NHL STENDEN
computer vision & data science

Detecting Defects in Sewage Pipes Using Deep Convolutional Neural Networks

NHL Stenden Centre of Expertise in Computer Vision & Data Science

Alpay Vodenicharov

Supervisors: Willem Dijkstra, Hossein Rahmani

Abstract— The functionality of sewer pipes is important in order to avoid social problems and the spread of diseases. For that purpose, the project tackles the research question "Is it feasible to detect defects in sewer pipes using deep-learning with convolutional neural networks". Taking a dataset of text-annotated videos and annotating them by hand, 1000 samples of infiltration defect were finalized. Splitting 60% train, 20% validation and 20% testing for experiments using two networks – U-Net and Mask-RCNN. After training the networks and performing validation, U-Net resulted in 0.85 IoU score, while Mask-RCNN 0.71. Hence it is concluded that detection of infiltration in sewer pipes is possible with the use of U-Net, however, more research needs to be done with a more reliable dataset and annotations in order for this to be applicable in the real world.

1 Introduction

The proper functionality of underground sewer pipes is vital for the civil infrastructure of cities. Modern sewer systems consist of a complex network of drainage pipes, which process waste water, ground water, etc. Sewer systems are prone to many types of defects, which lead to massive consequences for the society, such as exposure to sewage gas, spreading of diseases, flooding of surfaces, etc. To avoid the mentioned inconveniences, it is important to track and monitor the condition of the sewer systems. With modern technology, it is still a difficult problem to resolve due to humans being involved in the loop, i.e it is possible to inspect the sewer pipes using a remote-controllable robot that goes inside the sewers with a camera and streams live footage of the condition, but it is still required to review and annotate the defects manually. This approach is prone to errors and very time-consuming, considering the length and amount of the pipes in a city. This research paper will concern the defect detection in pictures of sewer pipes using deep learning by answering the following main question: **Is it feasible to detect defects in sewer pipes using deep-learning with Convolutional Neural Networks?**. Those defects come in different shapes, hence why this research will focus on neural networks with different outputs than the standard bounding boxes, such as rotated bounding boxes and multiple-point based polygons. The end-result will be a neural network that will serve as a proof of concept whether it is possible to achieve different types of detection outputs and measure their performance.

Currently, in terms of object detection, there are several state-of-the-art methods. Ronneberger et. al.[1], in their paper "U-Net: Convolutional Networks for Biomedical Image Segmentation" propose a method for image segmentation using biomedical images. The approached training strategy relies on the use of data augmentation to make more efficient use of the dataset. The architecture consists of a downsamping path to capture context and an upsampling path that enables precise localization. The unconventional specialty of this network is assumed to be extensible to other peculiar tasks such as defect detection.

Another paper by Kaiming He et. al.[2] propose Mask-RCNN. Mask-RCNN is a flexible and general method for instance segmentation. The approach taken detects objects in images while also generating segmentation masks for each detected instance. Mask-RCNN is an extension of Faster-RCNN[3] with the addition of a branch for mask predictions in parallel with the previous branch. Mask-RCNN has high generalizability, ranging from pose estimations, nuclei detection, human detection, etc. Experiments will prove whether the generalizability of this network can extend to more uncommon tasks such as defect detection in sewer pipes. Finally, it is possible to conduct a research on the main question using the mentioned state-of-the-art and other tooling.

Alpay Vodenicharov is a Computing Science student at the NHL Stenden University of Applied Sciences, E-mail: alpay.vodenicharov@student.nhlstenden.com.

Willem Dijkstra is a researcher at the NHL Stenden Centre of Expertise in Computer Vision & Data Science, E-mail: willem.dijkstra@nhlstenden.com.

Hossein Rahmani is a researcher at the NHL Stenden Centre of Expertise in Computer Vision & Data Science, E-mail: hossein.rahmani@nhlstenden.com.

2 Materials and Methods

This section will concern the software, datasets, hardware and methodologies of choice for the experiments and why it was decided to use them.

2.1 Dataset

The dataset contains 18 videos and 49 images recorded by a remote-controllable robot inspecting the sewer pipes. The length of the videos vary between 15 to 24 minutes, running at 25 bitrates per second with a resolution of 352x288. Extraction of additional frames that contain defects of interest from the videos will be performed to increase the size of the image samples, which will be the main resource for training the neural networks. No negative samples will be present. The samples contain one defect class - infiltration. A wet brown spot on the sewer pipe indicates the infiltration of water inside it from outside sources. The final dataset will contain 1000 samples that will be split 60% for training, 20% for validation and another 20% for testing.

The brown spots in the walls of the pipe represent infiltration and sweating. Figure 2 shows the outline of the infiltration.

2.2 Annotation

The final dataset of image samples will be manually annotated using the LabelMe[4] tool for instance segmentation. LabelMe allows for annotating with closed polygons, which can be used to draw pixel-wise segmentation masks. LabelMe also supports drawing bounding boxes. Finally, the video datasets contain text annotation, however, visual annotations are not present. This means that the visual annotations will be done manually while confirming the defects in the videos. The annotations for this task will not be done by a professional specialized in defects of sewer pipes, which could rise potential issues.

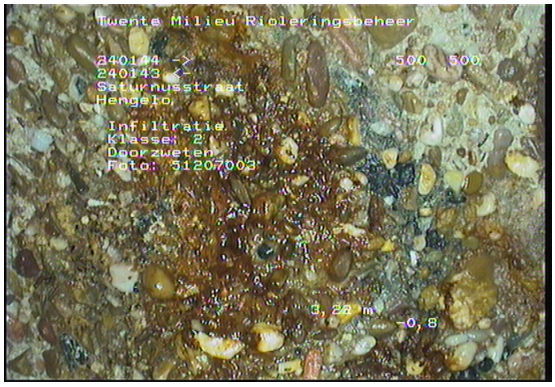

Fig. 1. Infiltration example

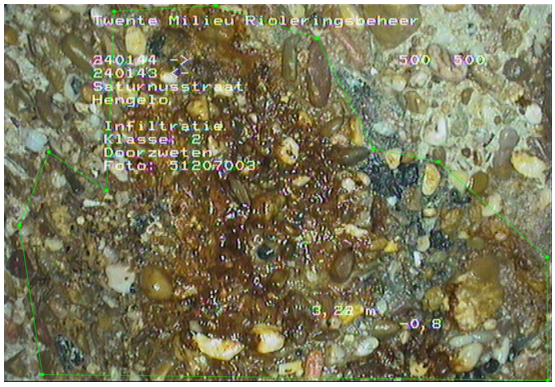

Fig. 2. Infiltration example

2.3 Instance Segmentation

Polygon-based detection, unlike bounding boxes, will cover only the area that is of interest with minimal clutter. The restrictive shape of bounding boxes leads to noise and unnecessary objects being placed inside Regions of Interests. Polygon shapes tackle this issue successfully. This will be achieved by using Instance Segmentation.

2.3.1 U-net[1]

The U-Net algorithm specializes in achieving a good performance with limited amount of samples as stated in their paper[1], such as in our case. It relies on data augmentation to generate a more differential dataset to avoid overfitting. Furthermore, since it specializes in medical imaging, it is proven that it can work in unconventional environments, i.e in this case inside a sewer pipe. Lastly, it is a fast network that does not require a long computational time. The U-net architecture is shown in Figure 3: The architecture has two sides - a downsampling (left) and an upsampling (right). The contracting side consists of a typical CNN architecture - two 3x3 convolutions followed by a ReLU and 2x2 max-pooling operations. The expansive side upsamples the feature map followed by a 2x2 convolution and two 3x3 convolutions, each of them with a ReLU. Finally, a 1x1 layer to map the feature vectors to the number of classes.

2.3.2 Mask-RCNN [2]

Mask-RCNN[8] is an extension of Fast-RCNN[3]. Mask-RCNN performs instance segmentation and object detection with horizontal bounding boxes. Furthermore it is easy to generalize the network for different goals due to the flexibility of its use case. Some examples include nuclei detection, human-pose estimation and defects

[1]https://github.com/zhixuhao/unet

[2]https://github.com/matterport/Mask$_R$CNN

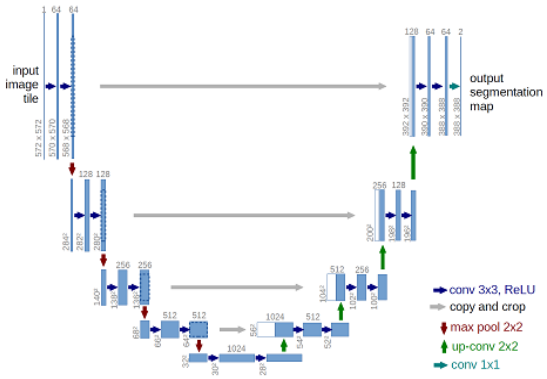

Fig. 3. U-net architecture[1]

detection, which is the main problem that will be covered in this research. Lastly, Mask-RCNN is a state-of-the-art that outperforms most other successful networks in different tasks as mentioned in their paper[2], which makes it a suitable candidate for our experiments. The network consists of the architecture depicted in Figure 4. Throughout the first stage, the RPN is responsible for

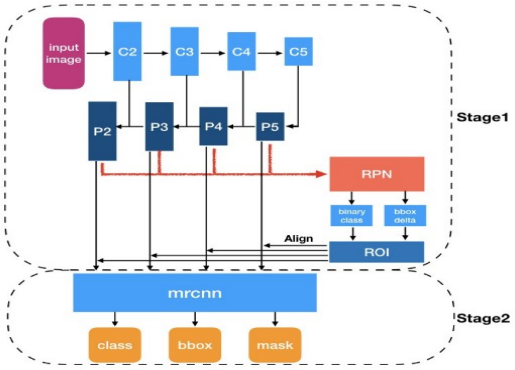

Fig. 4. Mask-RCNN Architecture[8]

scanning the feature maps and proposing regions that may contain the object. Furthermore anchors are used to bind the features in their raw image location. In the second stage, another NN processes the proposed regions by scanning the areas and generating object classes, masks and bounding boxes.

2.4 Software and Hardware

This subsection introduces the software and hardware used in the research project.

2.4.1 Annotation converter

Due to the reason that most networks use different annotation formats, a converter script has been created to transform 1 type of annotation into others. This methodology results in the swift conversion of annotations to save time. The annotation format used is COCO[7] for instance segmentation , annotated via LabelMe[4]. The resulting COCO format annotation is then converted into a custom base format that fits into the Mask-RCNN implementation. For U-Net, the images are converted into pixel-masks by using the coordinates from the annotation files.

2.4.2 U-Net display script

The final output logits of U-Net are converted to a resulting mask of the defect. This is done by taking the coordinates of the defect by a threshold value of less than 130 from the logits and placing a segmentation mask on the same coordinates on the real image that

was inferred. This results in a comprehensible image with a segmentation mask. This is an additional step that was required due to the reason that the implementation of U-Net not outputting the aforementioned result.

2.4.3 Hardware

The following hardware was used throughout the research project in Table 1:

Table 1. Hardware used to train the neural networks.

Hardware	Specification
GPU	NVIDIA RTX2070
GPU memory	8GB
RAM	14.75GB
CPU	12 Cores
OS	DEbian 10.1
CUDA versions	10.1

2.5 Data Augmentation

Data augmentation will be performed on the final dataset in order to increase the sample size and give a wide variety. This approach will reduce the likeliness of overfitting during training. This also tackles the issue that the training samples are not abundant. The augmentation techniques to be performed include random amounts of horizontal and vertical flipping of 180, random number of degree rotations and zooming between a minimum factor of 1.1 and maximum factor of 1.5.

2.6 Metrics

It is crucial to select appropriate metrices for the comparison of results to be precise and fair. In the case of this research, Intersection over Union (IoU) has been selected, since it provides a ratio of correctness of the output predictions and the ground-truth labels[6]. Due to varying factors, a complete match between the ground-truth and predictions is nearly impossible, as stated by Adrian Rosebrock [6]. IoU is an evaluation metric that rewards predicted areas for heavily overlapping with the ground-truth. An example of IoU performances by [6] is given in Figure 5. The traditional loss value

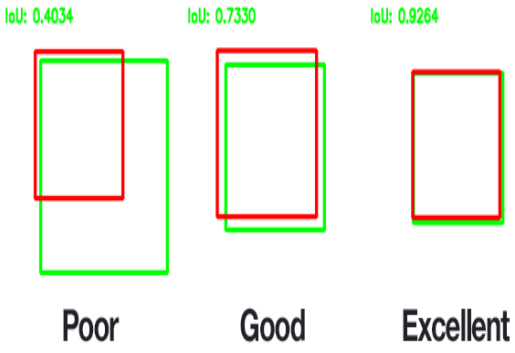

Fig. 5. IoU performance by Adrian Rosebrock [6]

from the validation dataset will be used to determine the performance of the training. The optimization to be used will be Adam[8].

Lastly, a confusion matrix will be drawn to compute the false-positives(green), false-negatives(blue), true-positives(yellow) and true-negatives(purple) of the outputs for Mask-RCNN due to the large amount of inaccuracies. Vertically the true labels are represented and horizontally - the predicted labels. Samples from the testing set were tested in order to achieve a fair computation of the matrix.

3 Experiments

This Chapter elaborates the experiments to be prepared and executed to answer the research question "is it feasible to detect defects in sewer pipes using deep-learning with Convolutional Neural Networks".

3.1 Pre-processing

Several pre-processing methods will be tested and compared throughout the experiments phase.

3.1.1 On-the-fly Augmentation

Augmentation will be performed before the images are being processed by the network. As indicated in Chapter 2 Materials and Methods, this pre-processing will increase the sample size of the datasets, give more variety and reduce the potential of overfitting.

3.2 Experiment: U-net

U-Net, proposed by Olaf Ronneberger et. al. [1] is a convolutional neural network specializing in medical imagery. Hence it is expected that U-net will produce favorable results once re-trained with new images due to the reason that it detects stains in images, which the defective spots could be related, whereas Mask-RCNN specifically detects objects. Unlike Mask-RCNN, U-Net requires the masked images to be provided by the researcher, instead of automatically creating them on-the-fly based on annotations. The masks for the datasets will be manually created with a Python script that draws the masks based on the annotation files, and store the results in the appropriate folder. U-net will be trained for 150 epochs, which is a good estimation of when it will stop improving a lot, and have a learning momentum of 0.99, in order for the majority of previously seen training samples to determine the update in the new optimization step. Throughout the training, the best model with the lowest loss on the validation set will be saved and finally tested. For this implementation of U-net, the default hyperparameters have been used.

3.3 Experiment: Mask-RCNN

The implementation of Mask-RCNN by Kaiming He et. al[8] will be trained on the pair of colour and grayscale datasets. In order to train Mask-RCNN, the datasets must contain annotations of segmentation masks. The segmentation will be drawn using point-based polygons that cover the defect area. The network will use the Resnet-50 backbone and train all the layers from the beginning. The network will be trained for an epoch of 150, which is a good estimation of when it will stop improving a lot, while the steps-per-epoch parameter will be equivalent of the number of sample size times the batch size. Further hyperparameters and requirements to run Mask-RCNN can be found in Appendices.

3.3.1 Grayscale conversion

As proposed by Joshua Myrans et. al.[5], conversion of the images from color to grayscale could prevent blunders caused by the illumination in the pictures. According to their experiments, this preprocessing reduced the noise and eliminated the dependency on the variety of illumination in the images. However, the infiltration defect is distinguished by its color, which may create complications with grayscale images, hence the experiment to be performed will also include colour pictures, resulting in two datasets. This experiment will determine more authentic results for comparison.

3.4 Video Inference

To test the usability of the models, a video inference will be done on test data unknown to the models. Frames of a video will be extracted, inferred, then assembled back together into a video. The inference will draw appropriate segmentation masks to the detected areas. The measurement and comparison for this experiment will be qualitative - based on visual judgment of the results.

3.5 Comparison

Finally once the experiments have been performed, a final experiment to compare the results of the best produced networks will be done. This will conclude the best suitable network for use. Section 2.6 delves more into the calculation of performance.

4 Results

The final results from the experiments designed and performed from the experiments Chapter will be elaborated in detail within this Chapter. In order to achieve equal and comparable results, all networks were trained on epochs of 150 due to an estimation that it will stop improving by a large margin then.

4.1 U-net

U-net has shown to have the highest IoU score out of both the networks compared for detecting infiltration type defects. The performed video-inference also clearly shows higher accuracy and much fewer false-positives. The U-Net algorithm's inference outputs a segmentation mask of the input and calculates the IoU by comparing the output and ground-truth labels. The following graph represents the improvement of IoU during a period of 150 epochs. The final achieved IoU is a score of 0.85 on the validation set in Figure 7.

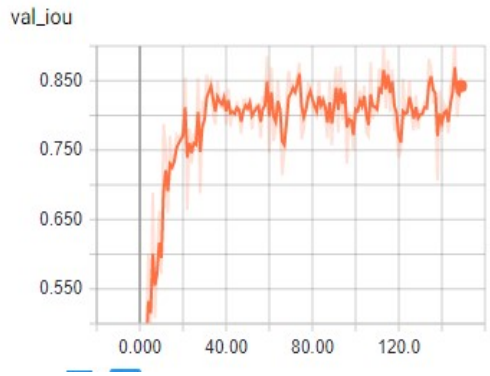

Fig. 6. Number of epochs trained in X-axis and IoU of U-net Y-axis

Further graphs on additional metric values, i.e loss, can be found in Appendices. An example image of U-net detection is below in Figure 8. U-net detection is a success and the detection performance is very good in regard to recognizing defective spots.

Fig. 7. U-Net inference result from test data

4.2 Mask-RCNN

Mask-RCNN was concluded with a lower IoU score of 0.71 than U-net on the validation set and therefore comes 2nd. Video-inference shows that Mask-RCNN produces far more false-positives and misses many defective spots during testing. An example image of the performance from Mask-RCNN is in Figure 9.

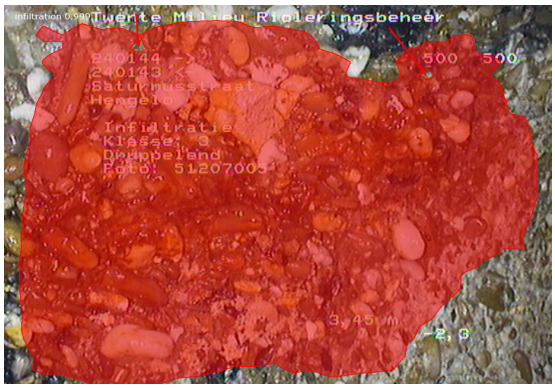

Fig. 8. Mask-RCNN example inference result from testing dataset

However, in zoomed-out scenes such as Figure 10, Mask-RCNN seems to produce less satisfying results.

Fig. 9. Mask-RCNN example inference result from testing dataset in zoomed-out view

Another less satisfying result of Mask-RCNN in Figure 11, in which half of the defect is missed:

Fig. 10. Substandard Mask-RCNN example inference result from testing dataset

4.3 Grayscale Mask-RCNN

Further experimentations with grayscale versions of the same images yielded in no better results, with a mean IoU of 0.68 on the validation set. The following example is an inference of a grayscale sample:

The detection performance in video-inference of both methods is

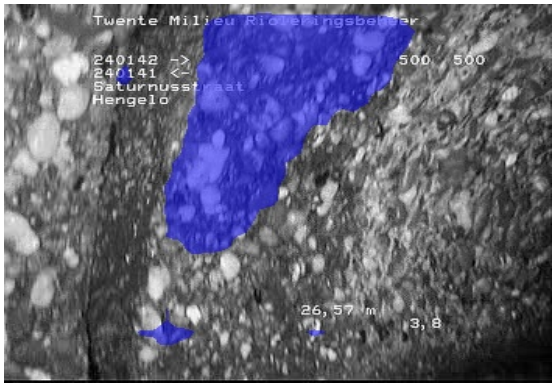

Fig. 11. Mask-RCNN grayscale example inference

significantly poor and critical spots are missed, while false-positives are also present.

4.4 Confusion Matrix

A confusion matrix was drawn with the intent to visualize the extent of false-positives, false-negatives, true-positives and true-negatives. The confusion matrix concluded 14 false-positives, 19 false-negatives and 35 true-positives. The results stem from inference on a small subset

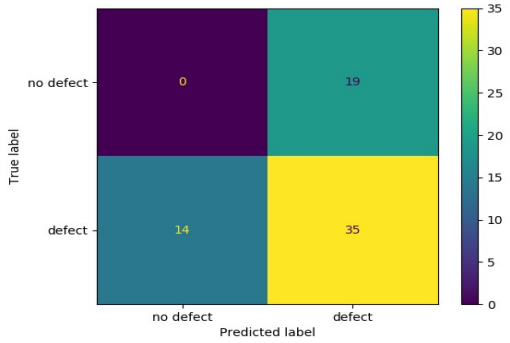

Fig. 12. Confusion matrix

of the validation dataset. To evaluate what was a correct detection, a threshold value of 0.65 was used. This means that detections with a confidence score of 0.65 or above were considered to be correct.

4.5 Comparison

The networks used were compared using the IoU metric, since it is assumed that it suits segmentation tasks due to the reason that it compares the ratio of correctness between the prediction and the ground-truth area pixel-wise. The achieved results from the experiments with the mean IoU value can be found in the following Table 2:

Table 2. Final mean IoU values of U-Net and Mask-RCNN.

Network	Mean IoU
U-Net	**0.85**
Mask-RCNN	0.71

The information on the table presents that U-Net was finalized with a mean IoU of **0.85**, while Mask-RCNN achieved 0.71. This makes a difference of 0.14 mean IoU score, which puts U-Net on top.

5 Discussion and Conclusion

After an extensive research has been conducted on the problem description of ”defect detection in sewer pipes using deep learning”, a conclusion has been reached. The focus of the research was mainly to detect infiltration type defects. The first experiment using Mask-RCNN had the goal to give an overview whether it is possible to use instance segmentation to detect defective spots in sewer pipes. With a dataset sample size of 1000 images and their annotations, the network was trained on 150 epochs. The performance of Mask-RCNN was not satisfying with an IoU of 0.71, few missed defective spots and false-positives. Hence Mask-RCNN is concluded to be not feasible for use in our goal. Further research with U-net has resulted in a higher IoU score and highly accurate predictions based on visual judgment on video inference. The video inference from U-net has also shown feasible results. With display adjustment scripts on the resulting outputs, U-net shows reasonable results on infiltration detection with an IoU of 0.85. Both networks, however, have difficulties detecting infiltration in scenes where the camera is not zoomed into the defective spot. It is assumed that the reason Mask-RCNN is outperformed by U-Net is due to the difference of goal orientations between both networks. Mask-RCNN, with ResNet backbones, has an orientation towards more general tasks, i.e detection of cars, humans, general every-day objects, etc. while U-Net is specialized for more unconventional cases, as it was originally developed for medical images and furthermore it does not require a large dataset to train. The detection of infiltration in sewer pipes is one such case, which allowed for U-Net to show its strengths. A further confusion matrix has been drawn on a sample from the validation dataset that showed how many false-positives, true-positives, false-negatives and true-negatives the network would produce, which concluded that Mask-RCNN detects certain areas as defects, when in fact the area is does not contain any defects. Moreover, certain actual defects, such as in the zoomed-out example in Fig. 9, are missed by the network. With 14 false-positives, 19 false-negatives, and 35 true-positives, it is concluded that Mask-RCNN has a higher error percentage, which can be suspected to be due to the lack of more data.

6 Future Work

This Chapter introduces certain shortcomings, how they can be tackled, and potential research material for upcoming future researchers.

6.1 Mask-RCNN

Mask-RCNN is currently outperformed by U-Net and produces false-positives or misses spots. Future work for this network could include better hyperparameter tuning, use of a smaller backbone, i.e resnet50, and enlarging the training and validation datasets and creating higher quality annotations.

6.2 U-Net

Improvements to U-net could include to change the network to directly output masks on the real RGB images instead of using a script to adjust the display and draw masks on the coordinates taken from the black and white images, however, those improvements are dependent on the original U-Net repository. Further tuning of the light hue threshold could also yield better results. Finally, experimenting with different architectures for U-Net could also provide better results.

6.3 Other detection classes

The current models detect only infiltration type defects. For a real-world application of the research, it is vital to add more classes of defects, such as physical damage to the pipe surface, for detection. With the addition of new classes, the model will be complete and usable in a real-world scenario.

REFERENCES

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. (18 May 2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. Retrieved from https://arxiv.org/abs/1505.04597

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. (24 Jan 2018) Mask-RCNN. Retrieved from https://arxiv.org/pdf/1703.06870.pdf

[3] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun (4 June 2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved from https://arxiv.org/pdf/1506.01497.pdf

[4] GitHub (2020) LabelMe. Retrieved from https://github.com/wkentaro/labelme

[5] Joshua Myrans, Zoran Kapelan, Richard Everson (2018) Automatic identification of sewer fault types using CCTV footage. Retrieved from https://easychair.org/publications/paper/ZQH3

[6] Adrian Rosebrock (Nov 7 2016) IoU for object detection. Retrieved from https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár (May 1 2014) Microsoft COCO: Common Objects in Context. Retrieved from https://arxiv.org/abs/1405.0312

[8] Diederik P. Kingma, Jimmy Ba (Dec 22 2014) Adam: A Method for Stochastic Optimization. Retrieved from https://arxiv.org/abs/1412.6980

7 APPENDICES

A MASK-RCNN HYPERPARAMETERS

Table 3. Mask-RCNN hyperparameters.

Parameter	Value
Learning momentum	0.9
RPN_NMS_THRESHOLD	0.9
USE_MINI_MASK	False
TOP_DOWN_PYRAMID_SIZE	256
POOL_SIZE	7
MASK_POOL_SIZE	14

B MASK-RCNN REQUIREMENTS

Table 4. Mask-RCNN requirements

Requirement
Imgaug library
Pycocotools
Keras 2.0.8
Tensorflow 1.4
Python 3.4

C U-NET REQUIREMENTS

Table 5. U-net requirements.

Requirements
Tensorflow
Keras 1.0-1.5
Python 2.7-3.5

D U-NET ADDITIONAL GRAPHS

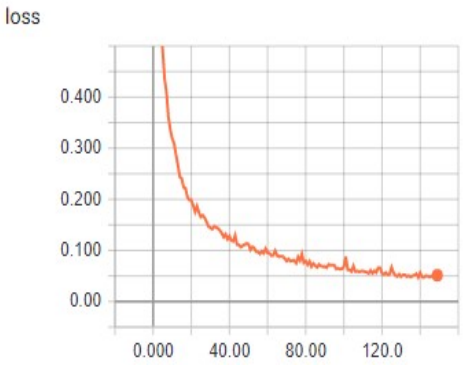

Fig. 13. Validation Loss of U-net

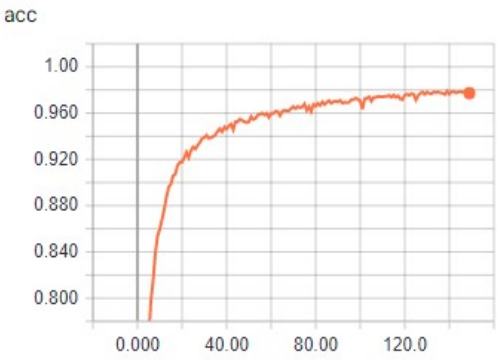

Fig. 14. Validation Accuracy of U-net

E MASK-RCNN ADDITIONAL GRAPHS

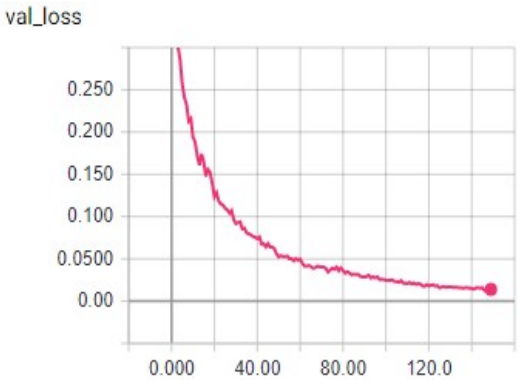

Fig. 15. Validation loss of Mask-RCNN

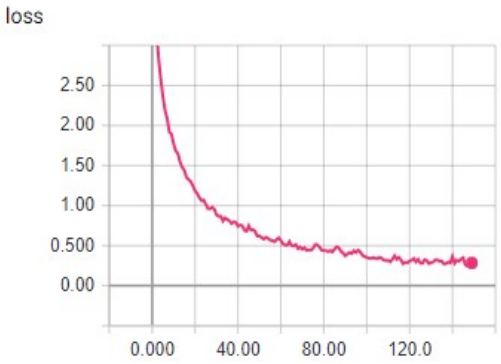

Fig. 16. Training Loss of Mask-RCNN