015deep**]. However, we later utilize Leslie Smith's 1cycle policy [**1cycle**] that allows us to trian the network faster and with an higher accuracy. We minimise the binary Cross Entropy for the classifier and the Mean Square Error (MSE) for the regressor.

1.2.1 Experimental validation

We select as *validation* set ten percent of the training data. Since we store each run of Krock as a *.csv* file, validation and train set do not overlap. The test set is composed entirely by the Quarry map, a real world scenario. Table **??** tells in detail the configuration used in each map of all sets.

We also evaluate the model on the following additonal maps

add arck rocks

1.2.2 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.

1.3 Quantitative results

1.3.1 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 1.1. 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 [**omar2018traversability**].

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 1.2. 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.

1.4 Final results

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

Dataset			micro-resnet		Size	Resolution(cm/px)
Type	Name	Samples	ACC	AUC		
Synthetic	Training	429312	-	-		2
	Validation	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 1.3. Final results on different datasets.

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. With dropout plus coarse dropout.

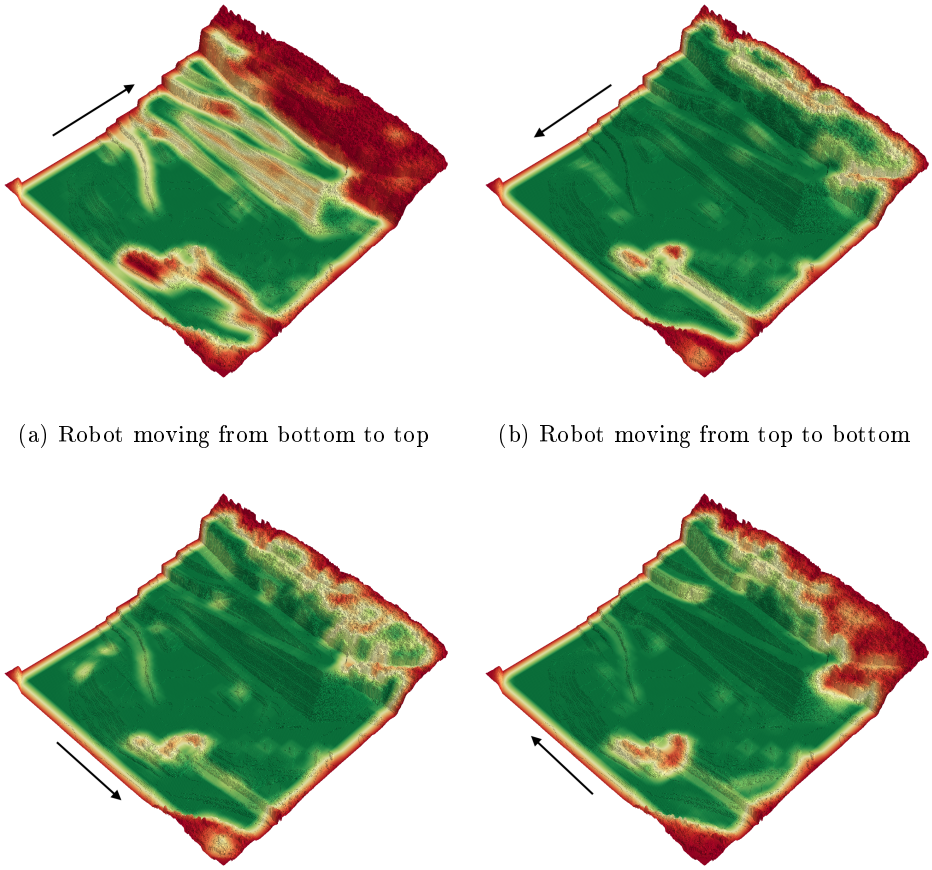
1.5 Qualitative results

We qualitative evaluate the model predictions by showing the traversability probability on different maps in 3D as a texture. Specifically, we used a sliding window to extract the patches from the heightmaps and create a texture based on the model's output for the traversable class. Then, we apply this texture on the image and colour using a colormap. 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.

1.5.1 Quarry

The first map we evaluate is Quarry, this map is 32×32m long and has a maximum height of 10m. We can use some of the terrain interesting features, such as are three bis slopes and the rocky ground on top, to evaluate the model. For instance, w expect the trail on the slopes to be traversable at almost any rotation, expecially from left to right and viceversa. While, the top part should be hard to traverse in almost any case. The following figure shows the traversability probability directly on the map.

some where place the colormap bar



(a) Robot moving from bottom to top

(b) Robot moving from top to bottom

(c) Robot moving from left to right

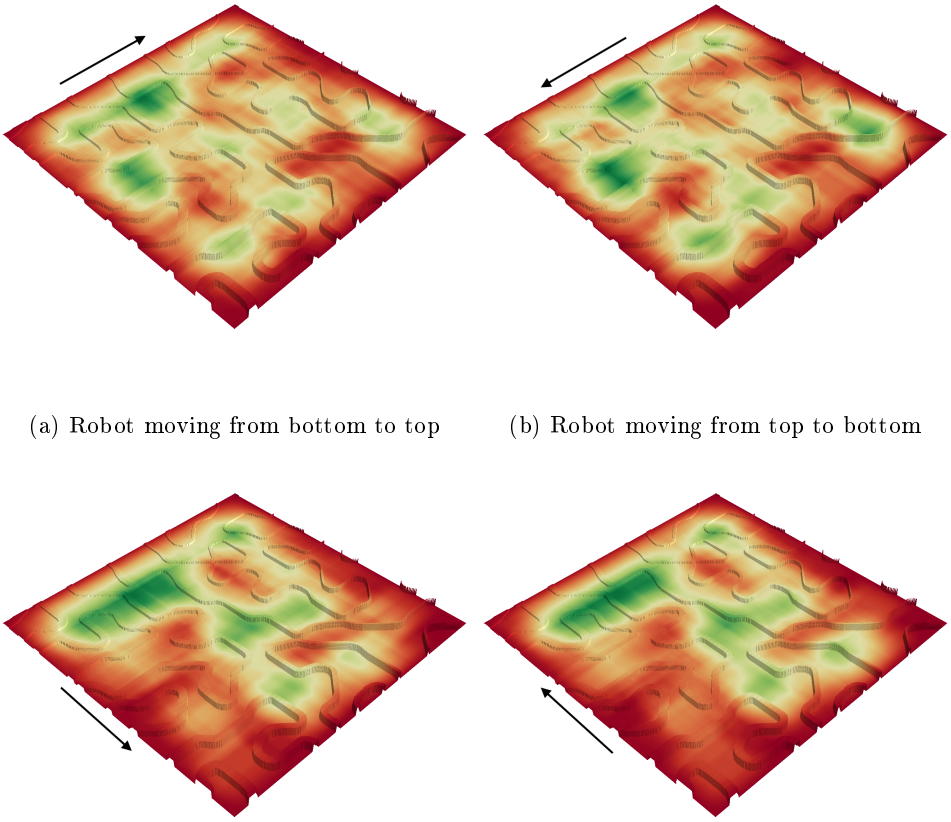
(d) Robot moving from right to left

Figure 1.1. Traversability probability on the Quarry map, 32×32m, for different Krock's rotation. The values are obtained by sliding a window on the map to create the patches and then predict the traversability for each one of them.

Correctly, the lower part of the map, composed by flat regions, was label with high confidence as traversable in all rotations. On the other hand, the traversability of the slopes and the bumps on the top region depends on the robot orientation.

1.5.2 Bars

Bars is a map composed by different heights wall, thus we expect similar probabilities with different orientation.



(a) Robot moving from bottom to top

(b) Robot moving from top to bottom

(c) Robot moving from left to right

(d) Robot moving from right to left

Figure 1.2. Traversability probability on the bars map, 10×10m, for different Krock's rotation. The values are obtained by sliding a window on the map to create the patches and then predict the traversability for each one of them.

This is a hard map for the robot due to the high number of not traversable walls. Interesting, we can identify a tunnel near the bottom center of the maps that shows how the model correctly label those patches depending on the orientation. The following figure highlight this detail.

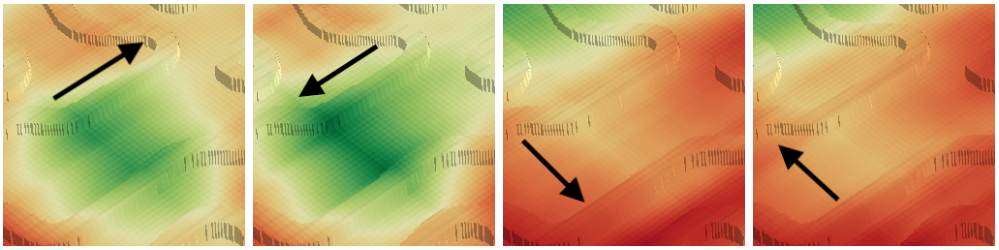


Figure 1.3. Detail of a region in the bars map where there are two walls forming a tunnel. Correctly, when the robot is travel following the trail the region is label as traversable.

1.5.3 Sullen

The following map is a small village downloaded from sensefly online datasets hub. Since the map is huge, we crop a