Chapter 1

Interpretability

In this section, we will evaluate the model's prediction to better understand it. We will find if there are any features in the patches that can confuse it and if the model's output is robust. First, we will introduce on technique used to highlight the region of the input image that contribute the most to the model predictions. Then, we will use it on the data from the *Quarry* test set to find out the patches were the model fails and analyze them.

Later, we will work with custom created patches with different features, walls, bumps, etc, to test the robustness of the model by comparing its predictions to the real data gathered from the simulator.

1.1 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) [**gradcam**] is a technique to produce v̈isual explanationsf̈or convolutional neural networks. It highlights the regions of the input image that contribuite the most to the prediciton.

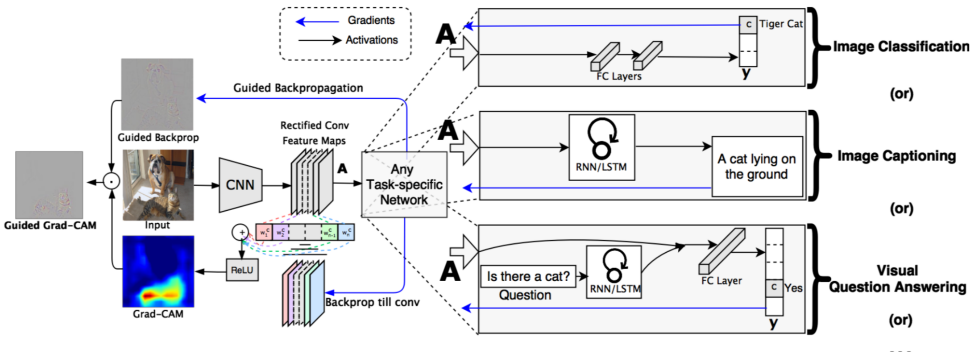


Figure 1.1. Grad-CAM procedure on an input image. Image from the original paper [**gradcam**].

In detail, the output with respect to a target class is backpropagate while storing the

gradient and the output in the last convolution. Then, global average is applyed to the saved gradient keeping the channel dimension in order to get a 1-d tensor, this will represent the importance of each channel in the target convolutional layer. We each element of the convolutional layer outputs is multiplyed with the averaged gradients to create the grad cam. This whole procedure is fast and it is architecture independent.

1.2 Robustness

To test the model's robustness we created custom patches with different features, walls,bumps,ramps, and test the model's predictions against the real robot advancement obtained from the simulator. Hopefully, model's outputs should match the ground truth or show a degree of uncertainty in the edge cases. According to the previous experiments, we used a threshold of 20cm and a time window of two seconds.

1.2.1 Untraversable wall ahead at increasing distance

The most trivial test is to place a not traversable wall in front of *Krock* at an increasing distance from the its head. At a certain distance, we expect the model's predictions to be traversable even if the wall itself is too tall. Why? Because the robot will be able to travel more than the threshold before being stopped by the obstacle.

We created fifty-five different patches by first placing wall 16cm long exactly in front of the robot and then move it by 1cm at the time towards the end. It follows some of the input patches ordered by distance from the robot. We remind to the reader that Krock traverse the patch from left to right.

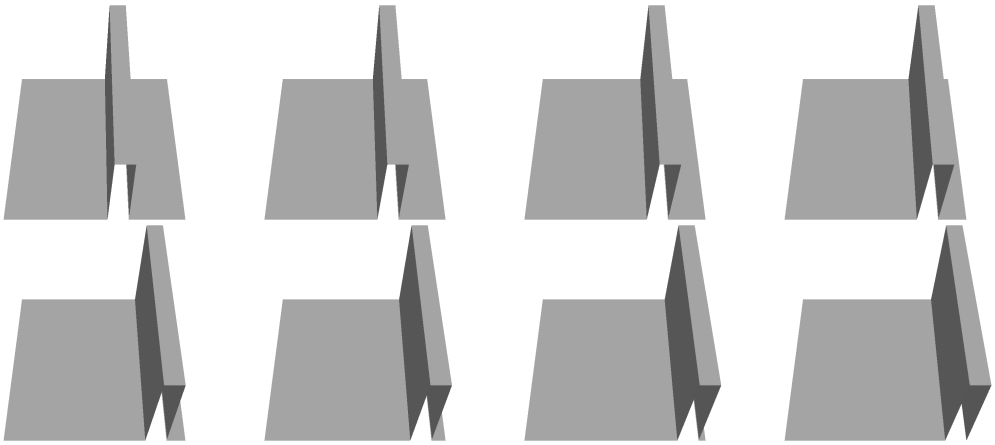


Figure 1.2. Some of the tested patches with a non traversable wall at increasing distance from the robot.

The model's predictions are shown in the following plot. We can see how the two classes invert their values around 20cm. Moreover, the predictions are uniform and do not change multiple times. Intuitively, if a wall is traversable from a certain distance, it will still be if we place even further.

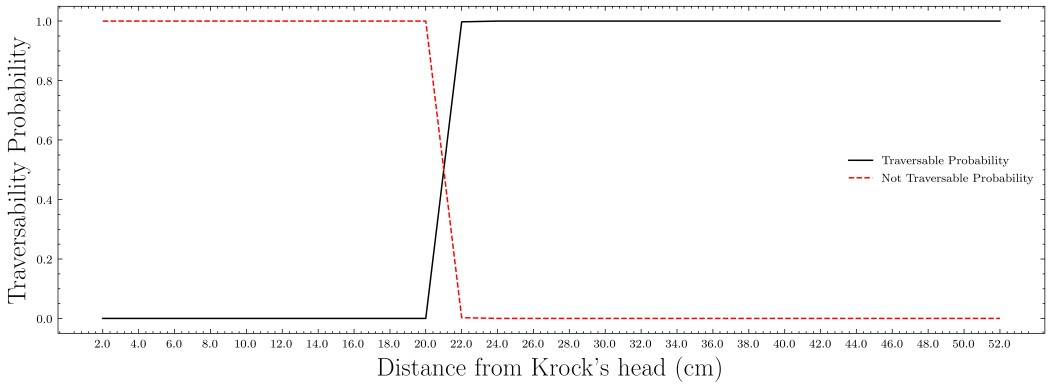


Figure 1.3. Traversability probabilities against wall distance from Krock's head.

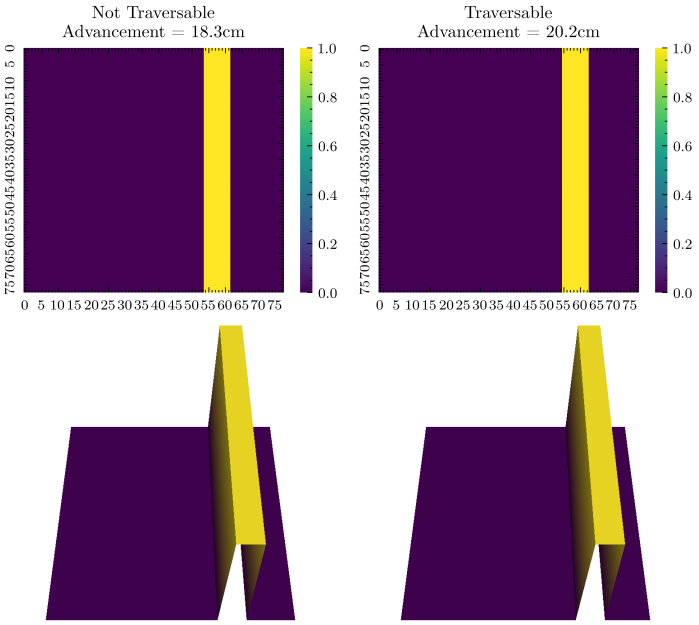
graph too tall, adjust the figure size

Summarized by the following table:

Distance(cm)	Prediction
0 - 20	Not traversable
22 - end	Traversable

Table 1.1. Model prediction from the wall patches.

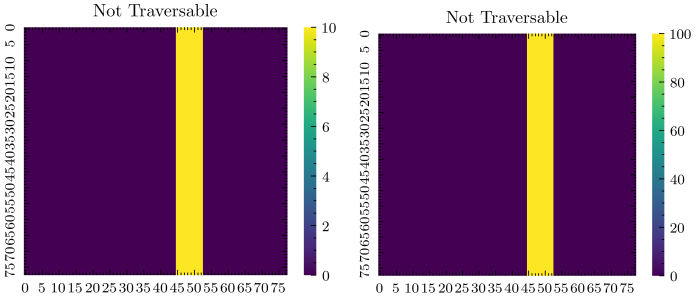
To be sure the results are correct, we run the last not traversable and the first traversable patch on the simulator to get real advancement. In the simulator, Krock advances 18.3cm on the first not traversable patch **??** (a) where the wall is at 20cm from the robot's head. While, on the first traversable patch, with a wall at 22cm, the robot was able to travel for 20.2cm. Correctly, the network's predictions are supported by the ground truth obtained from the simulation. Moreover, the model correctly undestand that the distance from the obstacle is more relevant than its height.



(a) Distance 20cm (b) Distance 22cm

Figure 1.4. Correctly, when the distance between the robot and the wall is greater than the select treshold, in our case 20cm, the patch is label as traversable.

Furthermore, we increased the wall size of the first traversable patch, figure **??**(b), to 10 and to 100m to see if the model will be confused. Correctly, the predictions did no change and those patches were still labeled as traversable



(a) height = 10m (b) height = 100m

Figure 1.5. Two patches with a very tall wall at a distance $> tr$.

WRONG, I must have change the patches somwehre

Correctly, the model classifies the patches as traversable and was not confused by the enourmous height of the two walls.

1.2.2 Increasing height walls ahead

An other test we performed was to place walls in front of the robot with increasing heights to check whether the prediction matches the real data. We run forty patches in the simulator from a wall's height of 1cm to 20cm. The following figure shows some of the inputs.

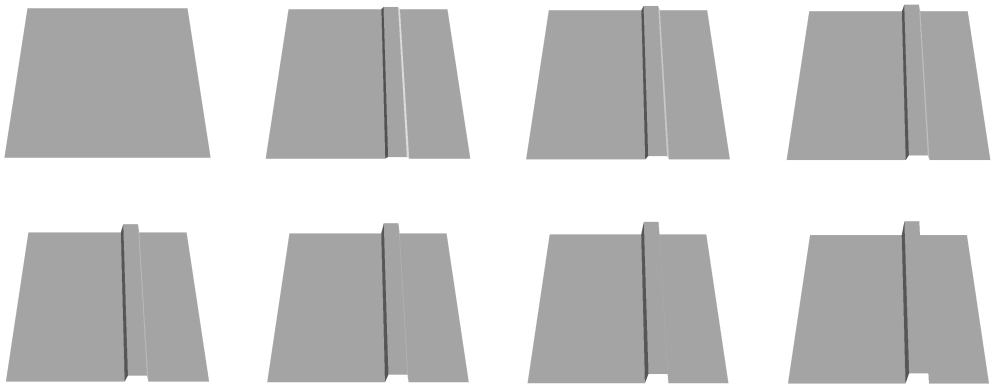


Figure 1.6. Some of the tested patches with a wall at increasing height ahead of Krock.

The models predicted that the walls under 10cm are traversable. We can see that in the edge case, $\approx$ 10cm, the model's prediction change smoothly revealing a degree of uncertanty.

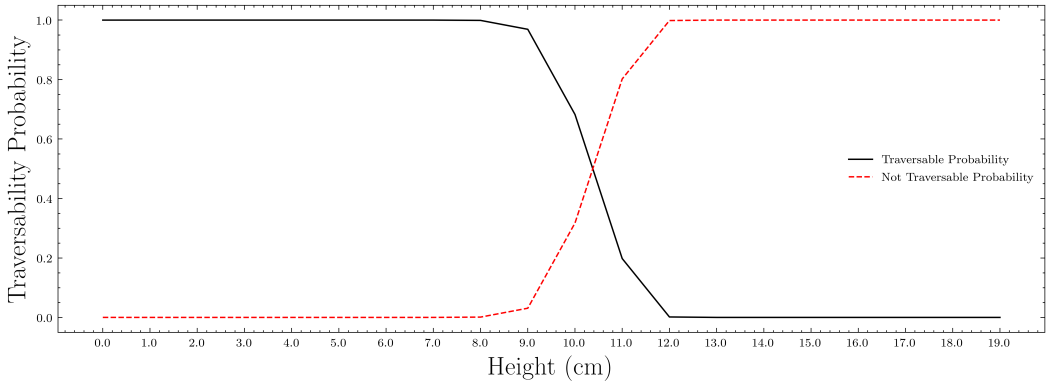


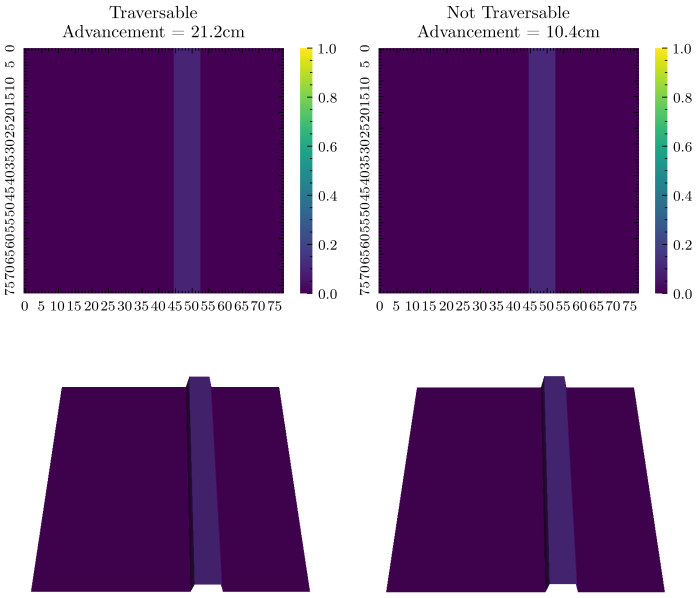
Figure 1.7. Traversability probabilities against walls height in front of Krock.

Height(cm)	Prediction
0 - 9	Traversable
10 - end	Not Traversable

Table 1.2. Model prediction for the wall patches

We can compare the model's prediction with the advancement computed in the simulator

using the same approach from the last section. The following figure shows the results from the last traversable patch and the first non traversable.



(a) height 9cm (b) Height 10cm

Figure 1.8. The last traversable and the first non traversable patches with a increasing height wall ahead of Krock. Correctly the model's prediction matches the advancement from the simulator.

In the first case, the simulator outputed and advancement of 21.2cm meaning that Krock was able to overcome the obstacle, while it failed in the second case. Correctly, the predictions matched the real data.

1.2.3 Increasing height and distance walls ahead.

We can combine the previous experiments and test the model prediction against the ground truth for each height/distance combination. To reduce the number of samples and improve readability, we limit ourself to consider only patches with a wall tall between 5cm and 20cm, we know from previous sections patches with a value smaller and bigger obstacle are traversable and not traversable respectively. Similar, we set the wall's distance from Krock's head between 1cm to 30cm for the same reasons. The following image shows the traversability probability for each patch.

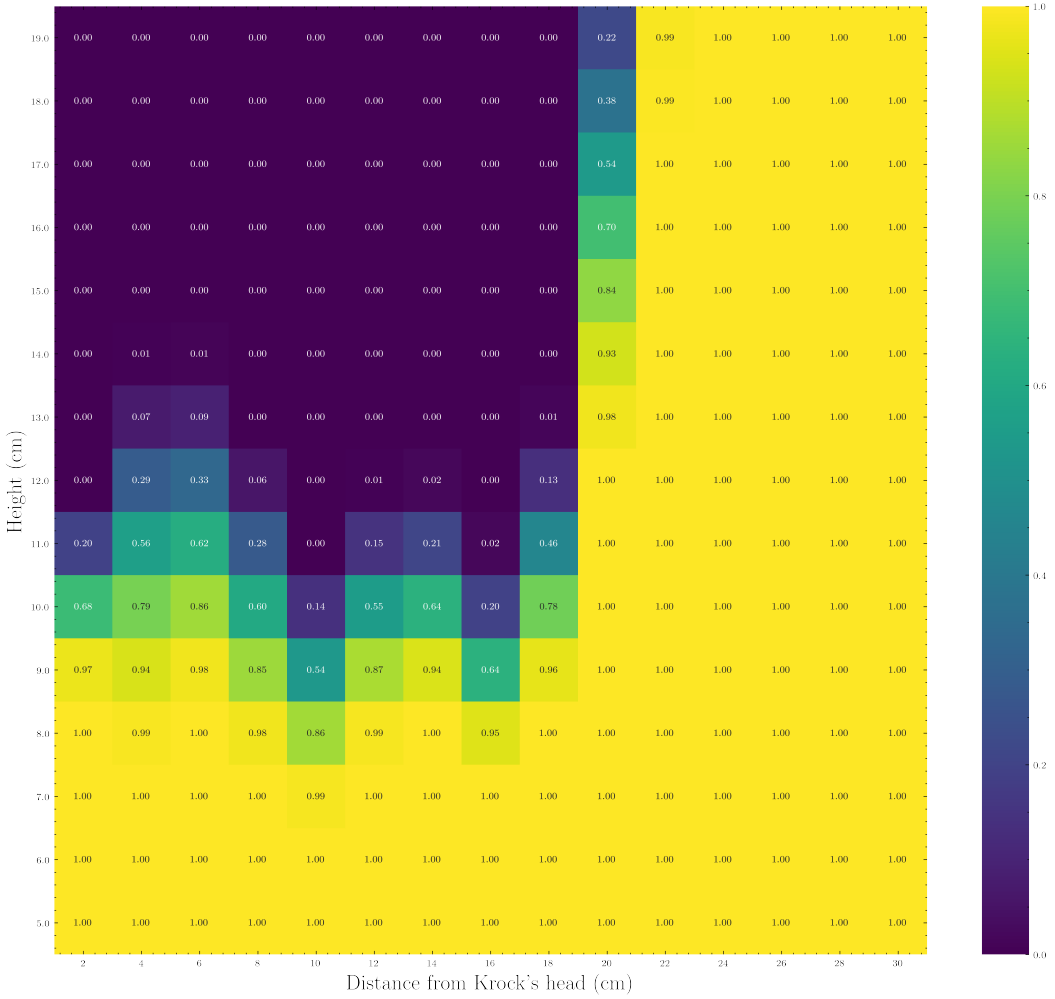


Figure 1.9. Traversability probabilities for patches with a wall of increasing height and distance form Krock's head.

missing ground truth heatmap

FINISH

1.2.4 Tunnel

do it

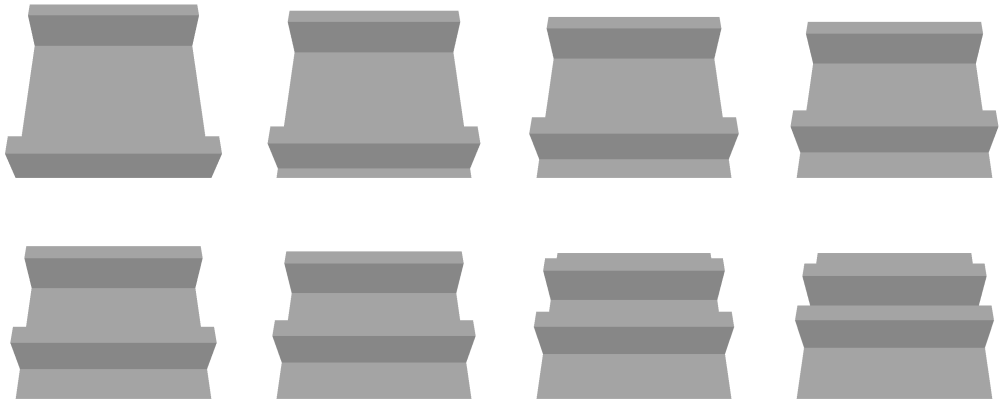


Figure 1.10. Some of the tested patches with tunnel at different distances.

1.2.5 Ramps

explain we had to square the linear ramps to create a small flat region

We generate twenty ramps with a maximum height from 0.25m to 4m. Below we plot the traversability probabilities against the maximum height of each ramp.

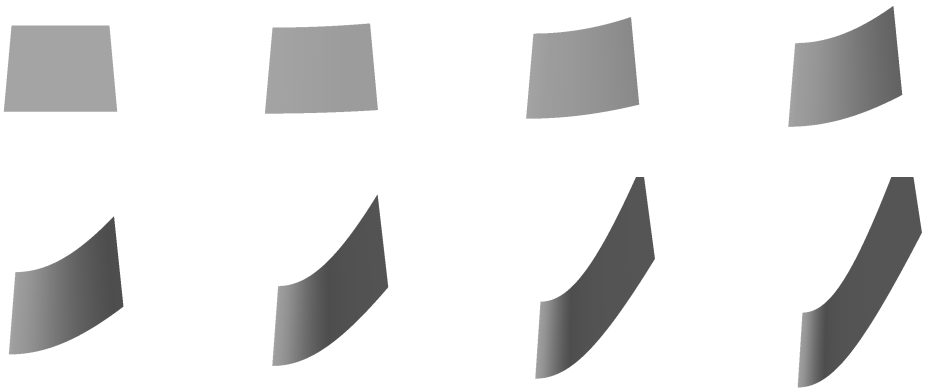


Figure 1.11. Some of the tested patches with steep ramps.

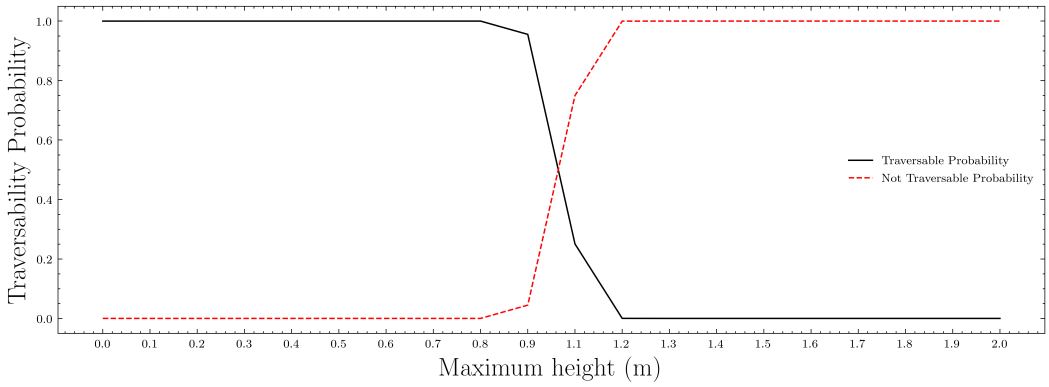


Figure 1.12. Traversability probabilities against maximum height of each ramp.

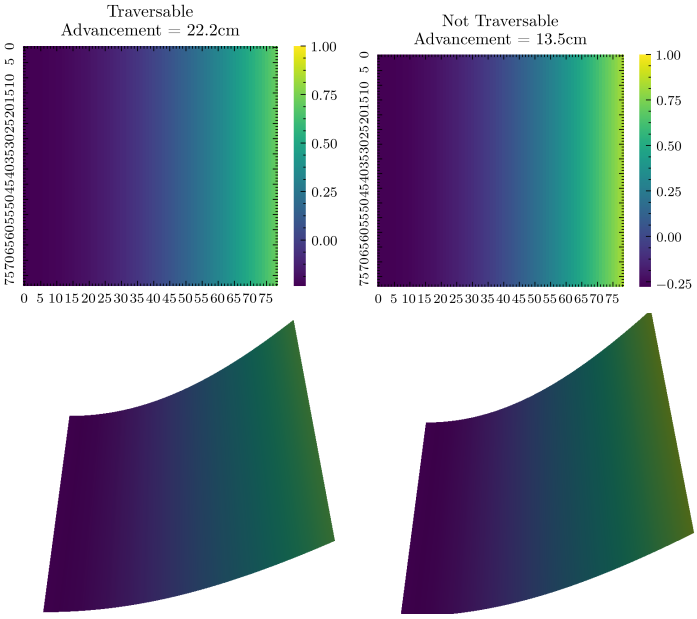
x labels are wrong, why?

The following table summarizes the results.

Height(m)	Prediction
0.5 - 1	Traversable
1 - end	Not traversable

Table 1.3. Model prediction for the ramps patches

We test the last traversable patch and the first not traversable with the real advancement gather from the simulator.



(a) Height 0.94cm (b) height 1.1m

Figure 1.13. The last traversable and the first non traversable patches with a steep ramp ahead of Krock.

scale is wrong

Krock is able to traverse up to $1m$ height ramps, this is confirmed using the simulation. We can add rocks to those patches to give Krock the ability to climb them better.

add rocks

1.2.6 Holes

do it