Table 6: Examples of detections with default settings but varying numbers of variations on samples, including rotations, scales, and random displacements.

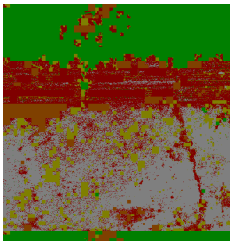
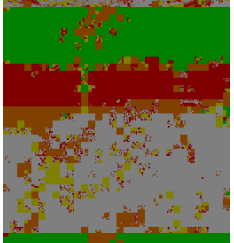
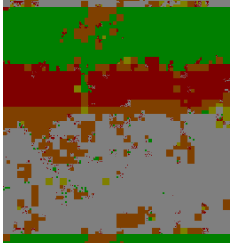
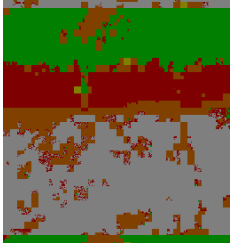
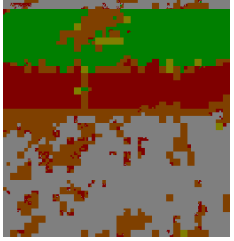
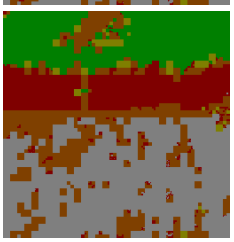
Image	Variations	Avg. time	Qs	Observations
	0	4013.8 s	205885	Very few samples resulted in a lack of confidence in the predictions of the KNN classifier, leading to many quadrants being forced into subdivision and exhibiting numerous noise in the detections, which was more pronounced in the smaller quadrants.
	50	523.5 s	26133	A fast start recognizing with good prediction probabilities but with some flaws. There are still noises in the detections, although more tolerable.
	100	353.7 s	18245	Fast recognition with good prediction probabilities resulting in fewer quadrants, although the loading time of samples started to become lengthy. There are still noises, although more noticeable only at the transitions between contexts.
	200	352.5 s	17677	More accurate quadrant recognitions, although the prediction begins to introduce loss of details and/or detect a quadrant that has two separate contexts as a third unrelated context, as observed in the transition from bricks to concrete being detected as dirt. Although the processing time is nearly identical to the previous test, the 200 samples took approximately 20 seconds to load, resulting in a total time period longer than the previous one.
	500	88.8 s	4313	Approximately 50 seconds to load the samples. Although the process was extremely fast, there were context detection failures in areas where they had not occurred previously. The issue of detail loss becomes more noticeable, and some erroneous detections persist in regions that had been presented before. The refinement down to the minimum quadrant size was not extensively explored due to the improved prediction probabilities at smaller sizes, resulting in higher prediction probabilities before reaching the minimum size.
	1000	98.2 s	4273	The samples loaded after nearly 2 minutes. Detail loss became more pronounced, with the quadrants having a higher prediction probability, not benefiting from many minimum-sized quadrants. Furthermore, the result remained the same or worse than the previous test, as a large number of samples were added for KNN classifier refinement, resulting in a loss of performance.

Table 7: Example of detections with default settings, but varying the probability threshold for subdividing quadrants.

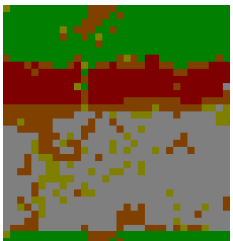
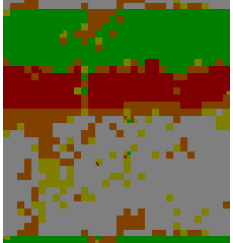
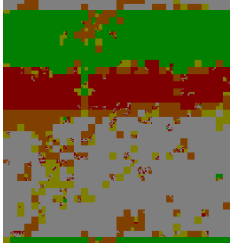
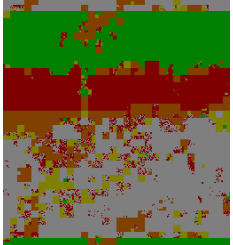
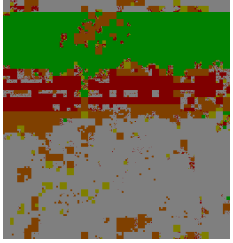
Image	Threshold	Avg. time	Qs	Observations
	10%	20.3 s	1365	No quadrant was subdivided beyond the maximum size. Extremely fast result, however, akin to a low-resolution view of the detection.
	50%	22.3 s	1405	A few quadrants were subdivided beyond the maximum size, yielding a result similar to the previous one.
	70%	128.1 s	6837	Result with more accurate detections, correcting some erroneous detections compared to the previous result. Few quadrants reached the minimum size.
	90%	522.1 s	27033	Standard result, with several quadrants reaching the minimum size, but still benefiting from a significant portion of the detection in both minimum and maximum-sized quadrants, contributing to detection performance.
	100%	1016.1 s	52337	Absolute probability. Multiple quadrants were subdivided, continuing to subdivide until an absolute result or the minimum size was reached. The image is extremely detailed, with minimal noise, but it also includes numerous erroneous detections, leading to an extremely time-consuming process.

Table 8: Example of detections with default settings but varying the number of threads. Since such a change does not significantly affect the final result, only the average time of image processing with the specific number of threads will be recorded, in an average of 31900 quadrants.

Threads	Avg. time	Observations
1	966.1 s	A single thread performing detections for each quadrant. The application did not experience any slowdown, although executing the task in this manner resulted in an extremely time-consuming final result.
2	606.8 s	With two threads, the application still did not experience any slowdown; however, surprisingly, the execution time was not halved, indicating a possible use of CPU by the application itself that affects the final result.
4	580.0 s	There was no significant change in the use of 4 threads, although there is still a slight improvement in time. The application starts to show a tiny slowdown in the range of milliseconds between detections.
8	507.3 s	Approximately 80% of the image was detected in one minute. The application's slowness becomes more evident, while the total processing time does not show significant improvements.
16	640.4 s	The application's slowness becomes more evident than before, likely due to the number of simultaneous checks being performed by the KNN classifier. As a result, the execution time starts to increase again.
32	690.2 s	The application freezes for half a second during the simultaneous detection of 32 quadrants, causing an increase in processing time. At this point, other detection approaches could be explored in order to reduce processing time.
512	740.0 s	The application experiences significant freezes, some lasting more than 5 seconds, during the processing of 512 quadrants simultaneously. Surprisingly, there were no great changes in processing time.