

# **Digital billboard damage detection using computer vision**

Niek Boersen

University of Twente and Hecla Professional

Supervisor: Dr. Ir. Cora Salm

Supervisor Hecla: Ing. Leo Kuipers

Critical Observer: Dr. M. Poel

20-06-2019

Creative technology

Graduation project 2019

## Abstract

Digital billboards for marketing purposes are nowadays commons sight along the Dutch highways. A company that sells, constructs and does the maintenance of these outdoor LED screens is Hecla Professional. Like all electrical devices, these billboards are susceptible to breakage. This paper focusses on the development of a system that detects broken LED tiles in billboards by artificially checking video streams. The system also informs the technical staff when it detects a malfunctioning LED tile. The prototyped system does this by using state-of-the-art deep-learning object detection methods. In this study, an accuracy of 83.79% was achieved on an experimental setup. These experimental results from an indoor setup demonstrate that the proposed prototype can, in fact, detect the defective LED tiles and that the system can be used for this job.

# Table of contents

1	Introduction .....	5
2	Literature review .....	6
2.1	Frameworks exploration.....	6
2.2	Damage specification .....	7
2.3	Object detection and training models .....	7
2.4	Data collection.....	8
2.5	Conclusion.....	9
3	Related work.....	10
3.1	Smoking scene detection .....	10
3.2	Crime scene prediction .....	10
3.3	Road damage detection by smartphone images.....	11
3.4	Human related-health actions detection .....	11
3.5	Conclusion.....	11
4	Methodology .....	12
4.1	Billboard research.....	12
4.2	Data synthesis and data collection.....	12
4.3	Defective tile detection.....	13
4.4	Thresholds and notifying.....	13
5	Billboards internal workings .....	14
5.1	Outdoor billboard overview .....	14
5.2	LED Panels.....	15
5.3	LED tiles and failure modus.....	16
6	The indoor setup.....	18
6.1	Setup components.....	18
6.2	Images synthesis.....	19
6.3	Data collection.....	20
6.3.1	Camera angle, camera distance, and lightning .....	20
7	Model training .....	22
7.1	Data pre-processing.....	22
7.2	Deep-learning framework.....	23
7.2.1	Faster R-CNN framework .....	24
7.3	Training .....	26
8	Evaluation model.....	28
8.1	Method.....	28
8.2	Results .....	29
8.2.1	Overall performance.....	29
8.2.2	Performance per condition .....	32

8.2.3	Speed .....	35
8.3	Analysis and conclusion .....	35
9	The output of the system and the database .....	37
9.1	The database .....	37
9.2	Output.....	38
9.2.1	Types of notifications .....	38
9.2.2	Conclusion and result .....	38
10	Conclusion.....	40
11	Recommendations .....	41
11.1	Training images .....	41
11.2	Database .....	41
11.3	Cameras .....	42
11.4	Boundaries and exceptions .....	42
11.5	Planned timeslot detections .....	42
11.6	Different feature extractor with higher accuracy.....	43
11.7	Multiple detections and anchor scaling .....	43
12	Ethical considerations.....	44
12.1	Ethical risks .....	44
12.2	Similar case .....	44
12.3	Ethical benefits .....	45
12.4	Misuse of the product .....	45
12.5	Iterating the process.....	46
13	References .....	47
14	Appendices .....	50
14.1	Appendix A .....	50
14.2	Appendix B.....	52
14.3	Appendix C.....	54
14.4	Appendix D .....	57

## List of figures

Figure 1 Overview of parts from a billboard.....	14
Figure 2 Example footage from the camera with panel size indicator .....	15
Figure 3 Different types of problems in LED tiles (IM) .....	17
Figure 4 Example footage of billboard with broken LED tile.....	19
Figure 5 Synthetic data generation process .....	19
Figure 6 Schematically representation of the IDS .....	21
Figure 7 Data annotating process .....	23
Figure 8 Example neural network setup .....	24
Figure 9 Overview of the RPN algorithm[20].....	25
Figure 10 Overview of the total Faster R-CNN system[20].....	25
Figure 11 Graph of the Total loss.....	27
Figure 12 ROC curve from the synthetic generated data .....	32
Figure 13 Schematic overview detection data handling .....	37
Figure 14 Example output of the system.....	39

## List of tables

Table 1 Overview different model frameworks .....	8
Table 2 Component list of LED panel.....	16
Table 3 Meaning of loss functions .....	26
Table 4 Confusion matrix with $p=0.5$ .....	29
Table 5 Percentiles of the dataset .....	30
Table 6 New confusion matrix with $p=0.6029$ .....	31
Table 7 Mean certainty of images under different conditions .....	32
Table 8 Bootstrap for Independent Samples Test on the difference of means – Light levels .....	33
Table 9 Bootstrap for Independent Samples Test on the difference of means – Camera height.....	33
Table 10 Bootstrap for Independent Samples Test on the difference of means – Camera distance.....	34
Table 11 Test of Homogeneity of Variances on means of the different perspectives .....	34
Table 12 ANOVA test on the means of the different perspectives .....	34
Table 13 Post-test on the means of different perspectives .....	34
Table 14 Significance overview of conditions .....	36

# 1 Introduction

In a world where time is money, it is important for companies to keep their services up and running to keep their customers satisfied. When the services of a company are down, all sorts of problems can occur. The sales might decrease, customers can get annoyed or the brand's reputation can be damaged. All this negativity is undesired and has to be prevented at all times. A business sector where time is literally money is the Audio and Visual sector. Downtime equals direct loss in income due to the fact that the customer does not pay when the Audio and Visuals are not working from the second they go off. For this reason, it is important to minimize the off-time for these products.

An innovative company in this field is Hecla Professional. This company takes care of the whole Audio and Visual package entailing design, installation and maintenance. They are selling this high fidelity package, next to a lot of other solutions, for large digital billboards that are placed along the Dutch highways. These digital billboards reach a lot of people due to their size and height. The problem with these digital billboards is that they, like many electrical products, can break. In the current situation, Hecla Professional is manually monitoring these billboards by cameras to see if they are functioning. When they notice that a digital billboard is not working properly, they send a mechanic. According to Hecla, the problem in the current situation is that it is a really time-consuming activity to monitor all these cameras and there is a chance that malfunctioning billboards will not be seen. Until now there has not been a more efficient solution to this time-consuming job. The graduation project research will be done in this field to optimize this part of the company's current workflow. This is important to provide an even higher uptime of their services. Next to this, innovating in this field will make sure that Hecla stays ahead of its competitors. A solution will also lead to higher profits because of the minimized down-time.

In order to improve the current workflow, a computer vision based system is envisioned. The already placed cameras will be monitored by a system that uses artificial intelligence to check for malfunctioning parts of the billboards. During this research, this specific system will be designed, prototyped and evaluated. At the end, there will be a recommendation section to find out if the system has potential for further development. Before beginning with the system, the background will be investigated to learn from earlier conducted researches.

## 2 Literature review

In order to find this solution in the world of computer vision, a lot can be learned from the literature, starting off with this literature review. The aim of the literature review is to provide insights into computer vision damage detection using deep convolutional neural networks (CNN). These CNNs are the state-of-the-art performers in object detection, which is in the Hecla's case the damage to digital billboards, detection[1]. To find a solution to the problem that is occurring in the current situation, it is necessary to investigate the currently used object-detection manners specifically for damage detection. For decent investigation, four different aspects will be covered. First, the different frameworks of object-detection that are available. This is to make the right choice in software that fits the needed solution. Second, damage specification. To train the object-detection models, it is necessary to know the symptoms of broken screens. Thirdly, different types and setups of CNNs are used to train models for different kinds of applications. This needs to be investigated for this specific case. And lastly, how will the required data be collected which is needed to train an accurate object-detection system. If these aspects are investigated and concluded, there will be a state-of-the-art review to show what has been done in this area already.

If every above-described aspects are researched, a conclusion will be drawn to conduct the rest of the project in well-deliberated fashion.

### 2.1 Frameworks exploration

Starting off, it is crucial to explore the different types of deep learning frameworks. These frameworks help in setting up an artificial agent that acts upon certain inputs. The different open-source frameworks all have certain positive and negative attributes and they suit different types of applications. This paper focuses on the frameworks that are contrived for computer vision and more specifically the ones capable of object-detection. Widely used frameworks are Caffe, Theano, Torch, TensorFlow and NEON. These are the most used frameworks based on the members in Google groups and GitHub contributors [2]. There are two different evaluation methods for these frameworks. On one side you have the benchmarking, mainly consisting out of running time, memory consumption and CPU- and GPU utilization[3]. On the other side, there are the evaluation methods base on usability, straightforwardness, and flexibility. These latter ones focus on the accessibility over several devices and the ease of implementation. This is case specific due to the technical backgrounds of the researchers and the systems requirements. When taking a look at the comparative study of Bahrampour et al[2], it is clear that Torch and Neon are the best frameworks regarding speed during training, where TensorFlow and Theano are the best with extensibility and flexibility.

TensorFlow is the final framework that is going to be the best in this situation. It is one of the slower frameworks during the training with CPUs [2], [3] but when using GPU it will have enough time to train, in the magnitude of milliseconds per image[2], within the scope of this project [4]. Additionally, it is a very flexible framework [2] which has high performance and is one of the better

ones in object-detection[5], the functionality that is needed. Another prominent aspect of TensorFlow over Theano is that the framework can be implemented in the Python environment[2]. This is very useful for the reason that the author has knowledge about this programming language. This will speed up the prototyping of the product. In the end, this results in an overall faster development of the envisioned solution. For all these reasons, TensorFlow is going to be the appropriate framework above its competitors.

## 2.2 Damage specification

To create the envisioned system that recognizes broken LED digital billboards from a live feed video stream, it is necessary to specify 'broken' to the system. The system needs to understand what a functioning LED screen is and how it behaves. It is also important to learn the system the specifications of a broken LED screen in order to differentiate between the two. Before specifying the differences between the functioning and malfunctioning LED panels, a better look at the blueprints of the LED screens is required. The screens are built up with multiple panels with a surface of 0.8847 square meters. Then, these panels consist of smaller square tiles with the actual LEDs placed on them. Every LED panel has three power supplies to power all the LED tiles on the panel. One power supply is dedicated to the red LEDs on the tiles. The other two are together powering the blue and the green LEDs on the tiles[6]. Next to the part on how the billboards work, there is the part of how the specific LED screens at Hecla break and how this breakage looks. The researcher H. Bierma did his thesis on analysing data of the malfunctioning screens at Hecla. He concluded that there are three main initiators of malfunctioning billboards. The first one is a defect tile. In such a case, a square LED tile completely turns off. The second one is a power supply failure. If that happens, a certain colour of the billboard will not work properly anymore. The last group was undetermined. This meant that there was no direct cause found that lead to the breakage[6], resulting in all sorts of malfunctioning billboards. Therefore this group will not be discussed any further. For the solution, the focus must be on detecting defect tile and the power supply failure because these are the two biggest groups and presumably easy to detect by computer vision. The looks of these damage types are now specified and can be learned to the system. The innerworkings of the billboard will be worked out further in chapter 5.

## 2.3 Object detection and training models

In the previous paragraph, the looks of the damage were specified and now the system can be learned to detect this. To detect objects, which is in this case the damage, using computer vision from a live feed video, a detection model is required. A framework containing a pre-trained model can be downloaded, which classifies a lot of standard objects, or a new model can be trained with a dataset to recognize specific objects like the billboards. The following citation explains why a pre-trained model should be used:



“There are many models we can use that have been trained to recognize a wide variety of objects in images. We can use the checkpoints from these trained models and then apply them to our custom object detection task. This works because, to a machine, the task of identifying the pixels in an image that contain basic objects like tables, chairs, or cats isn’t so different from identifying the pixels in an image that contain specific pet breeds.”[7]

Right now there are dozens of these pre-models available online to choose from. They all have different aspects and are made for a certain application. In the situation where this system is going to work, the most important metrics are speed and accuracy. Speed is measured in milliseconds per image to detect and accuracy is measured in mAP, which stands for mean average precision, on the standardized COCO dataset. The three state-of-the-art model frameworks at this moment are Faster R-CNN, SSD, and R-FCN[8]. At this moment Faster R-CNN and R-FCN are the higher performance models and are the better choice if comparing the speed and high mAP. Faster R-CNN has an even higher mAP than R-FCN which is desired. If speed is important, SSD is the better choice[8]. The best choice in case of the envisioned system would be Faster R-CNN due to the higher mAP. Additionally, you have three different versions of R-RCNN namely: R-CNN, Fast R-CNN, and Faster R-CNN. The latter is the newest and the fastest (at least 213 times faster than R-RCNN)[9],[10] so that will be the model framework to go. An overview of the positive sides and downsides of the different models will be given in table 1.

*Table 1 Overview different model frameworks*

<b>Model framework</b>	<b>Advantages and disadvantages</b>
R-FCN	- slowest - accurate
Faster R-CNN	+ fast + very accurate
SSD	+ fastest - worst accuracy

## 2.4 Data collection

Before the training of this Faster R-CNN model can begin, lots of data is required. Hundreds[10] to thousands[11] of images are needed before the model reaches the desired accuracy. Getting this data in a short amount of time can be difficult. To get the amount of data in time, there are several techniques to enlarge the data set. These techniques are cropping, flipping and altering the image’s saturation and shading[11]. Using these techniques to create synthetic data gives the system more input to train which will result in higher accuracy.

This data needs to be separated in at least two groups: training images and testing images. This

is necessary to get information about the model's performance in terms of efficiency and accuracy. Sometimes there is an extra group called validation data to test the model with novel, unseen data to give even better insights about the performance. The amount of data and the division over these three groups is really situation dependent and the literature does not give a unambiguous answer. This article [12] used a division of 80% training data against 20% validation data whereas this article [11] a division of 84% for training against 16% for testing. The recommendation would be, if there is enough time, to try different settings and check the results.

## 2.5 Conclusion

To conclude, this section researched several fields in the object detection field gaining imperative knowledge about the background of the graduation project. Starting off with the different types of frameworks currently used. Comparing the framework's benchmarks and their usability the conclusion came down to make use of Google's TensorFlow framework in C++. Then the LED screens themselves. The weak spots of the LED screens were uncovered and the symptoms of the malfunctioning systems were discussed. These specifications can now be taught to the Faster R-CNN framework. That was the best fitting model for the system because it had some competitive advantages above the others.

The sources that were used came close to the situation at Hecla and are probably trustworthy. On one topic, there were not enough sources. This was the damage specification section. Only one source was found about this topic and further research is needed to reveal the details.

This paper gave insights about the setup strategies for a deep learning convolution network for the current situation at Hecla. The existing literature revealed the best way to set up the experiment in mind. Making use of Google's TensorFlow API running on a Faster R-CNN with as many images as presumably the way to go. For further research, it is important to find out more about the Faster R-CNN framework. When all of this has been done, an experimental setup of the system is recommended to test the findings of this literature review more elaborately.

### 3 Related work

In order to establish an even better context around the subject, a state-of-the-art review is required. Trying to fill in the gaps in the knowledge that stayed open after the literature review. Next to that, to see what the possibilities are with the envisioned system. In this part, four example cases are given products or services that are currently the farthest developed in the field that is being investigated, the object detection field.

#### 3.1 Smoking scene detection

The first case is about the problem in India that smoking sometimes is promoted in films and video clips. The number of scenes in these clips and films where people are smoking is uncountable and are a type of surreptitious advertising. The system that this research team created tries to automatically detect scenes where people are smoking and then displaying a warning message if cigarettes are visible. This in order to warn the viewers about the downsides of smoking and hopefully prevent the viewers to start with smoking[10]. The researchers used the TensorFlow API from Google, which is the same as the one that is going to be used for the graduation project. With a dataset of only 600 images, they reached an accuracy of 94.08%. This case is similar to the graduation project due to the fact that they are both detecting small objects from video. The accuracy of this research is promising for the results that are expected in the graduation project.

#### 3.2 Crime scene prediction

The second case is about the prediction of crime scenes from CCTV cameras. Nowadays there are lots of cameras installed everywhere to monitor certain areas. All these cameras are monitored manually and it would be easier if artificial intelligence could do that. The researchers of this case created a system that predicts crime scenes based on the detection of knives and firearms[13]. The researchers stated that they achieved an accuracy of 90.2% but this is questionable. They trained their data set with stock images and not with real-time CCTV cameras. To conclude, they tried to detect small objects from a scene which is similar to the goals of the graduation project.

### 3.3 Road damage detection by smartphone images

The next case is in the damage detection field, really close to what the graduation project is about. The system that these researcher made tries to recognize road damage from smartphones. This in order to make the road maintenance more efficient by letting the roadworkers know the damaged roads are. The cameras from the smartphones are being used to retrieve images from. These images are then analysed and classified as damaged or not damaged. After training with these datasets, the accuracy of the system reached 0.62%. The certainty increases when multiple detections of damage are close to each other[14]. To conclude, the system works reasonably but multiple detections should be combined to increase the certainty. This is useful knowledge because it can increase the certainty in the case of the graduation project as well.

### 3.4 Human related-health actions detection

The last case is about detecting emergency actions to help monitor unattended children, individuals with special needs or the elderly. This all by making use of the TensorFlow object detection API. Making use of the Android camera placed on the Samsung Galaxy S6, they try to detect cases as falling, headache, nausea, and sneezing. They reached a total accuracy of 93.8%[15] which is really high. Something that is questionable about this research is that they trained on static images where the object is precisely doing what the researchers expect. This might result in lower accuracy in real life situations. To conclude, this article shows that it is possible to detect certain behavior of people based on cameras from cell phones. This is a little bit further from what the graduation project is about but it shows the possible accuracy of the API that is going to be used.

### 3.5 Conclusion

By reviewing these products on certain aspects multiple insights came to mind. The overarching conclusion is that the systems are all capable of having really high accuracy in detecting the objects they are trained for. This is promising for the graduation project because a higher accuracy is of course desired. Next to the accuracy, there was the insight from the road damage detection project. They concluded that using multiple detections with low certainty in the same area of the image can upscale the total certainty of a certain detection. This can be really useful for detecting the damage in billboards too. Another great insight from the first three articles is that the systems are able to detect small objects too even if the size of the training data set is not enormous. This gives confidence that objecting small damages to billboards will presumably be detectable as well. Together with the literature, a lot was learned from the state-of-the-art. In the next section, the methodology of the research will be worked out.

## 4 Methodology

During the literature review, the needed background knowledge was gathered. This readily available knowledge is now at disposal and saves work during the development of the prototype. Then, the state of the art research exposed the possibilities of the currently newest technologies. Knowledge about these useful applications in roughly the same domain preserve us from making mistakes and developing impossible systems. The next phase will describe different types of researching methods to in the end develop the envisioned system. In this section, these research- and evaluation methods will be worked out in depth before conducting them.

### 4.1 Billboard research

First, a clear overview of the internal workings of the digital billboards needs to be established. The literature exposed the type of defect that is going to be tackled during the research, namely the defective tile. To find out more about this specific defect it is a requirement to dive deeper into the inner workings of the billboards that Hecla uses. The first step will be an open interview with one of the technicians will be used. These people are working every day with the billboards and have knowledge about the inner workings of the billboards. This qualitative data will be processed in chapter 5. Besides talking to the employees, the datasheet of the billboard will be studied in depth. All the technical details will be in this sheet and therefore valuable to the research.

### 4.2 Data synthesis and data collection

In order to train a model that will be used inside the system, a lot of data from broken billboards is required. This data will not be readily available in the given time because these LEDs tiles do not break that often. For this reason, this data will be artificially generated by written software. From the literature review, it is known how defective tiles appear on the video stream, namely as a black square. These black square will be mimicked by editing in a black square into copyright-free standard photo- and video footage. This footage will be displayed on an indoor LED screen and the result will be a representation of a billboard with a broken tile. Just like in the real world situation, a camera will be placed at a certain distance away from the screen to take video footage from the representation of the billboard. During the data collection phase, the distance and also the angle of the camera will be varied to get a more varied data set to train on. It will also provide the possibility to test the influence of certain conditions. Next to the different camera placements and angles, the light levels will be controlled.

### 4.3 Defective tile detection

With the collected data an object detection model will be trained. The collected data will be used to train parts of the Faster R-CNN framework. The training will be monitored with TensorBoard to see if the model is improving over time. The total loss function graph is one of the more important graphs in this process. When the loss function is beneath 0.10, the training will be stopped. When the model is well-trained, the complete model needs to be evaluated. Then, 1234 new images will be reviewed by the model. First, the overall performance of the model will be calculated by making up a confusion matrix. Derivations from the confusion matrix will inform us about the accuracy of the model and the false positive rate. A recommendation value for a threshold will be calculated based on the percentiles table to reduce the false positive rate. There will also be a ROC table which will help to alter the threshold in the future. The second part will evaluate the system under different conditions. These conditions were changed during the data collection phase. A permutation test on the differences of the means between these conditions will be used to see if there is significant importance. In order to test the difference between the three means of the perspectives, a one way ANOVA test will be performed together with a Post-test to see where the actual differences are.

### 4.4 Thresholds and notifying

For the last phase, thresholds and a notifying system need to be developed and tested. The desired output of the system needs to be defined in consultation with the people who are going to work with the system. Besides the threshold that considers if a new detection is a true positive, there needs to be a threshold that sets the minimum amount of detections in the same place before sending output. This output is still undefined and needs to be researched further again together with the employees that will eventually use the system. They will provide qualitative information about their preferences that can be implemented into the system. Finally, the complete system will be evaluated with the innovation manager to find the last design flaws.

## 5 Billboards internal workings

During the literature review, the different failure modes of the total billboard were discussed. From H. Bierma his research the conclusion was drawn that a defective LED tile was the most occurring defective. In order to understand how the billboard works and so how the billboard breaks, more knowledge about the blueprints of the billboard is required. This chapter will start broad with the complete billboard in scope. Later the focus will mainly be on the failure modus of component that will be tackled during this research: the defective LED tile.

### 5.1 Outdoor billboard overview

All the billboards that Hecla builds are custom made. Despite this fact, a lot of things are in common between these billboards. Therefore, one billboard will be worked out to get a better overall conception for most of the billboards. A billboard consists out of parts which on their behalf consist out of smaller parts. When starting on the biggest scale, there is a complete billboard. This billboard then consists of LED Panels. Every LED panel consists out of 24 so-called LED tiles, six in height and four tiles in width. Every LED panel powers and controls all of its 24 LED tiles. These LED tiles consist of 12x12 RGB LEDs, resulting in a total of 144 LEDs per tile. On the back of each tile, there is a circuit board that powers and controls all the specific LEDs. A schematic overview can be found in figure 1.

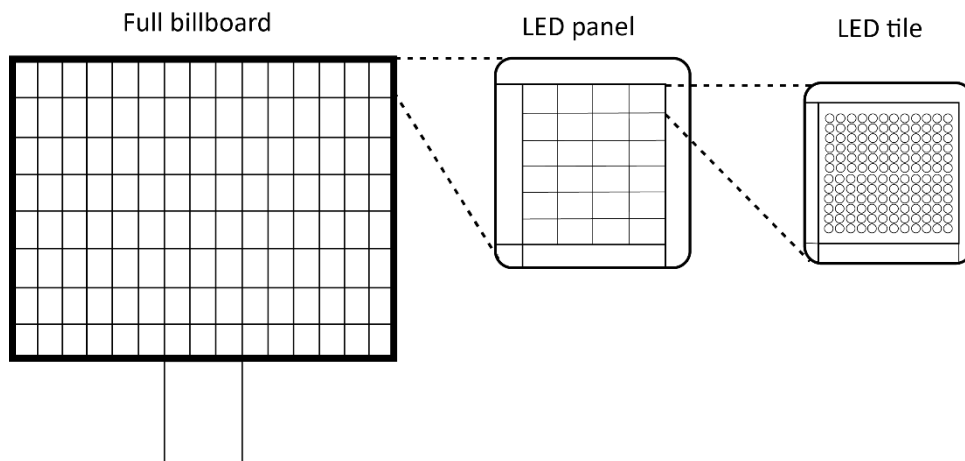


Figure 1 Overview of parts from a billboard

An example of such a custom billboard is the one located at Brienenoordburg A16, Rotterdam. This billboard is eight LED panels high and fifteen panels wide. This results in a total of 120 panels which all together have a total surface area of 106m<sup>2</sup>. These panels are placed on a metal construction which is constructed by a third party. With this information, the resolution of the billboard can be calculated.

The calculation for the resolution for this billboard will be as followed:

*Brienoordburg resolutions calculation.*

Width: 15 panels \* 4 tiles \* 12 LEDs = 720 pixels

height: 8 panels \* 6 tiles \* 12 LEDs = 578 pixels

This resolution of 720x578 matches a 1.25:1 aspect ratio. As described before, the billboards are being monitored by cameras. The camera used to monitor the billboard at the Brienoordburg is an Axis Q 6035-E camera. This camera is capable of filming 1080 HD at 30FPS and has a 20x optical zoom with auto-focus. An example of the footage that is delivered by this specific camera is given in figure 2.

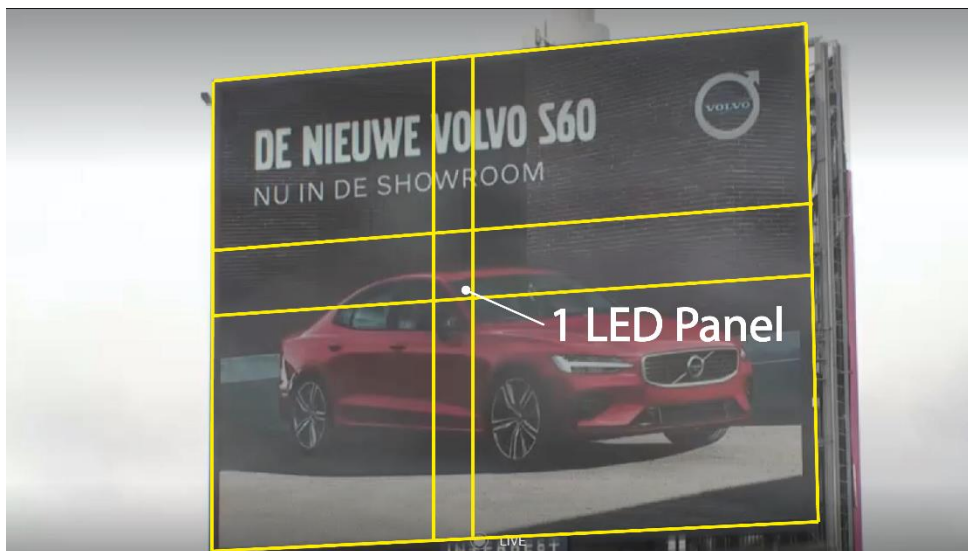


Figure 2 Example footage from the camera with panel size indicator

## 5.2 LED Panels

After the worked out the global overview, it is time to have a closer look at the components that make up the total billboard: the panels. Starting off with the brand that makes these separate panels. The panels are made by Lighthouse and the specific model that Hecla uses the most is the Impact I9. This type is 768mm wide, 1.152 mm tall and weigh 39 kilograms. Inside these panels, there are a lot of different electrical components. These components can be found in table 1.



Table 2 Component list of LED panel

Component	Description	Quantity
1	Intelligent Module (IM) including IM driver / LED PCBA	24
2	New Panel Controller Board (NPC)	1
3	Environmental Management Board 2 (EMB2)	1
4	Temperature Data Board (TDB)	1
5	25W/12V Switched Mode Power Supply	1
6	100W/ 5V Switched Mode Power Supply	1
7	300W/5V Switched Mode Power Supply	2
8	300W/3.3V Switched Mode Power Supply	1
9	12V DC Fan	2
10	EMI / RFI AC power line filter	1

All these components are susceptible to failure at some point, but for this research, the focus will be on the LED tiles. This is because H. Bierma showed in his thesis that this component breaks the most. As shown in the table, there are 24 LED tiles (Intelligent Module) placed onto each LED panel. All LED tiles have their own driver electronics, a data cable, and a power cable to operate the LED tile. Each row and column of LEDs at the tile are addressed independently by a so-called IM address. In the next section, these LED tiles will be worked out in depth.

### 5.3 LED tiles and failure modus

The interesting part for this research is actually on how these LED tiles break and how do they affect the appearance of the billboard. There are different failure modus which result in a malfunctioning tile. According to the datasheet of the LED tiles, there are two different types of defects. One is just a single tile that malfunctions and the other side there are the defects that affect a whole column of the LED panel. See figure 3.

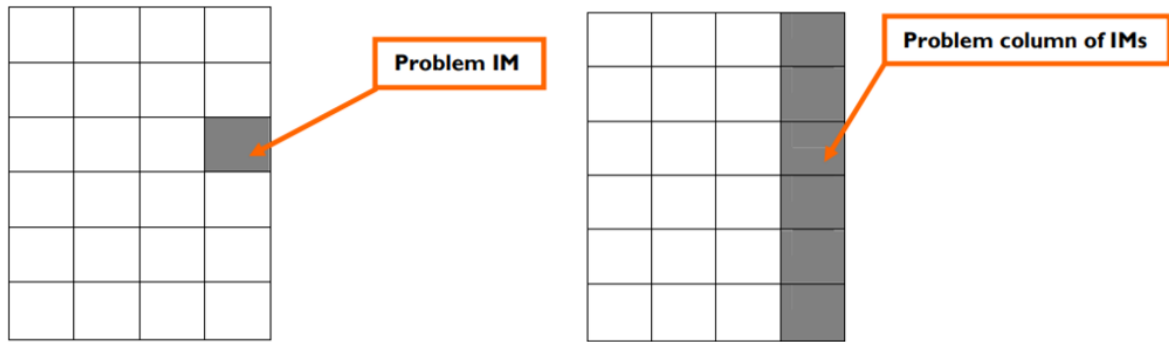


Figure 3 Different types of problems in LED tiles (IM)

For both problems, there are three common phenomena of malfunctioning. The first one is an overall poor functioning LED tile. This results in a tile that has interference or is flickering. The most common causes of this behaviour are a malfunctioning NPC (controlling overall tile performance), damaged PCBA (circuit board) or loose cables for data and or power. The second issue is the color problem. This can be caused by wrong settings for the tile or, again, a malfunction or broken NPC. The last malfunctioning behaviour is an image problem. In this case, the images are not placed in the location where they are supposed to be. This last behaviour is caused by wrong settings in the IM address, which has everything to do with the tile's position.

Next to these internal failures modus, there are other reasons that break these LED tiles. During H. Bierma his thesis time at Hecla, he had a conversation with Jan Löbker, Sr. Operational project manager. He explained that about  $\pm 90\%$  of the defects in tiles are caused by water damage resulting in a complete defect of the concerning tile. Then, the complete tile needs to be replaced. When such a tile is completely defected due to water damage, it completely stops emitting light. This failure has for the most billboards a higher severity than the other failure modus and it occurs the most. Therefore, the aim of the prototype will be to detect these water-damaged tiles that do not emit any light.

## 6 The indoor setup

To acquire a trained model with the desired accuracy, a lot of data is required for training. The time for this research is limited so there will not be enough of this data. Therefore, an indoor setup will be created to mimic the real world situation. This way it is easier and less time consuming to get the data that is required for training. This process consists of two parts. The first part is the generation of images that represent broken screens. The second part is acquiring data about the ‘broken’ led screens with a camera just like it is working in the outdoor situation. This chapter will elaborate on how this indoor representation of the outdoor situation is set up, data synthesis and data collection.

### 6.1 Setup components

In the previous chapter, the inner workings of a billboard were described. This knowledge can be used to mimic the real world situation indoors. An indoor setup (IDS) is more convenient than the real situation because the camera placement can be controlled easily as well as the environment around the billboard. The situation outside will be represented by three components inside.

The first one is the billboard. This will be represented by a movable RGB LED screen from the brand Lighthouse with a resolution of 1008x588, a width of 260.5 cm and a height of 151.5 cm. The aspect ratio is 1.85:1. Unfortunately, this does not match the exact aspect ratio of billboards outside which is 1.25:1. Normally a difference in aspect ratios would result in a deformation of shapes and distortion of the images[16] that are visible on the screen. However, in this case, it will cause no problems in mimicking the defective tiles with this IDS. The defective tiles outdoor are fully square and the squares on the IDS have a height of 64 mm and a width of 63 mm. This minimal difference will not be captured with the cameras and is therefore negligible.

The second component is the camera, which will be represented by a webcam in the IDS. In the outdoor situation cameras from the company Axis are used. These cameras are Full HD(1920x1080), focus automatically and have an autoexposure feature. The IDS equivalent for this camera will be the Logitech HD Pro Webcam C920. The webcam has the same resolution as the outdoor cameras and it is also automatically focussing and has the same autoexposure feature. The webcam will be connected to a computer via USB 2.0 to operate it.

An important component in the outdoor situation is the environment around the first two components. This factor needs to be represented in the setup as well. Therefore the choice was made to control the light levels during the indoor setup. This will be used to simulate the day and night circle and darker and brighter days that are outside caused by weather. Light is an important factor in taking video in general because the quality of an image decreases when the light levels are dropping[17]. For these reasons, it is taken into account during the experiments in the IDS by controlling the lights.

## 6.2 Images synthesis

Now the indoor setup is worked out, the defective tile can be mimicked. From chapter 5, conclusions were drawn on how a defective tile behaves. When a defective tile occurs, all of its 144 LEDs will stop emitting light. This will result in a square of no light emitting LEDs, visible as a black rectangle. An example of this is visible in figure 4.



Figure 4 Example footage of billboard with broken LED tile

With the above-described setup, this can be mimicked by images that have black rectangles edited into themselves. Since black pixels in rectangles result in LEDs being off at the screen. For this reason, this simulates a broken LED tile perfectly. These synthetic images are automatically generated by a code. This code first extracts separate images from stock footage video files. Then, it randomly selects one of these images and edits in a black square at a random position. The size of this black square was determined by a calculation. Footage from the Brienenoordburg billboard with a broken LED tile was analysed and the number of pixels with the broken tile was counted. The number of pixels was calculated back to the dimensions of the square that is pasted into the synthesized images to mimic a defective tile with precision. Finally, the software saves the image with a black square as a new image which then can be displayed on the LED screen via a slideshow. This synthesis process is schematically presented in figure 5. Example images, as well as a picture of the full setup, can be found in Appendix A.

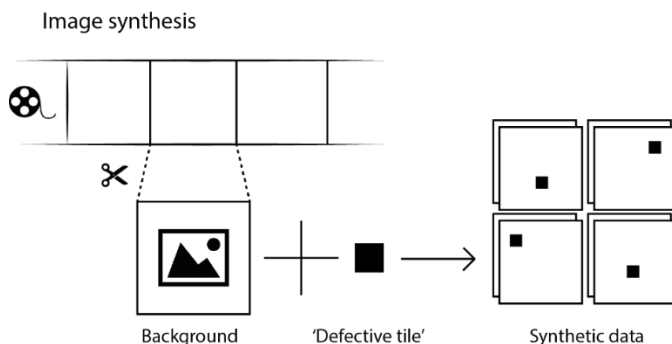


Figure 5 Synthetic data generation process

### 6.3 Data collection

The final step of this phase is collecting the data with the Logitech webcam. The earlier compiled images are displayed one after another on the LCD screen. Then, the webcam is pointed at the LED screen to capture the images and simulate the real world situation. In this real-world situation, the cameras are not always close and/or perpendicular to the screen. Hecla monitors a total of 16 cameras that are pointed at the billboards. Not all of these cameras are suitable for computer vision solution because it is physically not possible to see the defectives because the cameras are too far away or the angle with the billboard is too high. Therefore, four cameras were selected to be mimicked in the IDS. These were selected because they are close to the billboards and the angle between billboard and camera is small enough to see the defective tiles.

The four selected cameras are (example footage from the camera in appendix A) :

- RM028 Leiden
- RM039 Eindhoven
- RM065 Rotterdam Brienoordburg A16
- RM084 Zoetermeer A12 Zevenhuizen

When visually analysing the video streams from the cameras three conclusions can be drawn. The first one is that two out of these four billboards are really close to the camera and two are a little bit further. The second conclusion is that the angle from the cameras to the billboards is different in all four situations. And last but not least, the lightning conditions differ over time due to the weather cycle. These three aspects need to be controlled during the data collection inside to train the model on a setup that is comparable to the real world situation.

#### 6.3.1 Camera angle, camera distance, and lightning

The three mentioned conditions will be varied during the data collection. This will result in a diverse set of images to train the model for all sorts of situations. This way it will also detect defective tiles in situations less optimal.

First off, the angle between the cameras and the billboard. During a visual analysis the conclusion was drawn that all the cameras have different camera angles with respect to the billboards. Therefore, this needs to be taken into account for the IDS as well. Five different angles were chosen to represent all these most of the angles. Next to the varying angle of the webcam, there are different locations of the black squares. These locations are randomly spread over the LED screen. This will contribute to the angle differentiation in the dataset because this changes the angle as well.

Then there is the distance between the webcam and the LED screen that needs to represent the real world situation. As said before, from the selected four cameras, two billboards are far away and two billboards are close to the camera. It is necessary to know how much pixels in the image contain data about the actual billboard. Therefore the billboard cover rate of the image will be calculated. This

will be done by counting the pixels containing data about the billboard and then divided by the total amount of pixels. When this is multiplied by a hundred the cover rate in percentages for a certain image. This is necessary to determine the distance between the webcam and the LED screen. For the closer situated cameras, this results in a billboard cover rate in the image from 60%. For the cameras that are positioned farther away, this percentage is 25%. To get the same cover percentage in the IDS, the webcam will be located at 2m and 3m. This will result in the same cover percentages of the billboard in the images. A schematically representation of the IDS can be found in figure 6.

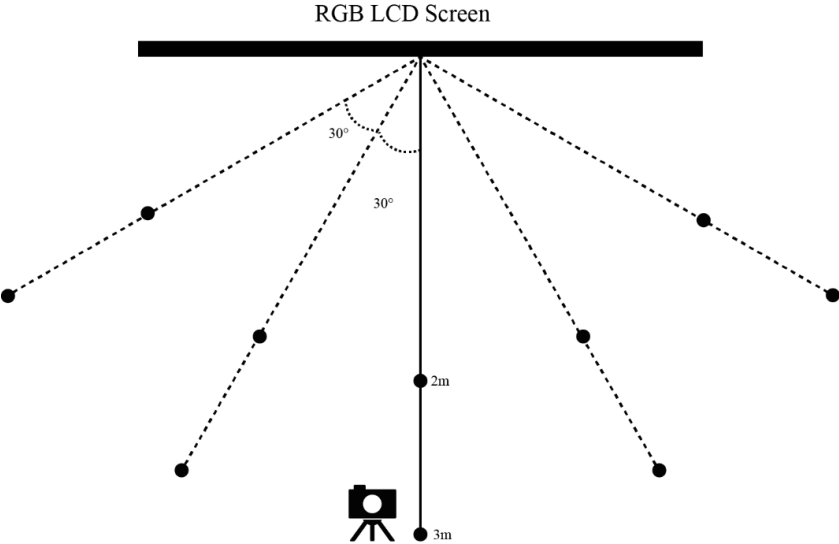


Figure 6 Schematically representation of the IDS

The last condition is the light level during capturing with the webcam. During the day the light level change and because lightning has a significant impact on the image quality [17] this needs to be taken into account. The two biggest differences in light levels are day and night so these two will be represented during the data collection phase in the IDS. The light levels from during the day will be represented by the lights being on and the light levels of the night will be represented by the lights being off. In total, a lot of different pictures were taken with all sorts of conditions changed. The number of images per different setting can be found in Appendix B.

## 7 Model training

In the previous phase, data was collected to train the model. In order to train the model in good fashion, this data first needs to be pre-processed. When this has been done, the training will start. This chapter will dive into the data pre-processing and the actual training of the model.

### 7.1 Data pre-processing

The training of the model requires the data to be in a specific format. The first requirement is that the images have to have the same filetype. Additionally, the images should not be too big because this will increase the training time a lot due to the fact that the code has to analyse more data. And lastly, the images have to be annotated with labels. This is to teach the model what in the images has to be detected.

The data that was captured with the webcam is in .mp4 format. From these video files, frames will be converted to separate images. This is done by the code used earlier in the images synthesis part but this time the code will just cut out images. Two different filetypes can be chosen: PNG and JPEG. The PNG filetype has transparency support and works really well with images containing text. The JPEG filetype, on the other hand, has a good compression rate but no transparency support[18]. In this case, the JPEG filetype is more suitable because the images that are going to be used do not contain transparent parts. The other positive side of JPEG above PNG is that it is really good at compressing the image without losing too much detail. This greatly reduces the size of the images which on its behalf then reduces the training time. Therefore the images cut out of the raw material will be in JPEG format.

The cut-out images are still between 160Kb and 360Kb. This is the result of the still high resolution of 1920 by 1080 pixels. The pre-trained model that will be used prefers images with lower resolutions to train faster. Therefore, the resizer script from EdjeElectronics at GitHub will be used to resize all the taken images to half of their original size. The result of this script is that all the images now have a resolution 960 by 540 pixels and the file size decreased by 50%.

Now all images have been modified to the right filetype and the right image size, they have to be annotated. This is to tell the model what it is trying to detect in the image. In the case of damage detection, it tries to detect the black squares in the images. Therefore, these black squares will be annotated with the LabelImg software. This software is used to draw bounding boxes over the black squares. Then, the software exports the coordinates of the bounding boxes to an XML file. The annotating process is shown in figure 7.

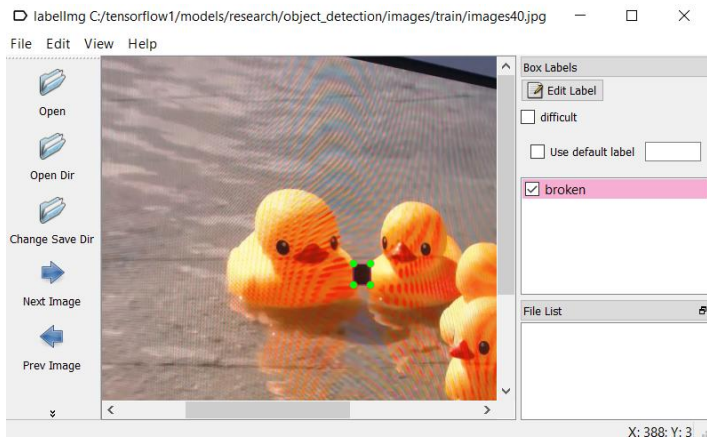


Figure 7 Data annotating process

The last step of pre-processing is to convert the XML files from LabelImg into readable CSV files that are used by the training code. This is done by another code from EdjeElectronics which cuts out the necessary information from all the XML files and puts it from all the images into one CSV file.

Examples of both of these files can be found in Appendix B.

## 7.2 Deep-learning framework

To achieve object detection inside the system, a convolutional neural network (CNN) will be used. These neural networks consist of layers, neurons, and connections. Layers are columns of neurons which are connected to different layers by connections. These connections have a specific weight which is just an arbitrary number. When a new number arrives at the neuron, the value will undergo a set of mathematical calculation including a multiplication with the weight of the specific neuron[19]. Then, the adjusted value will be passed over all the neuron's connections to the next layer.

The first layer of the network is called the input layer where the last layer is called the output layer. The input layer sends all values into the neural network which are called the hidden layers. In the case of the prototype, the image its pixels are fed into the input layer. The output of the neural network consists in case of the prototype out of two neurons representing defective tile or nothing. The values at the output layer are between 0 and 1 and represent a certainty of being correct. An example of a simple neural network can be found in figure 8.



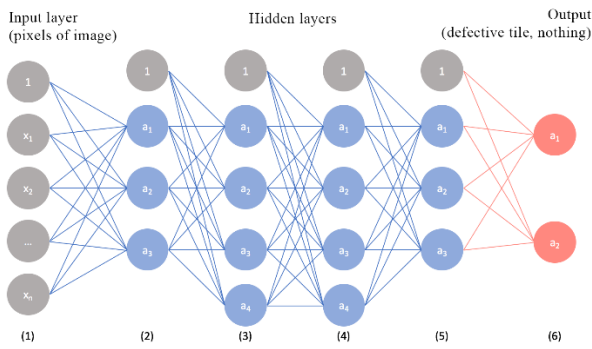


Figure 8 Example neural network setup

During training, the weights of these connections between neurons are tuned and tweaked to reach a state where the predicted output of the model equals the actual output. Setting up such a model from scratch takes a lot of time, training the model requires a lot of data and the efficiency might not be optimal. Therefore a deep learning framework will be used as a base where only specific parts of the framework will be altered during training to detect the desired objects.

One of the requirements of the system is that it can operate in real-time. In order to satisfy this requirement, the Faster R-CNN model was chosen. This model was proposed by Girshick et al. in 2016 and operates at 5-17 FPS[20]. It is a combination of a Region Proposal Network(RPN) and the predecessor of Faster R-CNN, which is Fast R-CNN. This state-of-the-art framework is according to B. Liu et al. one of the best ways to detect objects using CNNs series[21]. In the next section, the inner workings of this model will be explained.

### 7.2.1 Faster R-CNN framework

The framework of Faster R-CNN is, in fact, a combination of two systems combined. The first system is a Regional Proposal Network and the second one is Fast R-CNN, the predecessor of Faster R-CNN. The total Faster R-CNN contains four aspects in order to detect and classify an object[22]. First, the image is pushed through a pre-trained CNN which is already capable of detecting objects. This pre-trained CNN is called a feature extractor. In case of the prototype this the pre-trained Inception V2 model, which has been trained on the COCO dataset. This feature extractor generates a so-called feature map from the input image which highlights parts of the image that might contain useful patterns for classification[23]. Then, the RPN generates candidate regions that may contain objects. It does this by using a 3x3 sliding window that moves from anchor point to anchor point over the feature map. At every anchor, it generates  $k=9$  anchor boxes to be evaluated by two different layers of the RPN network[24]. The first layer is the classification layer (cls) which outputs  $2k$  scores whether there is an object or not. The second layer outputs  $4k$  rough coordinates for the bounding boxes of the detections. An overview of the RPN network can be found in figure 9.

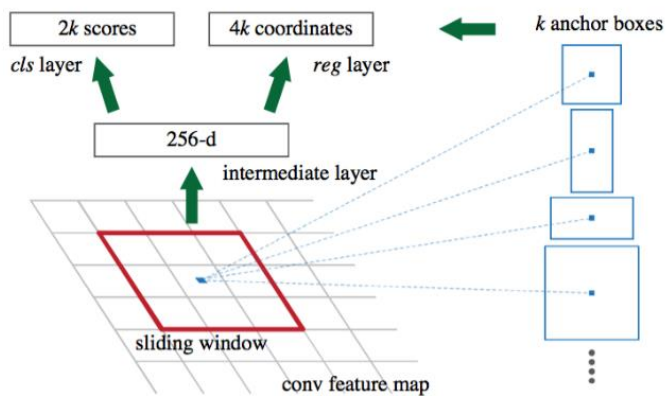


Figure 9 Overview of the RPN algorithm[20]

When RPN found object proposals, these proposals will go through the second part of the framework, the Fast R-CNN part. Where the RPN only provides information about the presence and a rough location, the Fast R-CNN part provides the classification and exact bounding box. It does this by combining the proposals from the RPN together with the feature map. This data will be fed into a so-called Region Of Interests pooling layer of Fast R-CNN[21]. From there, they will be passed into the classifications layers of Fast R-CNN. These classification layers refine the bounding boxes from the RPN and classify the objects in the image. The classification and bounding boxes will be saved in two separate arrays which later will be stored in the database of the system. An overview of the total system can be found in figure 10.

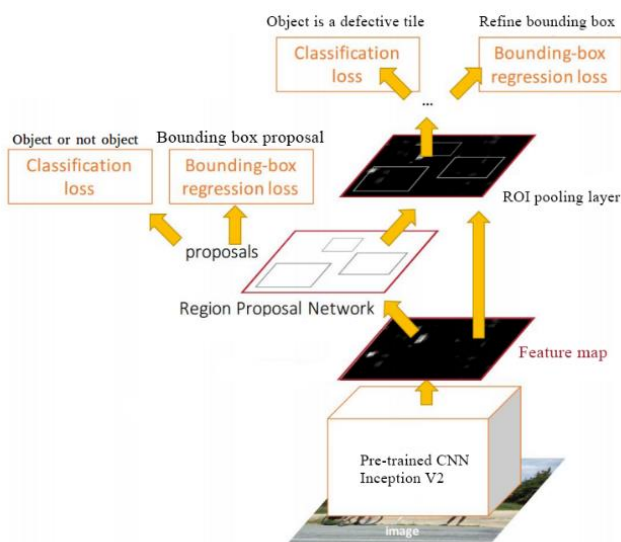


Figure 10 Overview of the total Faster R-CNN system[20]

### 7.3 Training

All the data has now been processed to the right format, size and with the accompanying bounding box information in a .CSV file. The actual training of the system can now begin. As explained in the previous chapter, only certain parts of the Faster R-CNN parts will be trained to achieve proper results in a limited time. These parts are the RPN neural network and the Fast R-CNN part. The training is an iterating process that evaluates images and adjusts the weights of connections between the neurons. The goal of these iterations with adjustments is to get the model's predictions closer to the actual ground truth of the image. The dataset that will be used to train on consists out of 3474 RGB images with a resolution of 960 by 540. These images are, as earlier mentioned, taken with different perspectives and light levels. In every image is one bounding box which sets the number of bounding boxes equal to the number of images. The only class, which is the object, that is going to be detected is the class 'Broken'. The dataset is divided over images for training and images for testing with a division of 0.81 for training against 0.19 for testing. The training will be excavated with the TensorFlow-GPU 1.12.0 framework. The training will take place on a Lenovo P50 ThinkPad with an Intel Core I7 6700 HQ @ 2.60 GHz with 16GB of RAM and an NVIDIA Quadro M1000M with 2GB of VRAM. The initial learning rate will be 0.0002, which is the standard initial learning rate of Faster R-CNN. Then, the training was ready to start.

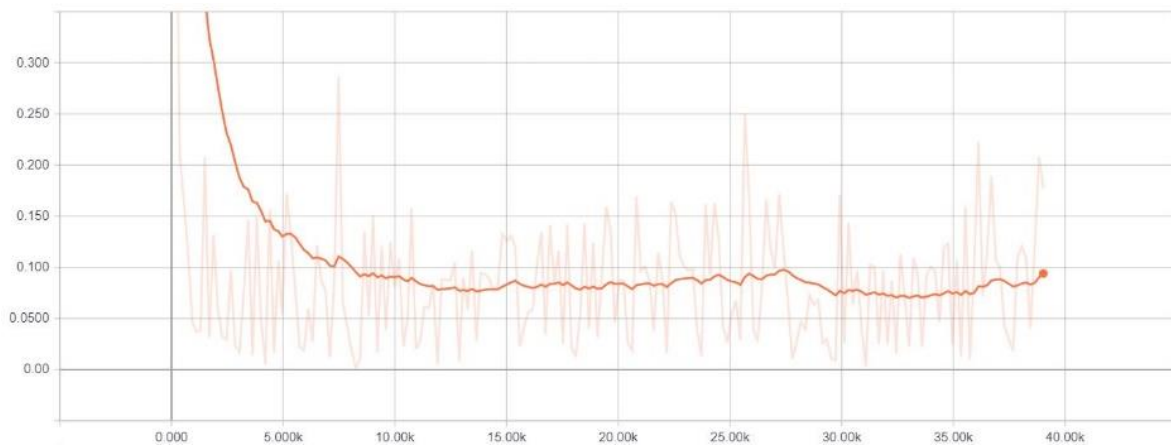
During training, it is important to monitor the loss functions of the model. These functions inform about the progress that is made during the training. In total there are five different loss functions that will be monitored. The definition of these loss functions will be given in table 3.

*Table 3 Meaning of loss functions*

<b>Loss function</b>	<b>definition</b>
RPNLoss localization loss	Localization Loss from the RPN.
RPNLoss Objectless loss	Loss of the RPN proposals that classifies if an anchor box is an object of interest or background.
BoxClassifierLoss Localization loss	Localization Loss of Fast R-CNN network
BoxClassifierLoss Classification loss	Classification loss of the Fast R-CNN network
Total Loss	A combination of all losses

The RPN losses are there to monitor the step before it moves into the Fast R-CNN model. This is important to find problems halfway the process. The BoxClassifier losses are from the model as a whole. The most useful graph to monitor is the TotalLoss graph. This is a combination of all losses which is easy to monitor during the training.

During the first hour of training, the combined total loss function dropped significantly, which suggests that your model is training correctly. According to Evan Juras his article[25] about training a Faster R-CNN computer vision model, the model can be considered trained when the Total Loss drops below 0.05. Unfortunately, this did not happen so at step 25.000 the decision was made to decrease the initial learning rate from 0.0002 to 0.00002. This resulted in a small reduction in loss level but not significant enough to reduce the learning rate even further. The total loss graph is given down below and the other loss function graphs can be found in Appendix C. The bright orange line is a smoothed out to see the overall trend. The raw data points are displayed in the light orange.



*Figure 11 Graph of the Total loss*

As seen in figure 11, the loss function reached a plateau. At a final loss value of 0.091, the training was forced to stop. The model was then saved to a useable inference graph for the actual detections. In the next section, the trained model will be evaluated on its accuracy and false positive rate in order to see if the training was successful.

## 8 Evaluation model

The trained Faster R-CNN model reached a plateau at a considerably low loss function value. These are signs of a well-trained model but it does not give information about the model's performance in the real world. In this section, the model will be evaluated on for the system unseen images from the real world. This in order to obtain insights about the system's performance.

### 8.1 Method

The goal of this part is to find the model's overall performance and the performance per condition that was changed during the data collection. This will be done by loading images into the recently trained model and letting the model detect if the images contain any broken tiles. The outcome of this test is a confidence level of containing a defective tile per image. These confidence levels can vary between 0 and 1, where 0 means that the model does not think the image contains any defective tiles and 1 is when the system is absolutely positive that the image contains a defective tile.

First, a test on the global performance of the model will be performed. A confusion matrix will be made up and several derivations from this matrix will be given. Next to these matrix derivations, a threshold will be defined. This threshold sets a minimum confidence level to be recognized as a detection. E.g. if the threshold is set at 0.5, an image with a certainty of 0.3 of containing a defective tile will then be classified as no defective tile in the image. Using this threshold will give different results in the confusion matrix which will benefit the performance of the system.

The second evaluation method concerns the importance of the different variables. The altered conditions during the data collection were: camera distance and height, light-levels and angle with respect to the screen. It is useful to find out if these conditions influence the performance of the system. The images are used are again unseen to the model and are classified in different groups with the same conditions. The groups of images with the same settings can be found in Appendix B. All the images from these different groups are run through the model and the certainty of containing a defective tile was saved per image and group to a database. The average mean per group was calculated so that every condition had an average certainty. This average means of certainty will be tested on significance by a permutation test. This test confirms if there is a significant difference between the means or not. The last altered condition, camera with respect to the screen, will be tested in a different way for the reason that it contains three means that need to be tested against each other. This will be done by a one way ANOVA test. If the result of this test is positive, a post test will be performed to find out which means differ significantly with respect to each other. At the end, the results will all be analysed and a conclusion will be drawn.

## 8.2 Results

With the above-described methods of evaluation, the images will be evaluated on the certainty of containing a broken tile. In total 1234 images, 50% with broken tile and 50% without broken tile, will be run through the model. The results will be discussed in this chapter and the visual results will be in appendix C. The results will be split up in two different parts. The first parts focus is on the overall model performance. The second part will provide the results of the earlier mentioned conditions that were changed during data collection. This in order to see if these variables have an influence on the performance of the system. Next to these two statistical tests, there will be an evaluation on the speed of the model.

### 8.2.1 Overall performance

To find the overall performance of the model, the accuracy and precision need to be calculated. Before this can be done, an initial threshold needs to be set. This threshold defines from what certainty a detection can be considered valid. First, a threshold of  $p=0.5$  will be chosen. Later, a look at the descriptive statistics of the complete dataset might lead to a better threshold. The threshold will shift the numbers in the confusion matrix and therefore shift the derivatives true positive, true negative, false positive and false negative. In case of the system, especially true positives and false positives are important. This is because true negatives and false negative will be ignored by the system at all times. True positives are the images that actually contain a defective tile and the system predicts a defective tile. The false positives are the images that do not contain a defective tile but the system predicts a defective tile. Therefore are these two important to monitor in the confusion matrix. Examples of these two derivatives can be found in Appendix C.

By altering the threshold, the numbers in the confusion can move places. First, a confusion matrix with threshold  $p=0.5$  will be set up. This confusion matrix is given in table 4 down below.

*Table 4 Confusion matrix with  $p=0.5$*

Threshold	predicted		
$p=0.5$	n=1234	positive	negative
Actual	positive	491	126
	negative	65	552

With this data, the accuracy can be calculated by the following formula.

$$Accuracy = \frac{true\ positive + true\ negative}{total\ positive + total\ negative}$$

With the data of table 4 the calculated accuracy of this model= 0.8452. Two other values that can be calculated using this matrix are the false positive rate and precision. In the case of the system, it is especially useful to know the false positive rate. This due to the fact that the system is continuously monitoring it is really undesired to receive notifications from the system when there are no broken tiles. When there is a broken tile, the system will probably find this broken tile eventually even with a higher threshold.

$$False\ positive\ rate\ (FPR) = \frac{false\ positive}{false\ positive + true\ negative}$$

The false positive rate= 0.1053. This is still a significantly undesired high number. When changing the threshold to a higher value, this will decrease. In order to find the ideal threshold, a better look at the descriptive statistics of the certainties is required. These statistics were calculated using SPSS. The table that is useful in this case is the one with the percentiles, which can be found in table 5 down below. Calculating these percentiles will tell from which certainty how much of the data is included in the data set. E.g. the value at the 95<sup>th</sup> percentile of images without a broken tile is 0.6029. This means that 95% of the images without broken tile has a certainty lower than 0.6029.

Table 5 Percentiles of the dataset

		Percentiles						
		5th	10th	25th	50th	75th	90th	95th
Images	Without broken tile	,0022	,0032	,0153	,1000	,2986	,5091	,6029
	With broken tile	,0766	,2449	,5608	,7792	,8823	,9186	,9425

As earlier mentioned, a low false positive rate is desired. This is because the system should not be notifying the company if it is not completely certain of a broken tile. When there is a broken tile, but

the system does not recognize it as a broken tile in the first place, it is likely that the system will find it eventually when the content is changing. With these percentile calculations, a new threshold will be set at 0.6029. This means that 95% of the images without a broken tile will be classified correctly. When setting this new threshold, a new confusion matrix will be calculated. This confusion matrix with a higher threshold and new numbers is given in table 6 down below.

Table 6 New confusion matrix with  $p=0.6029$

Threshold $p=0.6029$	predicted		
Actual		positive	negative
	positive	448	169
	negative	30	587

With this new data, the new accuracy and a new false positive rate can be calculated. The new accuracy= 0.8379 and the new false positive rate=0.0502. These values can be compared to the accuracy and FPR at a threshold of  $p=0.5$ . The decrease in accuracy is less than one per cent where the decrease in FPR is 52%. Therefore, this threshold can be seen as an improvement of the system with a final accuracy of 83.79%.

Right now this seems a reasonable trade-off between accuracy and the false positive rate. Nevertheless, if the Hecla decides that the false positive rate is too high in the future, a useful graph to evaluate is the ROC Curve. This graph is often used in binary classifiers [26] and it plots the *true positive rate* against the *false positive rate*. It informs about the false positive rate when the true positive rate increases. The trade-off can be calculated in percentages and a new threshold can be set according to these percentages. Concluding, the initial threshold will be 0.6029 but if in practice it happens to be that or the TPR is too low or the FPR is too high it can be adjusted according to this graph which is shown in figure 12.



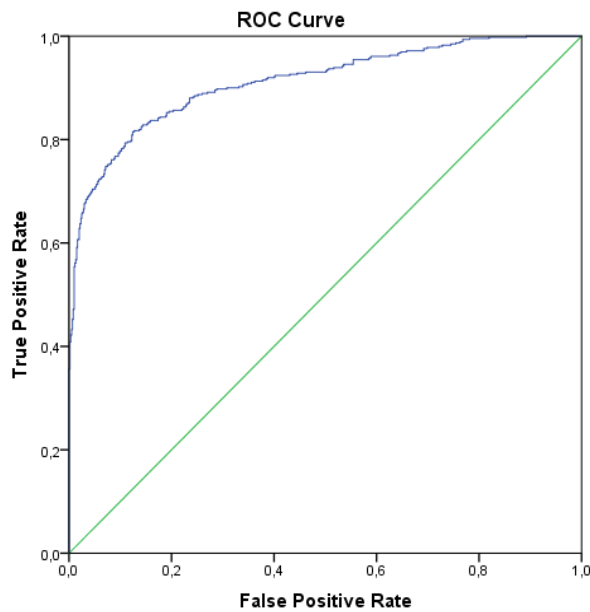


Figure 12 ROC curve from the synthetic generated data

### 8.2.2 Performance per condition

Where the previous section focused on the overall performance of the system, this part will focus on the importance of several variables that were changed during the data collection. The difference of means measured in certainty will be tested on significance. Finding if a specific altered condition is significantly important in the average of the mean, is useful for two reasons. First, in every situation, there is a different set of conditions. If it is known how the system performs under these conditions, the earlier set threshold can be adapted accordingly. Secondly, recommendations for screens that will be placed in the future can be written. E.g. if it is known that the camera angle with respect to the screen is significantly important, a recommendation would be to place the camera perpendicular to the screen. To start off, the descriptive statistics of each condition will be shown in table 6. This will show us the average certainty measured per image of the specific category. After the descriptive statistics of these different groups, tests will be performed on the difference between the means of the conditions. This in order to find out if a certain conditions influence the mean certainty. The of the descriptive statistics can be found in table 7 down below.

Table 7 Mean certainty of images under different conditions

Light-levels	Mean certainty	Std deviation
High	0,631	0,117
Low	0,729	0,097

Camera height	Mean certainty	Std deviation
Low	0,683	0,110

High	0,653	0,132
------	-------	-------

Distance	Mean certainty	Std deviation
2 meter	0,755	0,087
3 meter	0,577	0,010

Perspective	Mean certainty	Std deviation
0°	0,749	0,081
30°	0,692	0,111
60°	0,513	0,077

Now the mean certainty per group is calculated, the difference between means will be tested on significance. The varies light-levels, camera height and camera distance will be calculated using an independent samples test with bootstrapping. This non-parametric test is based on random sampling with replacement. This bootstrapping test will output a 95% confidence interval on the difference between means of the group. When the interval does not contain zero, it can be concluded at a confidence level of 95% that there is a significant difference in means. The results from the different tests will be given down below in tables 8,9 and 10.

Table 8 Bootstrap for Independent Samples Test on the difference of means – Light levels

		Mean Difference	Std. Error	95% Confidence Interval	
				Lower	Upper
Light Levels	Equal variances assumed	-,09837	,04202	-,17550	-,01318

\*Results based on 1000 bootstrap samples

Table 9 Bootstrap for Independent Samples Test on the difference of means – Camera height

		Mean Difference	Std. Error	95% Confidence Interval	
				Lower	Upper
Height	Equal variances assumed	,02971	,04408	-,05670	,11852

\*Results based on 1000 bootstrap samples

Table 10 Bootstrap for Independent Samples Test on the difference of means – Camera distance

		Mean Difference	Std. Error	95% Confidence Interval	
				Lower	Upper
Distance	Equal variances assumed	,17959	,03811	,10495	,25527

\*Results based on 1000 bootstrap samples

The last group that will be tested on the differences of the means is the angle with respect to the screen. This requires a test on Homogeneity of variances (table 11) to find out if the variances can be assumed equal. The second test that will be performed is a one way ANOVA test (table 12), in order to see if there is a difference between means. Lastly, a Post-test (table 13) will be performed to find where this actual difference between the means is. The tables with these tests are down below and the results will be analysed in the next section.

Table 11 Test of Homogeneity of Variances on means of the different perspectives

Levene Statistic	df1	df2	Sig.
1,189	2	37	,316

Table 12 ANOVA test on the means of the different perspectives

Angle

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,348	2	,174	13,250	,000

Table 13 Post-test on the means of different perspectives

Multiple Comparisons						
		Mean			95% Confidence Interval	
(I) Group	(J) Group	Difference (I-J)	Std. Error	Sig.	Lower Bound	Upper Bound
0°	30°	,05630	,04183	,379	-,0458	,1584
	60°	,23601*	,04677	,000	,1218	,3502

30°	0°	-,05630	,04183	,379	-,1584	,0458
	60°	,17971*	,04677	,001	,0655	,2939
60°	0°	-,23601*	,04677	,000	-,3502	-,1218
	30°	-,17971*	,04677	,001	-,2939	-,0655

\* The mean difference is significant at the 0.05 level.

### 8.2.3 Speed

The speed of the model is important because one of the requirements is that the detections have to be in real time. This means that the content on the billboards needs to be controlled directly when the video streams come in at Hecla and that no delay will build up. Therefore, the speed per detection will be calculated. For the last time, the 1234 images will be run through the model. The difference between begin time and end time is the total test time. The formula for calculating speed per image is given down below.

$$\text{Speed per image} = \frac{\text{Total test time (sec)}}{\text{Total number of images}} = \frac{474.38}{1234} = 0.38$$

As shown from the calculation is the speed per image 0.38 seconds per image which is the equivalent of 2.60 images per second. The frames come in at 30 FPS at Hecla, which means that around 27 frames will not be checked and these frames will be discarded. This is no problem because at this speed the defective tiles will be picked up fast enough to be considered real-time. If this system will be scaled up, the control speed of the system will decrease due to the fact that there are in reality 20 screens to be monitored. It would take  $0.38 * 20 = 7.6$  seconds to check one frame from every billboard Hecla owns. Altogether, this speed can be considered high enough for the system's purpose.

### 8.3 Analysis and conclusion

This chapter started off by explaining two methods of evaluating the model. The first method focussed on evaluating the model on its overall performance. By testing images with broken tiles and images without broken tiles, a confusion matrix came into existence. Together with this confusion matrix the accuracy and false positive rate were calculated. Together with the percentiles diagram from SPSS a threshold was set. By setting a threshold, the false positive rate decreased a lot. The trade-off setting this threshold was a slight loss in accuracy. There was also the ROC curve, which gave insights on how more true positive predictions would influence the increased rate of false positive detections. Altogether, an overall accuracy of 83.79% was achieved.

The second part of the evaluation concerned the altered conditions during the data collection.

Several tests were conducted on the performance of the system with these different conditions. On the first three conditions, a permutation test was conducted. This test finds with a 95% confidence level upper limit and lower limit of the difference in means. For the altered condition light-level and camera distance, a significant difference in mean was found. The height did not have a significant difference between the mean. It can be concluded that the significant variables changed in the conditions do have an influence on the performance of the system. Under low light, the system performs better. This is probably due to the fact that under low light conditions the contrast is higher between the content and the defective tile. The camera distance is also a proved variable that has an influence on performance. This is due to the fact that the images taken further away from the screen contain fewer pixels about the defective tile. This makes it harder for the model to predict the presence of the defective tile which drops the certainty. The height did not influence the performance of the system significantly. An overview of the conditions that have an significant influence on the mean can be found in table 14.

*Table 14 Significance overview of conditions*

<b>Altered condition</b>	<b>The difference in mean significant (influence the performance)</b>
Light-level	✓
Camera height	X
Camera distance	✓

The Post-test and the ANOVA test on the angle with respect to the screen resulted in table 12 and 13. At a significance level of 95%, between all groups, except 0°-30°, a significant difference in mean was proved. This proves that angles between 0° and 30° do not influence the performance of the system significantly. For the other angles, it happens to be that a greater angle deteriorates the performance of the system. This is probably caused by the fact that defective tiles in these images appear smaller.

The last evaluation test on the speed resulted in 2.6 images per second. This result is high enough to state that the system is detecting in real-time fulfilling the requirement. It also proves that the system can be scaled up to detect defective tiles in more than one billboard.

Altogether, the system's performance can be considered good for the indoor setup. The overall accuracy of 83.79% can be considered as good but there is still room for improvement, especially before rolling it out to the real world situation. The conducted significance tests prove that the conditions camera distance, angle with respect to the screen and light-levels influence performance of the system. Therefore, the recommendation would be to detect for defective tiles with low-light, a camera as close as possible to the billboard and as perpendicular to the screen as possible.

## 9 The output of the system and the database

The trained model is just one part of the complete system. The other two parts are the data handling and the output to the user. The first half of this chapter is about data handling. The model outputs certainties per frame but this data needs to be stored in a database. When the data is stored, a different part of the system will check if certain parts of the screen exceed the thresholds for notifying Hecla. The second half of this chapter focusses on the output of the system to Hecla, which can be done in multiple ways. In this chapter, these different manners will be explained and evaluated. Finally, a choice will be made which will be implemented in the system.

### 9.1 The database

When the model detects a broken tile, it has to save the data to a database. This is done by dividing the incoming images into four quadrants, all having their own table in the screens database. These four tables represent one-quarter of the image. When a new detection comes in, the centre of the bounding box will be calculated. The coordinates of this centre depend on where the record will be saved to. When a detection with a certainty that exceeds the threshold comes in, its centre will be calculated. E.g. there is a new detection with its centre in the upper right corner. Then, a unique ID, the time, location and certainty of the detection will be saved to table 2. A schematic overview is given in Figure 12.

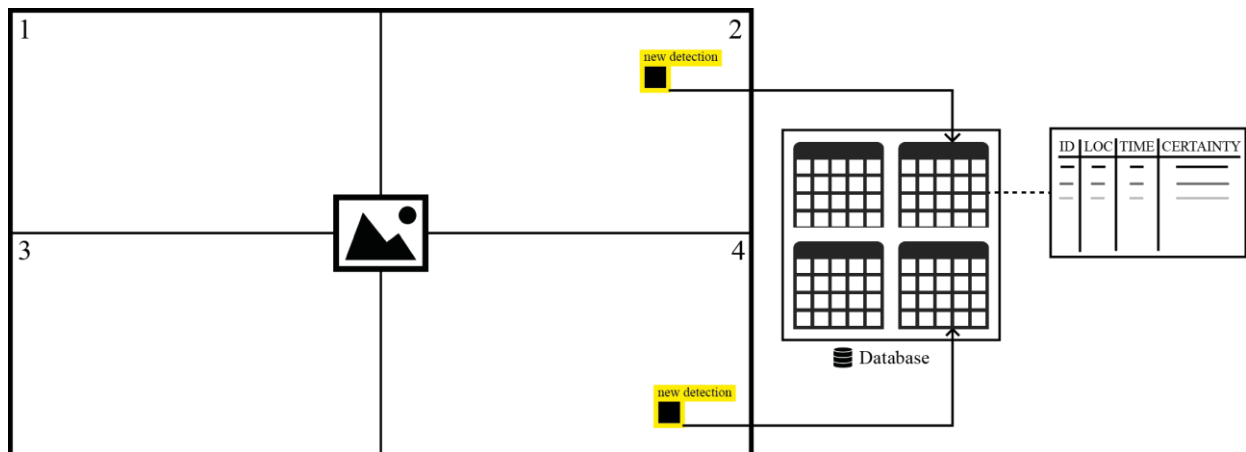


Figure 13 Schematic overview detection data handling

The database of the system is continuously being monitored on the number of records in every table. Earlier, a threshold at 0.6029 was set before detection was considered positive. To increase the confidence of the screen containing broken images another threshold is required. This threshold depends on what amount of positive detections in a table the system sends an email to Hecla. Defining

this threshold can be challenging due to the fact that it is really content dependent. Therefore an assumption will be made which needs to be altered in the future.

Assume the optimal case where a defective tile that is continuously been detected by the system. The speed of the system is on average three detections per second. After ten minutes, the table according to the location of the detections will contain 1800 records. This means that if the threshold is set to 1800, after ten minutes an email would be sent to Hecla.

This seems reasonable but this is only in the optimal case. When a defective tile is due to the content only visible for 1/10 of the time, it would take 100 minutes before the system sends an email. In consideration with L. Kuipers, these 100 minutes seemed justifiable and this threshold was therefore set as the initial threshold when testing with the real screens.

## 9.2 Output

The system's output is really important to the final user. For this reason, it is a requisite to research the possible outcomes of this system. The first decision before starting to look at the options is: does Hecla want the output to be in the physical domain or one in the digital domain. To answer this question a short interview with L. Kuipers was held to find the answer. He recommended staying in the digital domain. This because he thinks that flexibility is greater in this domain and therefore more suitable for the future. With this in mind, two different notification proposals are worked out to discuss further with the engineers at Hecla.

### 9.2.1 Types of notifications

Notifying the user by text notifications as SMS and WhatsApp are really fast and convenient. Within milliseconds the desired message arrives at the end user. SMS is the old fashioned one of the two. Unfortunately, with SMS it is not possible to send more than 160 characters in one message[27]. Therefore the text messages will be cut up in separate messages which might lead to higher costs depending on your subscription. For WhatsApp, this is not a problem because there is no text limit. A downside for WhatsApp is that a data connection is required to receive messages[28].

Next to the text notifications, there are email notifications. Just as text notifications are email notifications fast and reliable. Email notifications are also cheap and one of the biggest advantages is that they are permanent staying on the servers of the owner. This makes the emails accessible at all times from multiple devices at the same network.

### 9.2.2 Conclusion and result

The two manners of notifying the company were discussed with L. Kuipers. The decision was made to implement email notifications in the prototype and maybe for future plans add the feature to send notifications to mobile phones. The reason for that email notification was chosen, is due to the fact

that they are accessible from all devices and phones connected to the servers of Hecla. Another aspect in favour of email notifications is, that they can be archived better. Hecla is currently using an Enterprise Resource Planning (ERP) which is integrated into their email management system. The emails can be dragged into this existing system to be archived. It is useful to analyse the emails sent from the system at a later point in time. This in order to find flaws in the system and innovate where possible.

Now the conclusion is drawn, the actual email can be made up. Together with T. Ten Hove, from the division Technical support & ICT, four important variables from a detection were selected. The advice was given to also include a screenshot of the detection so that the staff can see what has been detected. Concluding, a set of variables and a screenshot of the detection will be included in the email. The four included variables in the emails are listed down below.

- Location of the billboard
- Date and time of detection
- Certainty of detection
- Direct link to the live stream of the billboard for manual check

The final output of the system is given in figure 13 down below. In the attachment of the email is a screenshot of the detection. A zoomed in version of this figure can be found in Appendix D.



Figure 14 Example output of the system



## 10 Conclusion

In this paper, a new computer vision damage detection product for the Audio and Visual company Hecla was proposed. The goal of the product is to improve the current workflow and improve the quality of the maintenance service. Together with Hecla a system was envisioned and prototyped in an indoor environment. The system was designed to detect one specific defect: a defective LED tile.

The system was evaluated by statistical tests. These tests showed how the system performed in several settings where the position of the camera and the light levels in the surroundings proved to have an important role. The main encountered difficulties by the system are the false positive detections. These detections are mostly tackled by set threshold although a minority will still be acknowledged as a defect by the system.

The foundation is laid for the further development of this damage detection system. However, the system is not ready to be tested in the real world situation. This current system is only detecting one type of defect and has not been tested in the real world. The most important recommendation is that Faster R-CNN needs to be trained with images from real broken billboards in order to be effective. Additionally, training with different defects is recommended to improve the system's effectiveness. Then, the system its full potential can be tested and evaluated for a possible entry in the service package of Hecla Professional.

# 11 Recommendations

In the given time, it was not possible to achieve the complete envisioned system. The decision was made to generate synthetic data to prove the concept of the system. This resulted in the indoor setup (IDS) which is described in chapter 6. Although the IDS represented the outside situation, there are still a lot of differences with the real world. A working prototype is now available but before deploying on the cameras, several aspects need to be tested more in-depth and some extra features are required. This chapter will elaborate on these recommended tests and features. Assuming that these recommendations are fulfilled, the first tests on real screens will be feasible.

## 11.1 Training images

During the data collection phase, about 3500 images were collected. These images were used to train the pre-trained Faster R-CNN model. In the end, a model with decent accuracy and a decent false positive rate were achieved. One problem with these images is that they are all taken in an indoor setup. When rolling out a system to the real outdoor billboards, the model needs to be retrained with images from the outside situation. The synthetic data used to train the current model is presumably not sufficient enough to find all the broken tiles at the billboards outside. To solve this, data about these outside billboards need to be collected and the model needs to be retrained with this data.

## 11.2 Database

The database as it is currently working is not optimal. The images are divided into four different quadrants and when detection is in the same quadrant it considers it to be in the same place. This can definitely be improved. The recommendation for this would be that every new detection will get its own table with the location. When the next detection comes in, the system needs to check if this detection is located close to earlier detection. When there is an earlier detection around the same place, it has to store the detection to the already existing table. If there is no table yet with that location, it should make a new one and save the record to the just made table. This way multiple detections on the same place will all get placed into the same table which makes it easier to tell if detections are at the same place. During the conversation with T. ten Hove, another idea came up to improve the systems database. In order to get more information about the defect, it would be useful to know which panel of the billboard is defect. An extra layer on top of the image with panel numbers will help to indicate better which panel contains the defective tile.

Another feature that needs to be added to the system, is access to the database with tables in an easy way. When this feature is added, it will also be possible to ignore detections in certain locations. This is required due to the fact that some content might contain black squares which to the model will be seen as defective tiles. This will result in false positive detections that will eventually send an email

to Hecla. The technical engineers at Hecla should be able to tell the system once that a location does not contain defective tiles and from then on the system should ignore that location.

### 11.3 Cameras

To monitor the billboards cameras from the company Axis are used. The most commonly used model is the Axis Q 6035-E. These high-end cameras are capable of delivering images suitable for defective tile detection. Unfortunately, the potential of this camera is not completely used. The live streams with the billboards are compressed a lot which results in an enormous loss in detail. For most of the live streams, the quality is too low for proper damage detection using this system. The first recommendation for the cameras would for that reason be to stream the full HD content to Hecla to optimize the potential of the system.

Next to the quality of the images, there is the fact that the cameras are most of the times far from the screen and not perpendicular. As shown in the evaluation chapter of this thesis, the angle between billboard and camera is really important to get true positive detections. The second recommendation would be to place the cameras closer to the screen and as perpendicular as possible. This will result in overall better performance of the system.

### 11.4 Boundaries and exceptions

In the current version of the prototype, the whole screen is continuously monitored. This means that not only the billboards in the images are evaluated but everything around the billboards too. To reduce the number of false positives even more, boundaries are recommended. These boundaries limit the system to certain parts of the image that in fact contain the billboard. This will also increase the speed of the system due to the fact that the number of pixels send into the CNN is smaller.

### 11.5 Planned timeslot detections

Detecting defective tiles in real content can sometimes be really hard. The defective tile can be part of the content displayed on the billboard which results in not being completely visible. Another aspect is the contrast between the black defective tile and the content. E.g. a defective tile is in the middle of a white background, it is more likely that the system will detect the defective tile. For this reason, a recommendation would be to plan in time slots where the content of the billboard is completely white. Defective tiles would be easily be detected by the system in just a few seconds. The system can be tuned to just detect in these specific timeslots for even better performance. Unfortunately, there are two downsides to this recommendation. The first one is that you have to schedule in white content for a few seconds every once in a while which might be undesirable by the customers. The second downside is that there is no information in between these planned timeslots. To solve this, a combination between the current prototype and these timeslots detection moments would be ideal.

## 11.6 Different feature extractor with higher accuracy

For the current system, the Faster R-CNN framework was used. The feature extractor, which is the first part of the Faster R-CNN, of this framework can be changed to get a different performance. The feature extractor used in the prototype is the inception V2 model. This model was chosen because it is a well-balanced model between speed and accuracy. This makes it easier to do rapid prototyping. A feature extractor that can be used as a base of Faster R-CNN is NAS. This model is more accurate but a lot slower too. When the requirements change and speed is less important, this extractor might be a better solution. Especially in combination with the planned timeslot detections. This would be an interesting model to train and test with the already collected data and see what the differences are.

## 11.7 Multiple detections and anchor scaling

In the current version of the prototype, the maximum amount of detectable defective tiles per images is set to 300. This seems like a lot but in reality, one defective tile can take up a lot of these detections due to the fact that it detecting the same defective tile multiple times in the same place. Even though the settings are set to 300 objects at once, it still needs to be tested how many it can actually handle without problems.

Another setting that can be changed to possibly increase the performance of the model is the scales factor of the anchor generator. As explained in chapter 7.2, anchors are placed over the image and the sliding window of the RPN jumps from anchor to anchor. Right now, this scales factor is set to default which is [0.25, 0.5, 1.0, 2.0]. This is a good scales factor for most object detectors but due to the fact that the squares are really small, they can fall in between two anchor points and not be detected with high certainty. When decreasing this scales factor, the anchors get closer to each other and it is likely that the performance increases. It is recommended to decrease this scale factor and investigate of the performance of the system increases.

## 12 Ethical considerations

This ethical reflection focuses on the newly developed computer vision damage detection system for Hecla. This to find design flaws that may cause harm to the users and bystanders. The aim of the new product is to reduce workload in process with the use of computer vision. To find the possible ethical flaws in the current design, parts of the Ethics in Tech practice toolkit written by S. Vallor, R. McKenna and D. McKenna will be used. This toolkit consists out seven tools that lead to ethical thinking. Some of these tools are used for this ethical reflection.

### 12.1 Ethical risks

When designing this product and service the intention is to satisfy most of the stakeholders in order to sell your product. To do this optimally, it is necessary to minimize the ethical risks for all of these stakeholders. First, the primary stakeholders need to be defined. In this case, there are two primary stakeholders. The first stakeholder is personnel of Hecla which can be split up in the direction and the people that will be working with the software. The second group contains all the customers, the ones that will benefit from the new service of Hecla. Next to the primary stakeholders, there are secondary stakeholders. These people are not directly influenced by the product but in some cases, they can get in contact with the product or its consequences. Really important secondary stakeholders are the commuters that drive by the billboards. They do not have anything to do with the system but they will be picked up by the cameras.

These stakeholders have different ethical concerns due to the nature of their function and way of interacting with the system. This results in varying severity concerning these ethical issues of the product. Starting off with a well-known ethical issue: privacy. In this case, the system is working with video streams that will be checked by an artificially trained agent. This agent might be capable of saving data that can be misused in the future. The video streams are not fully aimed at the billboards and can capture data that contains personal information like license plates and faces. This data can be valuable to third parties because it can help them in their business. Due to the fact that the people that are captured in this data did not consent, it violates their privacy. Next to this ethical issue, there is an issue for the employees that are currently monitoring the billboards. With this new system, their function might be endangered resulting in an uncertain future for their career at Hecla. This might be an unethical solution it might affect the employees their dignity and autonomy. Therefore these ethical issues need to be well considered before rolling out the system. As said before, these ethical issues do not apply to all the stakeholders but they do need to be taken into account.

### 12.2 Similar case

It is useful to have a look at a similar case that had ethical problems. This way, the knowledge gathered after such an ethical disaster can be used to build preventions in Hecla's design. One similar case happened in the United States. A company that builds license plate scanners had a data leak[29].

In total 65.000 files from many border commuters were stolen. According to the company, the data is now circulating on the dark web.

With the cameras used for damage detection, this could be a similar scenario. When the system is hacked, a lot of sensitive data can be leaked to the general public. The article says that the information probably leaked from a personal computer. This tells us that people had access to this sensitive information and that it was probably not secure enough. From this can be learned that it is probably smart to keep the system on just one of Hecla's computers that is reserved for the system. This way, people cannot accidentally click on phishing links for example. Another thing that can be done to manage these issues better, is to encrypt the sensitive information. Then, even if hackers breach into Hecla's systems, the data is encrypted and cannot be used.

### 12.3 Ethical benefits

When writing about all the ethical issues it, it feels like restricting yourself and putting more money and effort in a product than necessary. In a lot of cases, people tend to forget that new develop products, in the end, have the goal to improve the company, help its employees and make the customer happy. Therefore it is also useful to write down the ethical benefits of the new product. In the case of the new computer vision damage detections system, there are three main ethical benefits that come from this innovation. The first one is that the customers of the maintenance service are now more satisfied than before. They have on their behalf more happy customers because the content on the billboards is showing correctly. This leads to overall higher happiness among all the stakeholders. The second one is this new innovation leads to a reduced workload for the people that are currently monitoring the billboards. They only have to check the billboards manually if systems send out a notification. A reduced workload is better for overall happiness and therefore an ethical benefit. The last ethical benefit is less obvious. Being ethical with your products makes a company attractive for new clients which in the end enlarges the market share of the company. Altogether, being ethical can cost extra money but there are also a lot of benefits that will, in the end, help the company to develop.

### 12.4 Misuse of the product

Developing a product also includes thinking about people with bad intentions using your product. Protecting others about criminals using your product is something Hecla has to think about. For computer vision detection product, there are a lot of privacy issues concerning the cameras. The conventional cameras can be used for all sorts of spying but when the camera will be hooked up to artificial agents a lot more issues will appear.

The first issue was already mentioned in tool 1. When the system is not operating correctly it can pass sensitive information about commuters to people who do not have the rights to get this information. This can be scaled up to a point where everything the camera is filming will be

deliberately passed on to third parties. This can include for example license plates and if the camera improves maybe even faces. When faces get recognized at a high level all new sorts of problems occur like security problems. To prevent this from happening, the system needs to blur out these aspects from the image before third parties can access the data.

Another problem which is not obvious is Copyright infringement. The billboards are showing content to the people on the roads which might be copyrighted. In this case, the software is officially not allowed to save the data. This needs to be negotiated with the content creators but when people with bad intentions get their hands on this data they might sell it to third parties. Therefore, all the data that is saved by the system needs to be secured in a proper way. Another technique could be just throwing away the data immediately in order to prevent this from happening.

## 12.5 Iterating the process

Ethics are often seen as guidelines for being right or being wrong. The terms right and wrong are not defined in depth and can differ over time due to a changing society. This is something that needs to be taken into account when designing a product and producing a product. The computer damage detection system might not have any ethical issues at this point, but in the future, the thoughts on certain topics might change. Therefore it is important to keep iterating the design process. In the this case, the software needs to be reflected on. This in order to find out if the product is still compliant to the ethical thoughts of the society and Hecla. All the seven steps should then be repeated to get an overview of the system at that current moment in time. A recommendation would be to do this once every year on a small scale and once every five years a completely new review of the product. Keeping these things in mind, the software will stay as ethical as possible resulting in the earlier mentioned ethical benefits.

## 13 References

- [1] S. Tripathi, G. Dane, B. Kang, V. Bhaskaran, and T. Nguyen, "LCDet: Low-Complexity Fully-Convolutional Neural Networks for Object Detection in Embedded Systems," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 411–420.
- [2] S. Bahrapour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative Study of Deep Learning Software Frameworks," 2016.
- [3] A. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, "A comparative study of open source deep learning frameworks," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018, pp. 72–77.
- [4] R. Phadnis, J. Mishra, and S. Bendale, "Objects Talk - Object Detection and Pattern Tracking Using TensorFlow," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 1216–1219.
- [5] Sayantini, "Keras vs TensorFlow vs PyTorch | Deep Learning Frameworks | Edureka," 2018. [Online]. Available: <https://www.edureka.co/blog/keras-vs-tensorflow-vs-pytorch/>. [Accessed: 01-Apr-2019].
- [6] H. Bierma, "Design of Data Acquisition Solution for Prognostics & Health Management in LED Screens," p. 46, 2018.
- [7] J. H. Sara Robinson, Aakanksha Chowdhery, "Training and serving a realtime mobile object detector in 30 minutes with Cloud TPUs," 2018. [Online]. Available: <https://medium.com/tensorflow/training-and-serving-a-realtime-mobile-object-detector-in-30-minutes-with-cloud-tpus-b78971cf1193>. [Accessed: 01-Apr-2019].
- [8] J. Talukdar, S. Gupta, P. S. Rajpura, and R. S. Hegde, "Transfer Learning for Object Detection using State-of-the-Art Deep Neural Networks," in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2018, pp. 78–83.
- [9] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN Implementation Method for Multi-Fruit Detection Using Tensorflow Platform," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2018, pp. 337–340.
- [10] G. Poonam, B. N. Shashank, and A. G. Rao, "Development of framework for detecting smoking scene in video clips," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 1, pp. 22–26, 2019.



- [11] D. P. Sudharshan and S. Raj, "Object recognition in images using convolutional neural network," in *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, 2018, pp. 718–722.
- [12] D. Wilson, F. Yan, K. Sinha, and H. He, "Robotic Understanding of Scene Contents and Spatial Constraints," Springer, Cham, 2018, pp. 93–102.
- [13] M. Nakib, R. T. Khan, M. S. Hasan, and J. Uddin, "Crime Scene Prediction by Detecting Threatening Objects Using Convolutional Neural Network," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, 2018, pp. 1–4.
- [14] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, "A Deep Learning Approach for Road Damage Detection from Smartphone Images," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5201–5204.
- [15] F. Al-Azzo, A. M. Taqi, and M. Milanova, "Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API," 2018.
- [16] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM Trans. Graph.*, vol. 27, no. 5, p. 1, 2008.
- [17] B. Hultgren and D. Hertel, "Low light performance of digital cameras," *Image Qual. Syst. Perform. VI*, vol. 7242, no. May, p. 724214, 2009.
- [18] Hillary Grigonis, "JPEG vs PNG: Which Compressed Image Format Is Better? | Digital Trends." [Online]. Available: <https://www.digitaltrends.com/photography/jpeg-vs-png/>. [Accessed: 27-May-2019].
- [19] Armaan Merchant, "Neural Networks Explained – Data Driven Investor – Medium," 2018. [Online]. Available: <https://medium.com/datadriveninvestor/neural-networks-explained-6e21c70d7818>. [Accessed: 18-Jun-2019].
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." pp. 91–99, 2015.
- [21] B. Liu, W. Zhao, and Q. Sun, "Study of object detection based on Faster R-CNN," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 6233–6236.
- [22] G. Wang and X. Ma, "Traffic Police Gesture Recognition using RGB-D and Faster R-CNN," in *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2018, pp. 78–81.
- [23] Saagie, "Object Detection - Saagie," 2017. [Online]. Available:

- <https://www.saagie.com/blog/object-detection-part2/>. [Accessed: 17-Jun-2019].
- [24] Z. Yang, F.-L. Du, Y. Xia, C.-H. Zheng, and J. Zhang, “Automatic License Plate Recognition Based on Faster R-CNN Algorithm,” Springer, Cham, 2018, pp. 319–326.
- [25] E. Juras, “TensorFlow Object Detection API Tutorial Train Multiple Objects Windows 10,” 2018.
- [26] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. 2017.
- [27] Unknown, “What is the maximum length of a text message? TextMagic.” [Online]. Available: <https://support.textmagic.com/faq/maximum-length-of-a-text-message/>. [Accessed: 14-Jun-2019].
- [28] Support Whatsapp, “WhatsApp FAQ - Is it free to send messages over WhatsApp?” [Online]. Available: <https://faq.whatsapp.com/en/android/20965922/>. [Accessed: 14-Jun-2019].
- [29] J. D. E., “Hackers breach US license plate scanning,” 2019.

# 14 Appendices

## 14.1 Appendix A

*Synthetic data*



*The indoor setup*



*Example images screens*



## 14.2 Appendix B

Example data from .XML files (left) and CSV files (right)

<pre>&lt;annotation&gt;   &lt;folder&gt;all_images&lt;/folder&gt;   &lt;filename&gt;images2.jpg&lt;/filename&gt;   &lt;path&gt;C:\Users\Niek Boersen\*&lt;/path&gt;   &lt;source&gt;     &lt;database&gt;Unknown&lt;/database&gt;   &lt;/source&gt;   &lt;size&gt;     &lt;width&gt;960&lt;/width&gt;     &lt;height&gt;540&lt;/height&gt;     &lt;depth&gt;3&lt;/depth&gt;   &lt;/size&gt;   &lt;segmented&gt;0&lt;/segmented&gt;   &lt;object&gt;     &lt;name&gt;broken&lt;/name&gt;     &lt;pose&gt;Unspecified&lt;/pose&gt;     &lt;truncated&gt;0&lt;/truncated&gt;     &lt;difficult&gt;0&lt;/difficult&gt;     &lt;bndbox&gt;       &lt;xmin&gt;440&lt;/xmin&gt;       &lt;ymin&gt;32&lt;/ymin&gt;       &lt;xmax&gt;459&lt;/xmax&gt;       &lt;ymax&gt;55&lt;/ymax&gt;     &lt;/bndbox&gt;   &lt;/object&gt; &lt;/annotation&gt;</pre>	<pre>filename,width,height,class,xmin,ymin,xmax,ymax images100.jpg,960,540,broken ,799,99,811,117 images1000.jpg,960,540,broken ,777,176,788,191 images1001.jpg,960,540,broken ,656,121,673,141 images1002.jpg,960,540,broken ,512,211,529,232 images1003.jpg,960,540,broken ,446,29,468,49 images1004.jpg,960,540,broken ,767,216,777,227 images1005.jpg,960,540,broken ,405,154,424,173 images1006.jpg,960,540,broken ,190,69,215,93 images1007.jpg,960,540,broken ,531,281,551,301 images1008.jpg,960,540,broken ,551,239,571,262 images1009.jpg,960,540,broken ,533,280,549,300 images101.jpg,960,540,broken ,574,366,595,387 images1010.jpg,960,540,broken ,767,287,780,304 images1011.jpg,960,540,broken ,767,287,779,304 images1012.jpg,960,540,broken ,357,259,376,281 images1013.jpg,960,540,broken ,357,259,377,280 images1014.jpg,960,540,broken ,523,299,539,316 images1015.jpg,960,540,broken ,797,185,810,201 images1016.jpg,960,540,broken ,574,207,591,224 images1017.jpg,960,540,broken ,574,207,589,225</pre>
---	---

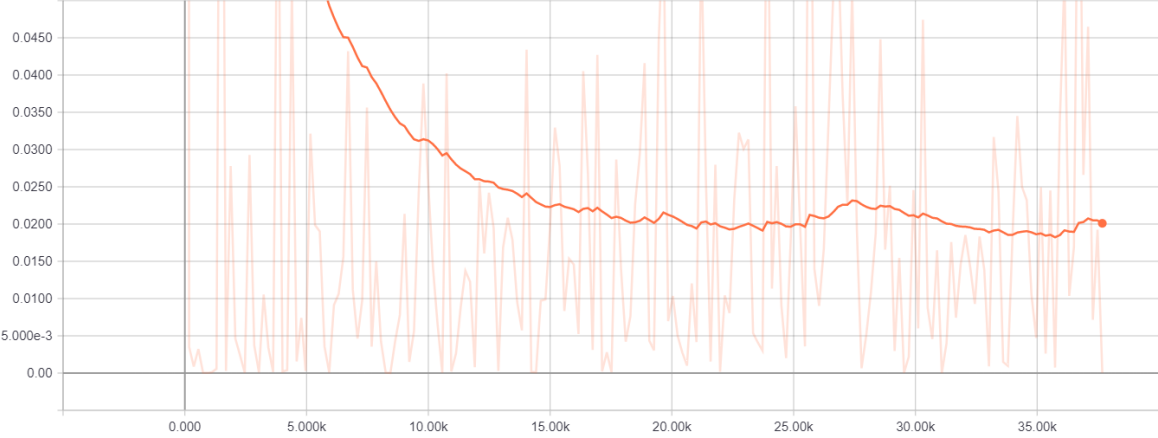
Collected images per group (ID)

ID	distance	height	light	angle	ID_begin	ID_end	total	#test	#train
1	close	high	on	30	2	119	117	23,4	93,6
2	far	high	on	0	120	214	94	18,8	75,2
3	far	high	on	30	215	305	90	18	72
4	close	high	off	0	306	423	117	23,4	93,6
5	close	high	off	30	424	521	97	19,4	77,6
6	far	high	off	0	522	623	101	20,2	80,8
7	close	low	on	0	624	715	91	18,2	72,8
8	close	low	on	30	716	811	95	19	76
9	far	low	on	0	812	896	84	16,8	67,2
10	close	low	off	0	897	994	97	19,4	77,6
11	close	low	off	30	995	1075	80	16	64
12	far	low	off	0	1076	1160	84	16,8	67,2
13	close	low	on	0	xxx	xxx	xxx	xxx	xxx
14	close	high	on	0	xxx	xxx	xxx	xxx	xxx
15	far	high	on	0	1161	1271	110	22	88
16	far	low	on	0	1272	1385	113	22,6	90,4
17	far	low	off	0	1386	1477	91	18,2	72,8
18	far	high	off	0	1478	1576	98	19,6	78,4
19	close	high	off	0	1577	1682	105	21	84
20	close	low	off	0	1683	1790	107	21,4	85,6
21	close	low	off	30	1791	1889	98	19,6	78,4
22	close	high	off	30	1890	1981	91	18,2	72,8
23	far	high	off	30	1982	2087	105	21	84
24	far	low	off	30	2089	2194	105	21	84
25	far	low	on	30	2195	2273	78	15,6	62,4
26	far	high	on	30	2274	2365	91	18,2	72,8
27	close	high	on	30	2366	2445	79	15,8	63,2
28	close	low	on	30	2446	2558	112	22,4	89,6
29	close	low	on	60	2559	2654	95	19	76
30	close	high	on	60	2655	2750	95	19	76
31	far	high	on	60	2751	2856	105	21	84
32	close	high	on	0	2857	2957	100	20	80
33	close	low	on	0	2958	3042	84	16,8	67,2
34	close	low	on	30	3043	3112	69	13,8	55,2
35	close	high	on	30	3113	3191	78	15,6	62,4
36	far	high	on	30	3192	3244	52	10,4	41,6
37	far	high	on	60	3245	3303	58	11,6	46,4
38	far	low	on	60	3304	3353	49	9,8	39,2
39	close	low	on	60	3354	3392	38	7,6	30,4
40	close	high	on	60	3393	3437	44	8,8	35,2
41	far	high	on	60	3438	3479	41	8,2	32,8
42	far	high	off	60	3480	3523	43	8,6	34,4
43	close	high	off	60	3524	3558	34	6,8	27,2
44	close	heel low	on	0	3559	3606	47	9,4	37,6
45	far	heel low	on	0	3607	3663	56	11,2	44,8

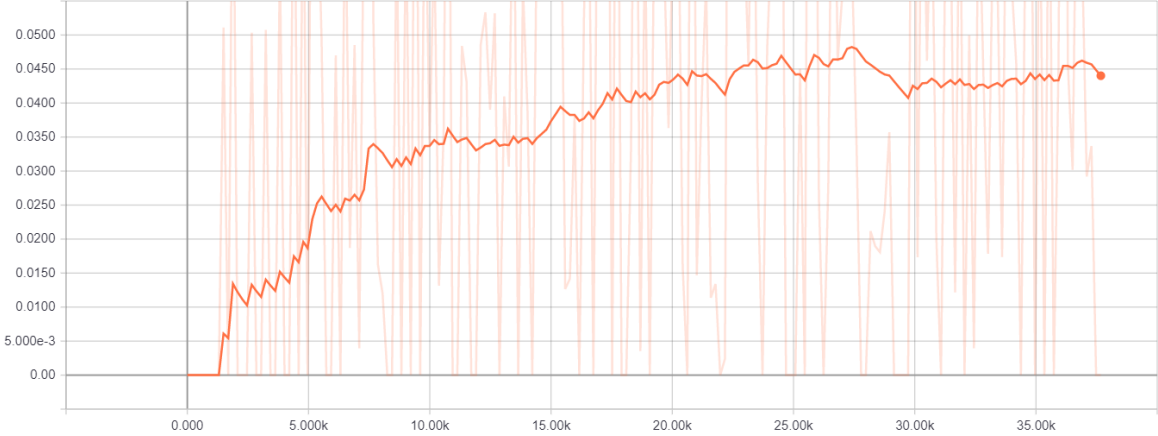
# 14.3 Appendix C

## Graphical representation of loss functions over time

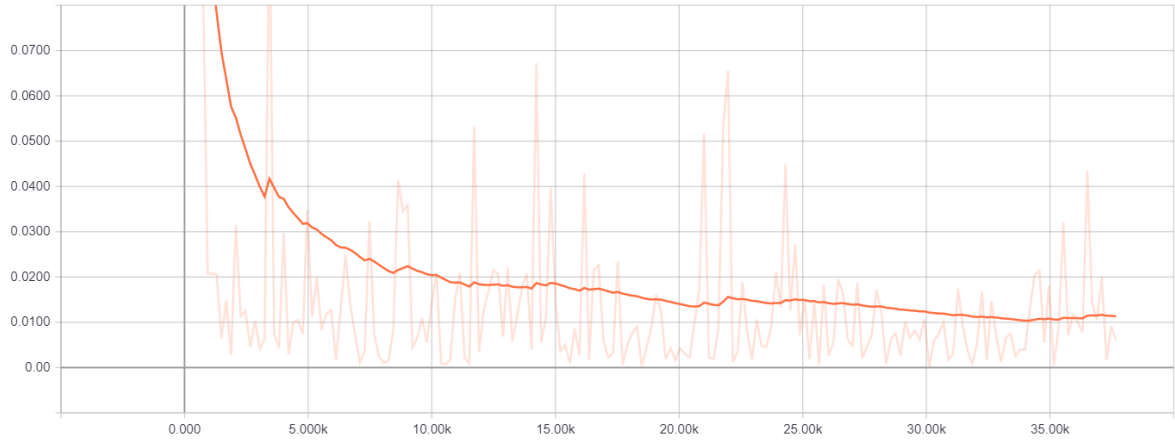
Loss/BoxClassifierLoss/classification\_loss  
tag: Losses/Loss/BoxClassifierLoss/classification\_loss



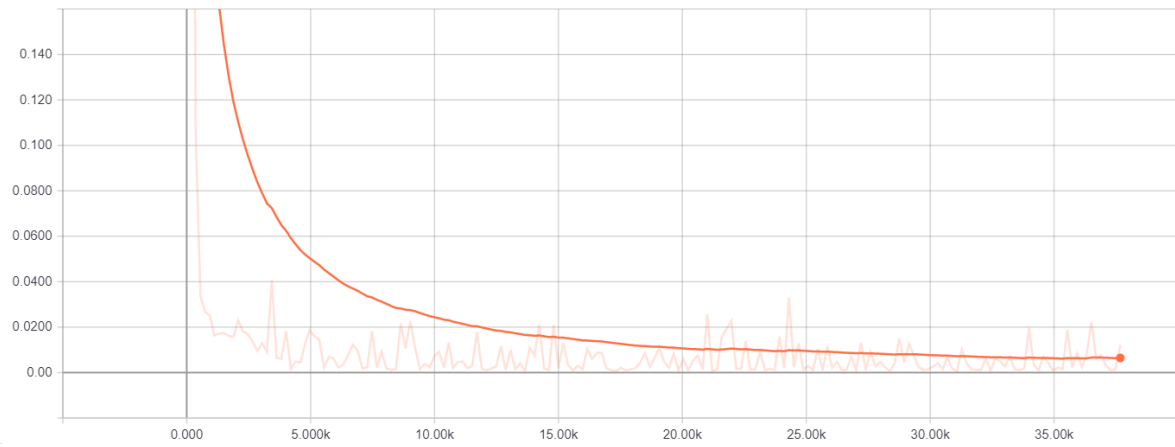
Loss/BoxClassifierLoss/localization\_loss  
tag: Losses/Loss/BoxClassifierLoss/localization\_loss



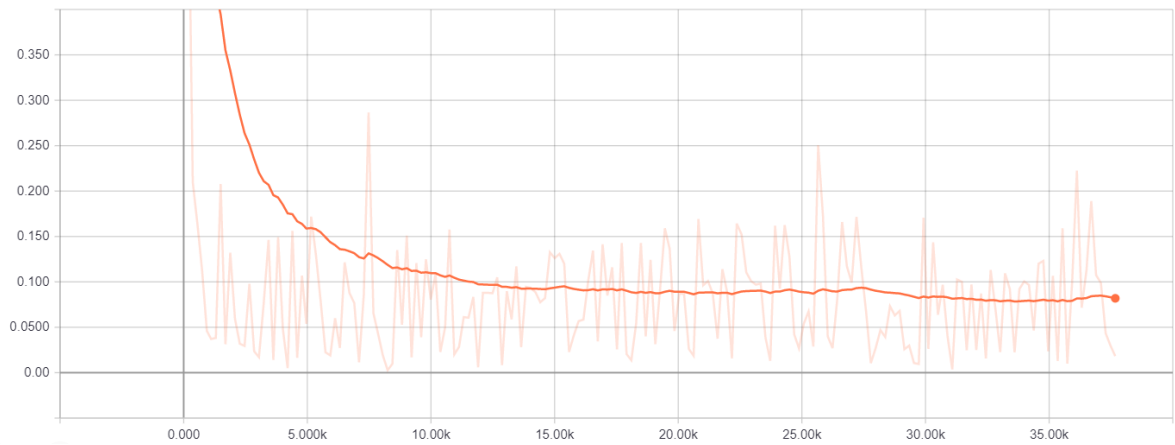
Loss/RPNLoss/localization\_loss  
tag: Losses/Loss/RPNLoss/localization\_loss



Loss/RPNLoss/objectness\_loss  
tag: Losses/Loss/RPNLoss/objectness\_loss



TotalLoss  
tag: Losses/TotalLoss





*Result image examples (both true positive results)*

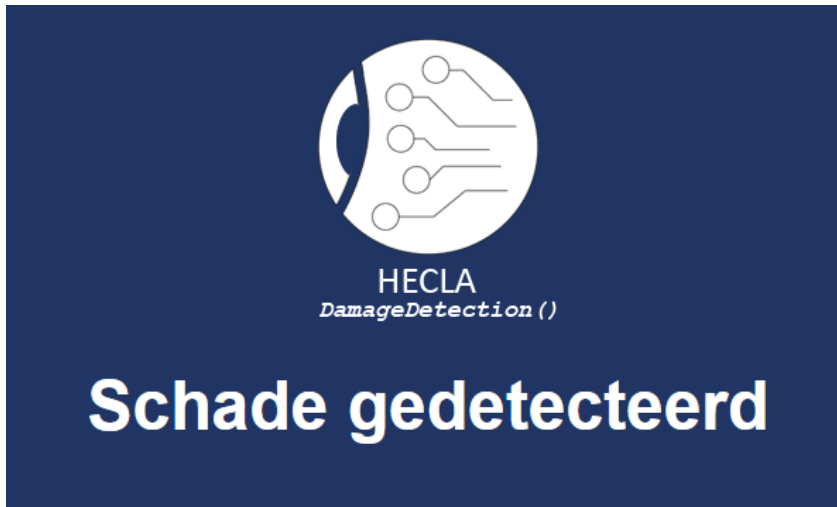


*Example of a false positive result*



## 14.4 Appendix D

*Example of an output email*



Goedemorgen Hecla,

Het systeem heeft een kapot scherm gedetecteerd. Het betreft het volgende scherm:

- RM065 Rotterdam Brienoordburg A16
- gedetecteerd op 06/14/2019 om 10:22:51
- Het systeem is zeker met: 78,2%

Deze email zal morgen weer opnieuw verstuurd worden als de defecte tegel morgen wederom gedetecteerd wordt. Wanneer dit een foutieve detectie is graag een email sturen naar: [tech@hecla.nl](mailto:tech@hecla.nl).

Klik op de onderstaande knop om de livestream te bekijken. Daarnaast is er een schermopname van het systeem is te vinden in de bijlage.

M.v.g. Niek

[Livestream](#)