

Chapter 1

Results

In this section, we evaluate five convolutional neural networks: the model proposed by Chavez-Garcia et al. ? and the four MicroResnet presented in the last chapter. We select the best performing one based on the classification score on the test set. We show the results of the same architecture trained on regression instead of classification and prove the latter yields better results. Finally, we qualitatively evaluate the best model by predicting traversability in real-world terrains.

1.0.1 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 the 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 selected 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 Training setup

Initially, to train the models we first used Standard Gradient Descent with momentum set to 0.95, weight decay to $1e - 4$ and an initial learning rate of $1e - 3$ as proposed in He et al. ?. However, we later switched Leslie Smith's 1 cycle policy ? that allowed us to train the network faster and with higher accuracy. The average training time is ≈ 10 minutes. We minimized the Binary Cross Entropy (BCE) for the classifier and the Mean Square Error (MSE) for the regressor.

1.2 Quantitative results

1.2.1 Model selection

We compared the four different MicroResnet architectures ?? and the Vanilla CNN ?? proposed in the original work ? on the test set using as metric the ROC AUC. Ideally, the second MicroResnet model should be able to extrapolate more features from the patch thanks to the big kernel size. We evaluated those models using a time window of two seconds, a threshold of 20cm and we

applied data augmentation as described in section ???. We run five experiments per architecture, and we selected the best performing network. Table 1.1 shows the results for each architecture. The best performing model was MicroResNet7x7-SE that obtained a ROC AUC of 0.896 on the test set. Moreover, the Squeeze and Excitation operator improved the performance of both MicroResNet3x3 and MicroResNet7x7 but only the latter gained a huge boost of 0.021.

Model	ROC AUC		Params
	Mean	Top	
Original (baseline)	0.892	0.890	974,351
MicroResnet3x3	0.881	0.883	302,610
MicroResnet3x3-SE	0.883	0.888	313,64
MicroResnet7x7	0.867	0.875	303,250
MicroResnet7x7-SE	0.888	0.896	314,28

Table 1.1. Comparison of the four MicroResnet architectures ?? and the original convolutional neural network (Original CNN) proposed in the original work ? on the test. We run five experiments per architecture, we showed the average and the top one for each metric. The best model is MicroResNet7x7-SE.

1.3 Classification results

Table 1.2 table shows in deep the score of the best network, MicroResNet7x7-SE, for each dataset. We obtained 95% accuracy and 0.96 AUC on the validation set and 88% accuracy and 0.89 AUC on the test set.

Dataset		MicroResNet7x7-SE		Size(m)	Resolution(cm/px)
Type	Name	Samples	ACC	AUC	
Synthetic	Training	429312	-	-	10 × 10
	Validation	44032	95.2 %	0.961	10 × 10
	Arc Rocks	37273	85.5 %	0.888	10 × 10
Real evaluation	Quarry	36224	88.2 %	0.896	32 × 32

Table 1.2. Classification results of MicroResNet7x7-SE on different datasets.

1.4 Regression Results

Using MicroResNet7x7-SE as regressor we obtained a MSE of 0.006 on the validation and 0.020 on the test set. Table 1.3 shows in detail the score for each dataset. The regressor can be used to classify patches using different threshold by just see if the predicted advancement is greater or lower than

Dataset			MicroResNet7x7-SE	Size	Resolution(cm/px)
Type	Name	Samples	MSE		
Synthetic	Training	-	-	10 × 10	2
	Validation	44032	0.006	10 × 10	2
	Arc Rocks	37273	0.020	10 × 10	2
Real evaluation	Quarry	36224	0.022	32 × 32	2

Table 1.3. Regression results of MicroResNet7x7-SE on different datasets.

a threshold. However, we always need to fix a threshold to estimate traversability. Once the value is fixed, the same model trained on the same dataset to classify patches drastically outperforms the regressor. Table 1.4 shows the accuracy of two MicroResNet7x7-SE on classification and regression after a threshold of twenty centimeters is selected. For this reason, we decided to evaluate the MicroResNet7x7-SE classifier.

	Regression	Classification
	ACC	
Top	72.8%	88.2%
Mean	73.6%	87.8%

Table 1.4. Regression and classification accuracy for the same model, MicroResNet7x7-SE, once a threshold of 20cm is fixed. The classifier outperforms the regressor by 15%.

1.5 Qualitative results

We evaluated the classifier predictions by plotting the traversability probability on different maps in 3D. We used a sliding window to extract the patches from the tested maps. Then, we created a texture based on the traversable probability. For each map, we evaluated four headings.

1.5.1 Quarry

The first map we evaluate the test set composed by a quarry map 32×32 m and with a maximum height of 10m. We expected the trail on the slopes to be traversable at almost any headings, especially when Krock move from left to right and vice-versa. The top part should be hard to traverse in almost any case. Figure 1.1 shows the traversability probability directly on the map.

some where place the colormap bar

Correctly, the lower part of the map, composed by flat regions, was labeled 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.

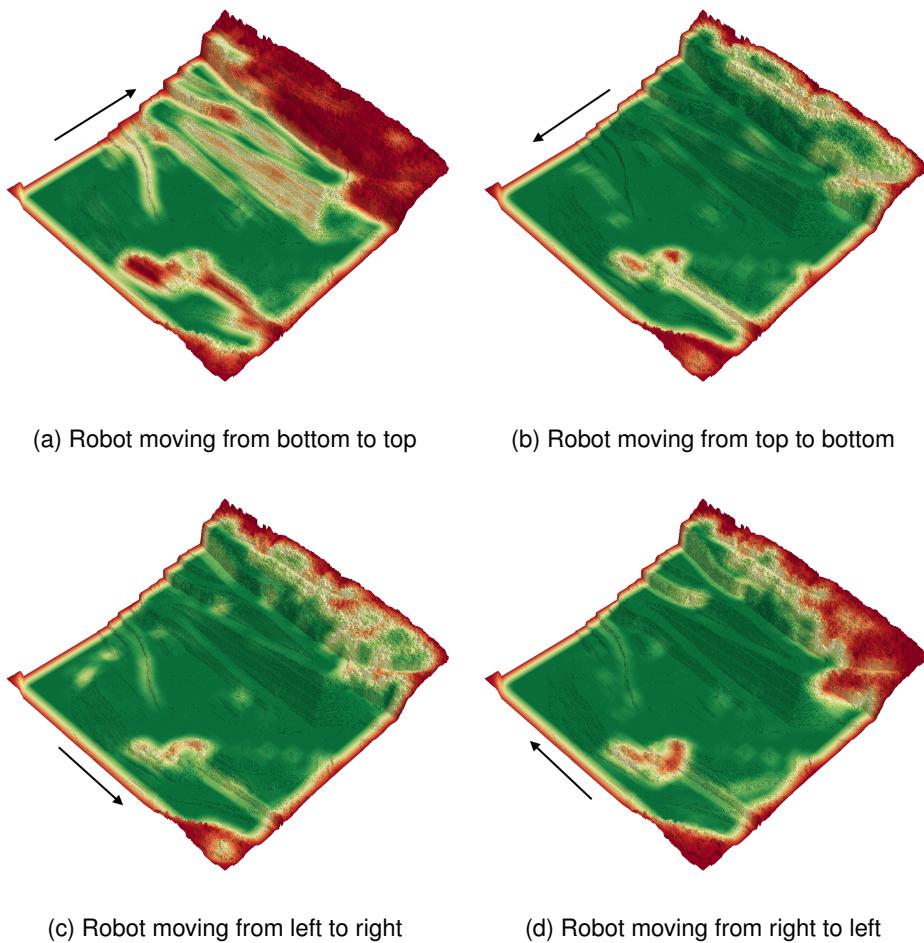


Figure 1.1. Traversability probability on the test set, a quarry 32×32 m and a maximum height of 10m, for different Krock's orientations.

1.5.2 Bars

Bars is a map composed by walls with different heights, thus we expected to be mostly not traversable at different headings. Figure 1.2 shows the traversability probabilities on the terrain.

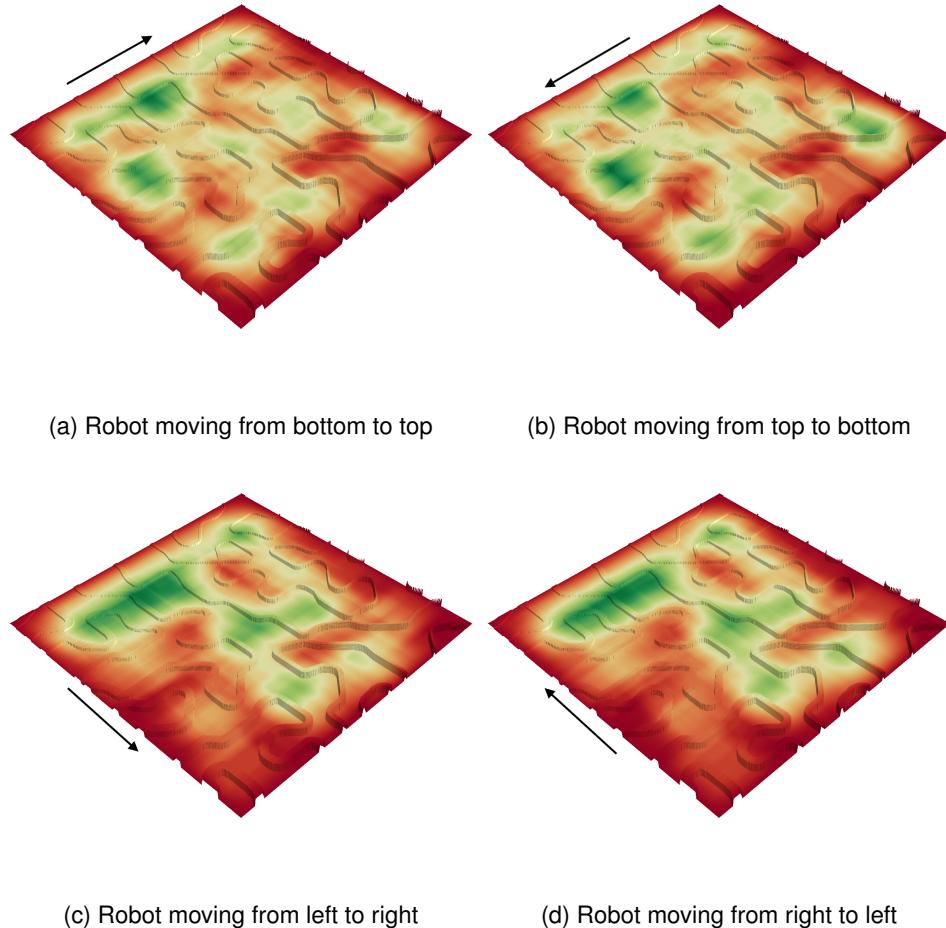


Figure 1.2. Traversability probability on the bars map, a $10 \times 10\text{m}$, for different Krock's orientations.

Definitely, this was 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. Figure 1.3 highlights this detail.

1.5.3 Small village

We applied the same procedure to also a $10 \times 10\text{m}$ small village map. Figure 1.4 describes its traversability for four different rotations.

All the streets were labeled as traversable, while the church's roof traversability depends on the orientation. For example, most steep roofs are not traversable when walking uphill. On the other

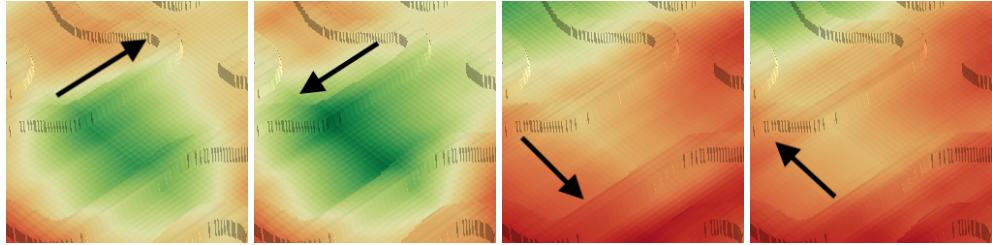


Figure 1.3. Detail of a region in the bars map where there are two walls forming a corridor. Correctly, when the robot is travel following the trail the region is label as traversable.

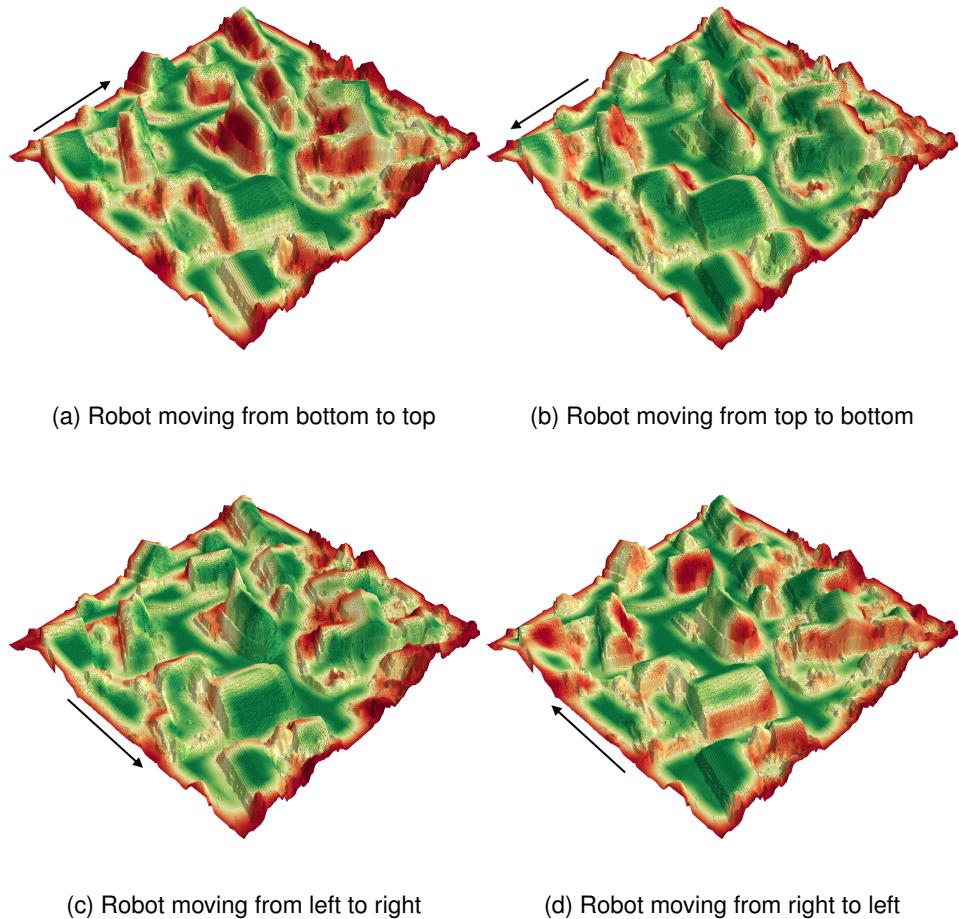


Figure 1.4. Traversability probability on the map of a small village for different Krock's rotation. The surface covers $30 \times 30\text{m}$ and has a maximum height of 10m. The values are obtained by sliding a window on the map to create the patches to predict their traversability.

hand, if krock walks side by side they can be traversed. Figure 1.5 shows this behavior on the church's roof.

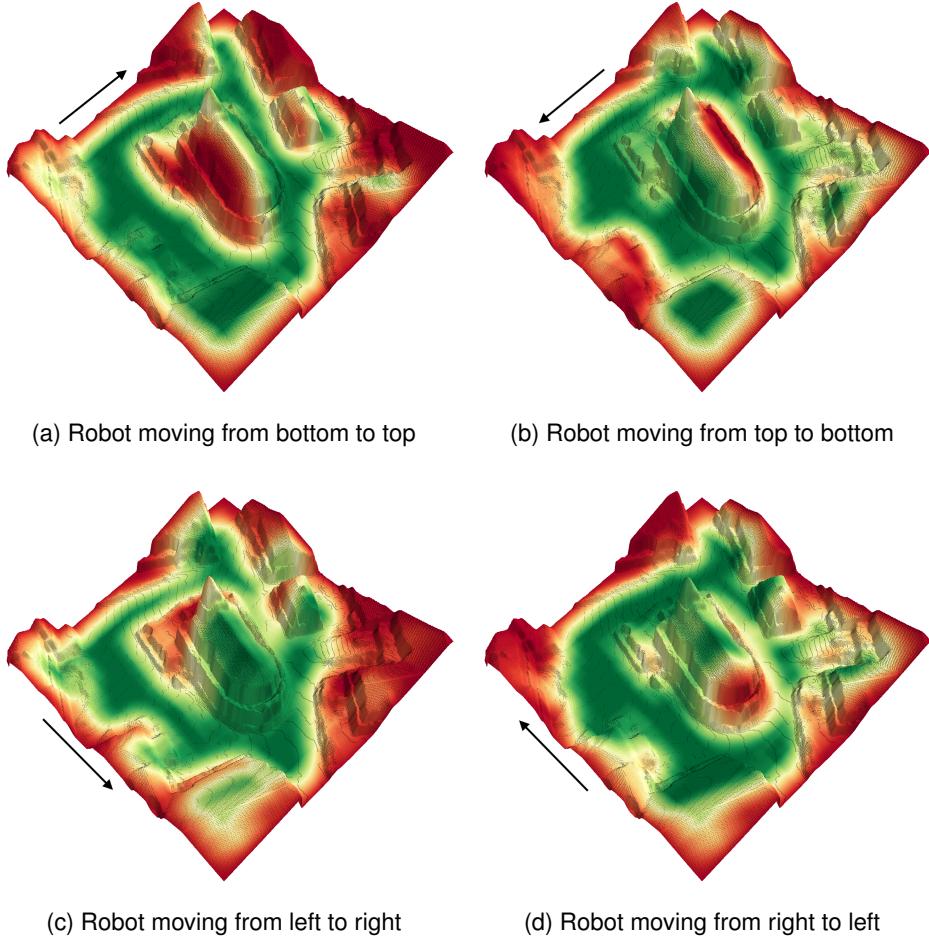


Figure 1.5. Detail of the church in the small village map for different Krock's rotation. All the fours images are correctly labeled accordingly to the robot orientation. In the first two images only downhill part of the roof is traversable. While in the last two the robot is able to travel till the end.