Vebjørn Bjørlo-Larsen
Endre Slettemark Skeide

Damage detection in fish farming nets using video analysis.

Bachelor's project in Automation Engineering
Supervisor: Arne Gerhard Styve and Saleh Abdel-Afou Alaliyat
May 2019

Bachelor's project

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Kunnskap for en bedre verden

Vebjørn Bjørlo-Larsen
Endre Slettemark Skeide

Damage detection in fish farming nets using video analysis.

Bachelor's project in Automation Engineering
Supervisor: Arne Gerhard Styve and Saleh Abdel-Afou Alaliyat
May 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Kunnskap for en bedre verden

Preface

This is a final major assignment of two automation students at the department of ICT and science at NTNU in Ålesund. The assignment gives 20 points of credits.

The assignment is made by Stranda Prolog AS, who supplies logistics solutions to the aquaculture industry. Stranda Prolog As has made a remote operated underwater vehicle called Manta net-cleaner that is used to wash the net in fish-cages. The assignment is to use video images from the net-cleaner to detect damage in the net.

The members of the group mainly chose the assignment due to interest in digital image processing.

Special thanks to:

Kjetil Osland Brekken and Jacob Hoseth at Stranda prolog AS.

Our supervisors at NTNU Ålesund: Arne Gerhard Styve and Saleh Abdel-Afou Alaliyat.

Ålesund, May 18, 2018

Vebjørn Bjørlo-Larsen

Vebjørn Bjørlo-Larsen

Endre S. Skeide

Endre Slettemark Skeide

Figure 1: Signature

Summary

Today, there is great focus on using new technology to automate the aquaculture industry. One of the companies that contributes to this is Stranda Prolog AS, who has submitted this bachelor thesis. Stranda Prolog has developed a remote-controlled underwater vehicle named Manta net-cleaner to clean the net in fish-cages for the aquaculture industry. The Manta Net Cleaner also delivers video streaming that shows the net.

The purpose and goal of this bachelor thesis is to create a computer software that can analyse video from the Manta net-cleaner to look for damage in the net. Such a software must be user-friendly and be able to present analysis results to the user in a good way. Image processing is used to perform the analysis. This report presents theory that is the basis for building the desired solution, and describes how the group has proceeded to develop the final solution using Scrum as development methodology.

The result of this project is a complete computer software for analysing video files from the Manta net-cleaner. The software is user-friendly and offers functionality that allows the user to customize the analysis to different videos. The program presents the results in a graphical user interface, in addition, it offers functionality to generate a PDF report.

The prepared solution meets the goals that formed the basis for the project.

Contents

Abbreviations

GUI Graphical User Interface, user interface between user and the program

MOA Minimum object area, parameter used in knots density analysis algorithm to check if an area is big enough to be considered as a hit.

KDA Knot Density Analysis

LMA Large Mask Analysis

LRA Large Region Analysis

XML eXtensible Markup Language

PDF Portable Document Format

List of Figures

List of Tables

Chapter 1

Introduction

Compared to other industries, the aquaculture industry has been behind when it comes to using new technology and automated solutions. Today, the aquaculture industry is growing rapidly. [25] In connection with this, the focus on using new technology and automated solutions has also increased in the industry. [38] [1] One of the companies working on such solutions aimed at the aquaculture industry is the group's assignment leader, Stranda Prolog AS. They have developed a remote-controlled underwater vehicle named Manta net-cleaner to clean the net in cages for the aquaculture industry using high pressure flushing. The net-cleaner also delivers video of the net in the cages as it is washing. The task is to use this video to automatically detect damage in the net.

At the start-up meeting for the project (see appendix), Jacob Hoseth (chief technology officer at Stranda Prolog AS) gave a good introduction to the current issue. Existing product has no automatic function for detecting damage in the net. As the product works today, it depends on the pilot that is operating the Manta net cleaner to detect any damage manually.

Damage to the net could lead to fish escaping. This is a major environmental problem because farmed fish can mate with wild fish and destroy the genes of the wild fish. [25] For those who own and operate the fish farms, fish that escape both lead to lost income and a weakened reputation. An automatic detection of damage in the net can therefore be an attractive additional function for the net cleaner. But it requires that the function is accurate.

In collaboration with the client, the group chose early to use Java and OpenCV to create software that uses image processing to detect damage in the net. It was also decided that the group will try to create an easy-to-use application for analysing video files from the Manta net-cleaner and then report any findings in the video to the user in a good way. The software will be almost the same if you perform the analysis in real time or later. Because of this, the group in collaboration with the client choose to limit the task to focusing on processing stored video files, but also to prepare for implementation of real-time analysis.

One of the challenges with this task is that the camera angle is highly variable. The net cleaner is controlled manually and maneuvered by propellers, therefore it is not always close to the net. Another challenge is that there are different lighting conditions in the cages. This is particularly challenging if there is strong sunlight that is reflected through the water. In addition to this, there are knots and splices in the net that can make the pattern of the masks irregular. Another challenge is fish that swims in front of the camera.

1.1 About the Manta net-cleaner

The Manta net-cleaner is a remote-controlled underwater vehicle developed by Stranda Prolog AS to clean the net in the cages for the aquaculture industry using high pressure flushing. The Manta is controlled from a mother vessel that lowers it into the cages by using a crane. The driver (or pilot) operates the Manta from a control room aboard the mother vessel. The driver maneuvers the Manta around the cages using a video stream from cameras mounted on the Manta. These video images are displayed on screens in the control room. On the video stream, the driver of the Manta can see the net, and thus he can also see where he has washed and not. The driver drives the manta around in the cage and washes the net until the entire net is clean. If the driver of the Manta is to detect any damage in the net, he must be able to detect these on the video stream. The opportunity the driver has to see such damage in the net while operating the manta is limited.

For more information, see appendix, Meeting report 11.01.2019.

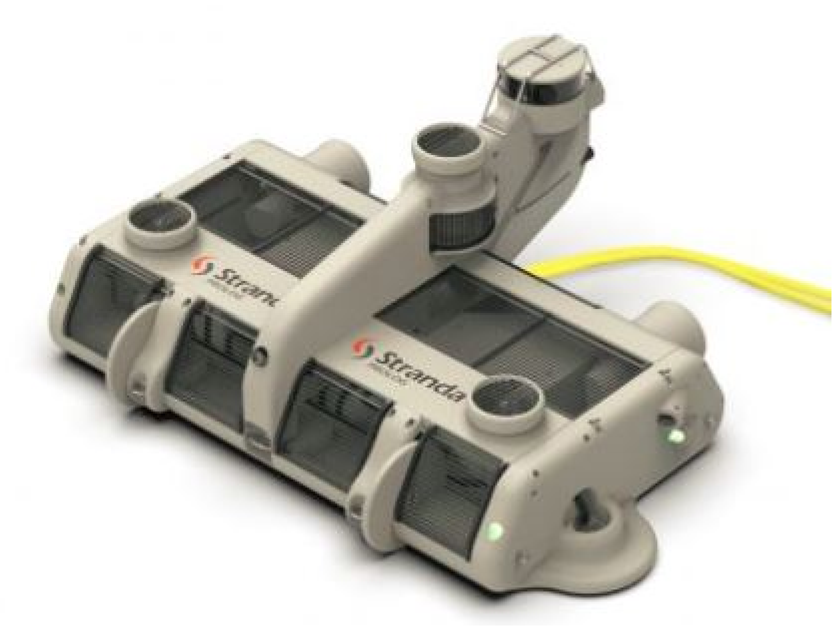


Figure 1.1: Picture of the Manta Net Cleaner. [6]

.

1.2 About the assigner

Figure 1.2: Picture of the Stranda Prolog AS logo.
[6]

The assignment provider Stranda Prolog AS was established in 1946 as a motor workshop. Today they deliver logistics solutions for the aquaculture industry. Ever since the beginning, the company has been owned and run the Hoseth family. The company currently has 60 employees and is mainly based in Kristiansund. They still have business in the original premises at Langøyneset on Averøya.[7] Stranda Prolog AS is a company with a long history that delivers quality products within the aquaculture industry.

1.3 Objectives

The basic problem is that Stranda Prolog AS would like their Manta net cleaner to contain functionality that makes it easier to detect and report damages / holes in the net. The objective of the assignment is that the group will prepare a solution to this problem by using image processing and analysis of video stream from cameras that already exist on the Manta net cleaner. By working on this assignment, the group will learn and practice much valuable programming and image processing knowledge. It is also a valuable goal to get training in performing such a project in collaboration with supervisors and company.

The Objectives for this report thesis are:

1. Obtain information and familiarize with image processing features that the group think may be useful.
2. Assemble such image processing functions to create complete analysis methods to detect damage in nets.
3. Use such image processing features to create functionality that can distinguish too-poor-quality images from being used to detect damage in nets.
4. Test and calibrate these analysis methods to work as optimally as possible.
5. Build frameworks and assemble these features into a complete working program.
6. Create functionality to store results in XML files and present them as PDF document.
7. Create a user-friendly graphical user interface for the program.
8. Document the entire project well for future use.

1.4 Limitations

The project has some limitations. The most obvious limitation the group sees is the quality of the video produced by the Manta. The video is filmed at an angle, and the irregular movements of the manta makes it hard to provide a safe analysis of the entire cage. The supply of large enough quantity of good video material will probably also be a limitation.

In order to best calibrate our analyses, we need a lot of video material. Since the goal is to detect damages in the net, we should ideally test and calibrate the program against video files that contain nets with a multiple different damaged parts. Such damages are rare, which makes it is difficult to obtain video containing such damages.

Another limitation is the group's unfamiliarity with OpenCV. The group has previous experience using MATLAB's Image Processing Toolbox, but will have to familiarize ourself with OpenCV to find out which functions are available. This will take some time, which is the greatest limitation the group sees.

Another limitation is processing power. The plan is to eventually run the program concurrently when washing the net, which means that the image processing and analysis can't take too long.

1.5 Structure of the Report

Further in the report you can read about the theoretical basis on which the project is based, and the various methods that have been used. The report thoroughly describes the approach and development of the project from start to end. The final results are described in the conclusion at the end of the report.

Chapter 2 - Theoretical basis: Chapter two gives an introduction to the theoretical background which is the basis for the group‘s work on this assignment.

Chapter 3 - Method: Contains a description of the methodology and materials that were considered throughout the project.

Chapter 4 - Result: Contains a description of what the group has developed, and how the group has found the different solutions.

Chapter 5 - Discussion: A discussion of the results, advantages, disadvantages, and plans for further development.

Chapter 6 - Conclusions: This chapter present an overall conclusion and final results of the whole assignment.

Chapter 2

Theoretical basis

2.1 Digital image processing

Digital image processing and analysis is a field that continues to experience rapid growth, with applications in many facets of our lives. Digital image processing, also referred to as computer imaging, can be defined as the acquisition and processing of visual information by computer.[44, Page xvii]

2.1.1 Colour depth

Colour images

Colour images can be modeled as three-band monochrome image data, where each band of data corresponds to a different color. The actual information stored in the digital image data is the brightness information in each spectral band. [44, Page 47]

Gray-scale images

Gray-scale images are referred to as monochrome ("one color") images. They contain brightness information only, no color information. The number of bits used for each pixel determines the number of different brightness levels available. The typical image contains 8-bits per pixel data, which allows for 256 (0-255) different brightness (gray) level values.[44, Page 46] Conversion from colour to grayscale is usually done using the following formula,

$GrayscaleValue = R * 0.299 + G * 0.587 + B * 0.114$

where R = red value, G = green value and B = blue value [30]

Binary images

Binary images are the simplest type of images, and can take on two values, typically black and white, or "0" and "1". A binary image is referred to as a 1-bit per pixel image, because it takes only 1 binary digit to represent each pixel. These types of images are most frequently used in computer vision applications where the only information required for the task is general shape, or outline information. Binary images are often created from gray-scale images via a threshold operation, where every pixel above the threshold value is turned white ("1"), and those below it are turned black ("0").[44, Page 45]

2.1.2 Smoothing

Smoothing or blurring is often used to reduce noise within an image or to produce a less pixelated image. The most basic smoothing filters are spatial filters using convolution to find the average or median value within a neighbourhood. This neighbourhood is often a 3x3, 5x5 or 7x7 area, but can be larger. [44, Page 82]

Gaussian Filter

Gaussian blur is one of the most commonly used smoothing filters. Gaussian filtering is done by convolving each point in the input array with a Gaussian kernel and then summing them all to produce the output array.[40]

Bilateral Filter

A bilateral filter is based on a spatial filer, such as a gaussian filter, but also takes into account how similar the pixel values are. In other words, a bilateral filter combines spatial and range filtering. In smooth regions, pixel values in a small neighborhood are similar to each other, and the bilateral filter acts essentially as a standard spatial filter. In boundary regions, values close to the origin pixel are weighted more heavily, resulting in a much sharper transition compared to simple spatial filters. [27]

2.1.3 Morphological operations

Morphology relates to the structure or form of objects. Morphological filtering simplifies a segmented image to facilitate the search for objects of interest. This is done by smoothing out object outlines, filling small holes, eliminating small projections, and other similar techniques. The two principal morphological operations are dilation and erosion. [44, Page 183]

Erosion

Erosion shrinks objects by etching away (eroding) their boundaries. The erosion operator uses two pieces of data as inputs. The first is the image which is to be eroded, and the second is a structuring element (or kernel). The structuring element determines the effect of the erosion on the image.[44, Page 183]

Dilation

Dilation allows objects to expand, thus potentially filling small holes and connecting disjoint objects.The dilation operator uses two pieces of data as inputs. The first is the image which is to be eroded, and the second is a structuring element (or kernel). The structuring element determines the effect of the dilation on the image. [44, Page 183]

Morphological Opening

Opening consists of an erosion followed by a dilation, using the same structuring element, and can be used to eliminate all pixels in regions that are too small to contain the structuring element. In this case, the structuring element is often called a probe, as it is probing the image looking for small objects to filter out of the image. [44, Page 185]

2.1.4 Skeletonization

Skeletonization is the process of eroding an object in a binary image to the point of it only being one pixel wide. [44, Page 191]

Morphological Skeleton

It is possible to find the skeleton of a binary image using morphological operations. A naive approach to finding the skeleton is to erode or dilate an image, and then either subtract the original image from the dilated image, or subtract the eroded image from the original image. This will leave an outer skeleton, but is often heavily fragmented and needs further processing to form a complete skeleton.

A more precise morphological skeleton can be found using multiple hit-or-miss operations. This is an iterative method, where the result from each iteration is defined as the result from subtracting the hit-or-miss operations from the original image, until no more changes can be done. [44, Page 191]

Zhang-Suen Thinning Algorithm

The Zhang-Suen Thinning algorithm is a fast parallel thinning algorithm. It consists of two subiterations: one aimed at deleting the south-east boundary points and the north-west corner points while the other one is aimed at deleting the north-west boundary points and the south-east corner points. End points and pixel connectivity are preserved. [48]

Branch points

A branch or fork point in a skeleton image can be defined as point where three or more branches meet. This can be found by searching for white pixels with three or more 4-connected white pixels [43], [5]. Connectivity refers to the way in which an object is defined, how pixels are connected to each-other. This is determined based on the neighbour pixels. The two main ways of defining connectivity is by looking at 4 (north, east, south, west) or 8 (all) neighbouring pixels, referred to as 4-connectivity or 8-connectivity. [44, Page 96-96]

2.1.5 Segmentation

Image segmentation is one of the most important processes for many applications. In image analysis and computer vision, segmenting the image correctly into real objects of interest is critical for object classification which will determine the success or failure of an algorithm. The goal of image segmentation is to find regions that represent objects or meaningful parts of objects [44, Page 165-166]

Thresholding

Image thresholding is a simple, yet effective, way of partitioning an image into foreground and background objects. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images. This is done using a threshold value, where all pixels above the threshold are set to 1, and all pixels below set to 0. Image thresholding is most effective in images with high levels of contrast.[42]

Otsu's Method

Otsu's method is an nonparametric and unsupervised method of automatic threshold selection for image segmentation. It works by searching for the threshold value that minimizes the within group variance. The within group variance is a weighted sum of the variances of the two groups [35], [44, Page 176-178].

Adaptive thresholding

In simple thresholding, the threshold value is global, i.e., it is same for all the pixels in the image. Adaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions of an image. [17]

2.1.6 Distance Transform

The distance transform is an operator normally only applied to binary images. The result of the transform is a graylevel image that looks similar to the input image, except that the graylevel intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point.[36]

2.1.7 Contour tracing

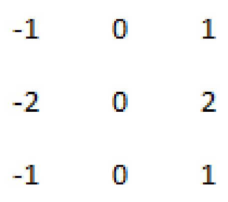
Contour or boundary tracing is used to define objects in a binary image. Contours can be found using OpenCV's findContours function [28], which is based on the algorithm described by Suzuki and Abe [39]. An area property of each contour can be defined as the number of pixels contained in the contour.

2.1.8 Edge Detection

Edge detection methods are used as a first step in the line detection process. Edge detection is also used to find complex object boundaries by marking potential edge points corresponding to places in an image where rapid changes in brightness occur [44, Page 122].

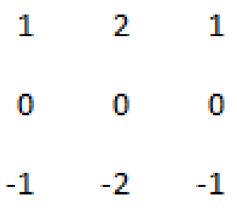
First derivative approximation

A first derivative approximation is done using two masks, one for vertical and one for horizontal edges. A commonly used approximation is the Sobel operator. The Sobel operator approximates the gradient by using a row and a column mask, which will approximate the first derivative in each direction, written as Gx and Gy. Another approximation of the first derivative is the Scharr operator. This operates in the same way as the Sobel operator, but have slightly different values.[44, Page 125]



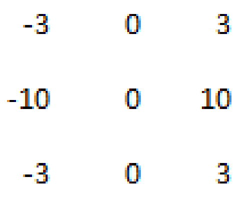
-1	0	1
-2	0	2
-1	0	1

Figure 2.1: Sobel Gx



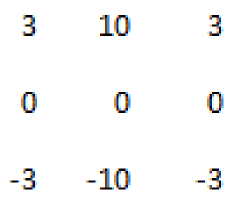
1	2	1
0	0	0
-1	-2	-1

Figure 2.2: Sobel Gy



-3	0	3
-10	0	10
-3	0	3

Figure 2.3: Scharr Gx



3	10	3
0	0	0
-3	-10	-3

Figure 2.4: Scharr Gy

Hugh Line Transform

The Hough Line Transform is a transform used to detect straight lines. OpenCV implements two kind of Hough Line Transforms: the Standard Hough Transform and the Probabilistic Hough Line Transform.[8]

2.1.9 Histogram

A histogram is a graph or a plot, which gives you an overall idea about the intnsity distribution of an image. It is a plot with pixel values (usually ranging from 0 to 255) in the X-axis and the corresponding number of pixels of this value on the Y-axis. [29]

Bimodal distribution

A bimodal distribution means that there are two clear peaks in the distribution. In a histogram, this indicates that there are two distinct groups of pixel values. [15]

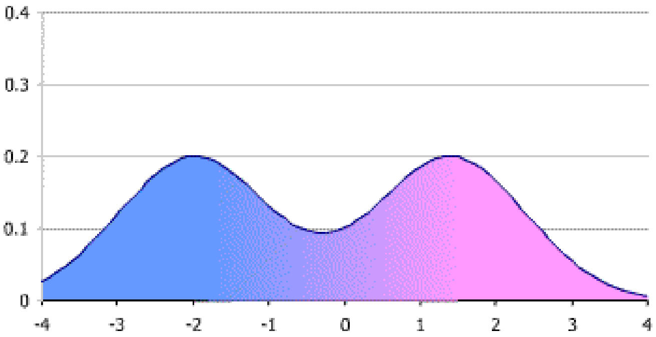


Figure 2.5: Bimodal distribution - credit: Maksim/Wikimedia Commons

2.2 Projects we have taken inspiration from

In the early phase of the project, we spent time preparing for the project by reading up on image processing theory and exploring imaging methods. As part of this preparation, we also searched for similar tasks that could be of interest to us. During the project, we have also looked for example on performing more specific image processing operations. Despite extensive searching, we have not found much information about or from previous projects that we have found relevant.

This is what we have found:

2.2.1 Forum: Pattern analysis of nets used in fish farming

A similar issue we came across was this: Pattern analysis of nets used in fish farming ([16]) The problem that is presented in this thread is very similar to our own, therefore it was of big interest to us. We are not familiar with the software that is used in this thread, nor have we prioritized this and therefore we did not get so much out of this. But we have read through the thread, and noticed some of the ideas presented. Especially the idea of looking at the net-masks as objects was something we wanted to investigate / experiment with.

2.2.2 Project: salamander pattern recognition

We also took some inspiration from a project from the Autumn 2018 Image Analysis (IE303512) course at NTNU in Ålesund. One of the projects presented illustrated a method of recognizing salamander patterns using a skeleton representation of the pattern. In this skeleton pattern end points and branch points were identified. We took inspiration from this in our knot density analysis (See 4.1.3).

2.3 Object oriented Programming

In this assignment, we have chosen to use object oriented programming to make our solution. At the heart of object oriented programming there are two concepts, objects and classes. Objects are created from classes. The class describes the kind of object, and the object represent individual instances of the class. We can communicate with objects by invoking methods on them. Objects usually do something if we invoke a method. Multiple objects can be created from a single class. [10, Page 3]

2.3.1 Coupling

The term coupling describes the interconnectedness of classes. We strive for loose coupling in a system - that is, a system where each class is largely independent and communicates with other classes via a small, well-defined interface. [10, Page 259]

2.3.2 Cohesion

The term cohesion describes how well a unit of code maps to a logical task or entity. In a highly cohesive system, each unit of code (method, class or module) is responsible for a well-defined task or entity. Good class design exhibits a high degree of cohesion. A cohesive method is responsible for one, and only one well defined task. A cohesive class represents one well-defined entity. [10, Page 260]

2.3.3 Encapsulation

Encapsulation is one of the fundamental principles of good class design. The encapsulation guideline suggests that only information about what a class can do should be visible to the outside, not about how it does it. Proper encapsulation in classes reduces coupling and thus leads to a better design.[10, Page 266]

2.3.4 Responsibility-driven design

Responsibility-driven design is the process of designing classes by assigning well-defined responsibilities to each class. This process can be used to determine which class should implement which part of an application function. Responsibility-driven design expresses the idea that each class should be responsible for handling its own data. [10, Page 271]

2.4 Concurrency

2.4.1 Threading

A thread is a process created within a program that has unrestricted access to its shared memory. The threads of a Java program are all executed within the same Java Virtual Machine (JVM), which means they can all share the same resources. A JVM is typically executed as a single operating system process. This means that for two or more threads to truly run parallel to each-other multiple processing cores are required. Most commonly parallel operations are achieved through time-sharing and context switching. [45]

In Java, there are two main ways of creating a thread, by extending the Thread class and overriding the run method, or by implementing the Runnable interface, which can then be used as an argument when creating a Thread object. In most cases, implementing the Runnable interface is preferred as Java only allows for single-inheritance. [45]

JavaFX provides an additional method of doing this called a Task, which is implemented by extending the JavaFX Task class. Tasks are used to implement logic for doing work on a background thread. This is required to make sure the GUI stays responsive during longer tasks, as the GUI can only be updated from the main JavaFX application thread. [33]

2.4.2 Synchronization

When two or more threads try to access or write to a shared resource, there is a risk of data corruption or a race condition to occur. To prevent this, synchronization methods are implemented. In Java, each object is associated with a mutual exclusion lock, which can be accessed through the synchronization modifier. When a method is labelled with the synchronized modifier, access to the method can only proceed once the lock associated with the object has been obtained. Hence, synchronized methods have mutually exclusive access to the data encapsulated by the object. [45]

2.4.3 Events

In JavaFX applications, events are notifications that something has happened. As a user clicks a button, presses a key, moves a mouse, or performs other actions, events are dispatched. An event is an instance of the javafx.event.Event class or any subclass of Event. Registered event filters and event handlers within the application receive the event and provide a response. Using JavaFX FXML, the events are defined in the FXML document, and handled in the main JavaFX application controller. [34]

2.5 Formats

2.5.1 Extensible Markup Language (XML)

Extensible Markup Language (XML) is a simple and flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML files is both human-readable and machine-readable.[21]

2.5.2 Portable Document Format (PDF)

PDF stands for "Portable Document Format". PDF is a file format designed to present documents consistently across multiple devices and platforms. It was developed by Adobe in 1992 and has since become one of the most widely used formats for saving and exchanging documents.[11]

Chapter 3

Materials and methods

3.1 Data

3.1.1 Video files

In the development and testing of the program, the group have been using video files that StrandaProlog AS has recorded from the Manta. This videos are produced by the built in cameras on the Manta. The videos are taken when the manta has been in normal use, cleaning the net in fish-cages. The group have received 3 hours of video in mp4 format that has been used for testing. Early in the testing process, the group also used pictures of nets with and without damage from the web.

3.2 Tools and libraries

3.2.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. [41]

3.2.2 MATLAB

MATLAB is a programming platform designed specifically for engineers and scientists. MATLAB can be used to analyse data, develop algorithms and create models and applications. [24] It combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly.[23]

Image Processing Toolbox

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. You can perform image segmentation, image enhancement, noise reduction, geometric transformations, and image registration using deep learning and traditional image processing techniques.[22]

3.2.3 Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995 [18]. The Java programming language is a high-level language that can be characterized as class-based and architecture neutral among others. In the java programming language, all source code is first written in plain text files ending with ".java" extension. Those source files are then compiled into ".class" files by the java compiler. A ".class" file does not contain code that is native to your processor, instead it contains byte codes, the machine language of the Java Virtual Machine.[12]

3.2.4 Netbeans

NetBeans IDE is a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems. The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications.[31]

3.2.5 JavaFX

JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms. JavaFX 2.2 and later releases are fully integrated with the Java SE 7 Runtime Environment (JRE) and the Java Development Kit (JDK). Because the JDK is available for all major desktop platforms (Windows, Mac OS X, and Linux), JavaFX applications compiled to JDK 7 and later also run on all the major desktop platforms [26].

JavaFX Scene Builder

JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces, without coding. Users can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the layout that they are creating is automatically generated in the background. The result is an FXML file that can then be combined with a Java project by binding the UI to the application's logic. [32]

3.2.6 Apache PDFBox

The Apache PDFBox® library is an open source Java tool for working with PDF documents. This project allows creation of new PDF documents, manipulation of existing documents and the ability to extract content from documents. Apache PDFBox also includes several command-line utilities. Apache PDFBox is published under the Apache License v2.0.[14]

3.2.7 Java JDOM Parser

JDOM is an open source, Java-based library to parse XML documents. It is typically a Java developer friendly API. It is Java optimized and it uses Java collections like List and Arrays. JDOM works with DOM and SAX APIs and combines the best of the two. It is of low memory footprint and is nearly as fast as SAX.[47]

3.2.8 Bat to Exe Converter

Bat To Exe Converter by Fatih Kodak [20] is a portable free tool which converts batch or CMD scripts to exe files [46]. Packing the Java program into an Exe file makes it much easier to install and run the program on a windows machine.

3.2.9 Inno Setup Compiler

Inno Setup Compiler is a free installer for Windows programs by Jordan Russell and Martijn Laan. First introduced in 1997, Inno Setup Compiler today rivals and even surpasses many commercial installers in feature set and stability. Inno Setup Compiler may be used completely free of charge, even when deploying commercial applications.[19]

3.2.10 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.[37]

Bitbucket

Bitbucket is a Git code management. It is built for professional teams, and gives teams one place to plan projects, collaborate on code, test, and deploy.[4]

3.2.11 Overview of tools and usage

Java	Code language used for the program.
JavaFX	Used to create the GUI.
JavaFX SceneBuilder	Used to build the GUI.
OpenCV	Used for image processing.
Apache PDF Box	Used to create PDF document.
Java JDOM Parser	Used to create, read and handle XML files.
Netbeans	Used to produce the program.
Matlab	Used for testing image processing.
Bat to Exe Converter	Used to create exe file from batch file.
Inno Setup	Used to create installation file.
Bitbucket	Used to produce parts of the project

Table 3.1: Overview of tools and usage

3.3 Materials

3.3.1 Computers

The group has been using their own personal Windows computers, and have been running Windows 8 and Windows 10 on the machines.

3.4 Methods

3.4.1 Project management

Developement method

The group chose to use Scrum as development method. Scrum is a framework that helps teams work together. Scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their profits and losses to continuously improve. [13] This framework is suitable for projects that involve developing software. The method involves working in sprints, usually lasting 1-4 weeks. The group and supervisors decided to to work in two week sprints.

At each new sprint, the group has set up a plan with prioritized tasks to work with. At the end of each sprint, the group has written a status report where it has been documented how far the group came with the planned tasks, and why the group might not have reached the goals. Jira and Confluence has been used in the project management, for example to post status reports, meeting notes, sprint tasks and other information about the project. Overleaf is used as a tool to write the main report.

Meetings

Between each sprint, status meetings have been held with the group's supervisors Arne and Saleh. These meetings have provided an opportunity for the supervisor to provide feedback on the work along the way, and for the group to request advice and guidance. In addition to these, the group has had frequent meetings with representative of Stranda Prolog AS Kjetil Osland Brekken (see appendix for meeting notes from the meetings with supervisors and representative of Stranda Prolog AS). In this way, the company has had the opportunity to influence the development of the product along the way.

Jira

The group has used Jira as an aid in flexible project management. Jira is a family of products built to help all types of teams manage their work. Jira offers several products and deployment options that are purpose-built for Software, IT, Business, Ops teams, and more. Products and apps built on top of the Jira platform help teams plan, assign, track, report and manage work. The Jira platform brings teams together for everything from agile software development and customer support to managing shopping lists and family chores.[3]

Confluence

Confluence is a content collaboration tool used to help teams to collaborate and share knowledge efficiently. The group has used confluence to among other things make meeting notes and status reports. With Confluence, the users can create pages and blogs which can be commented on and edited by all members of the team. Confluence has also been designed to integrate with Jira and they have many integration points, giving Confluence users the ability to view, interact with, and reference Jira issues from a wiki page.[2]

Overleaf

Overleaf is a startup and social enterprise that builds modern collaborative authoring tools for scientists — like Google Docs for Science. Their primary product is an online, real time collaborative editor for papers, theses, technical reports and other documents written in the LaTeX markup language.[9]

Deliveries from the project

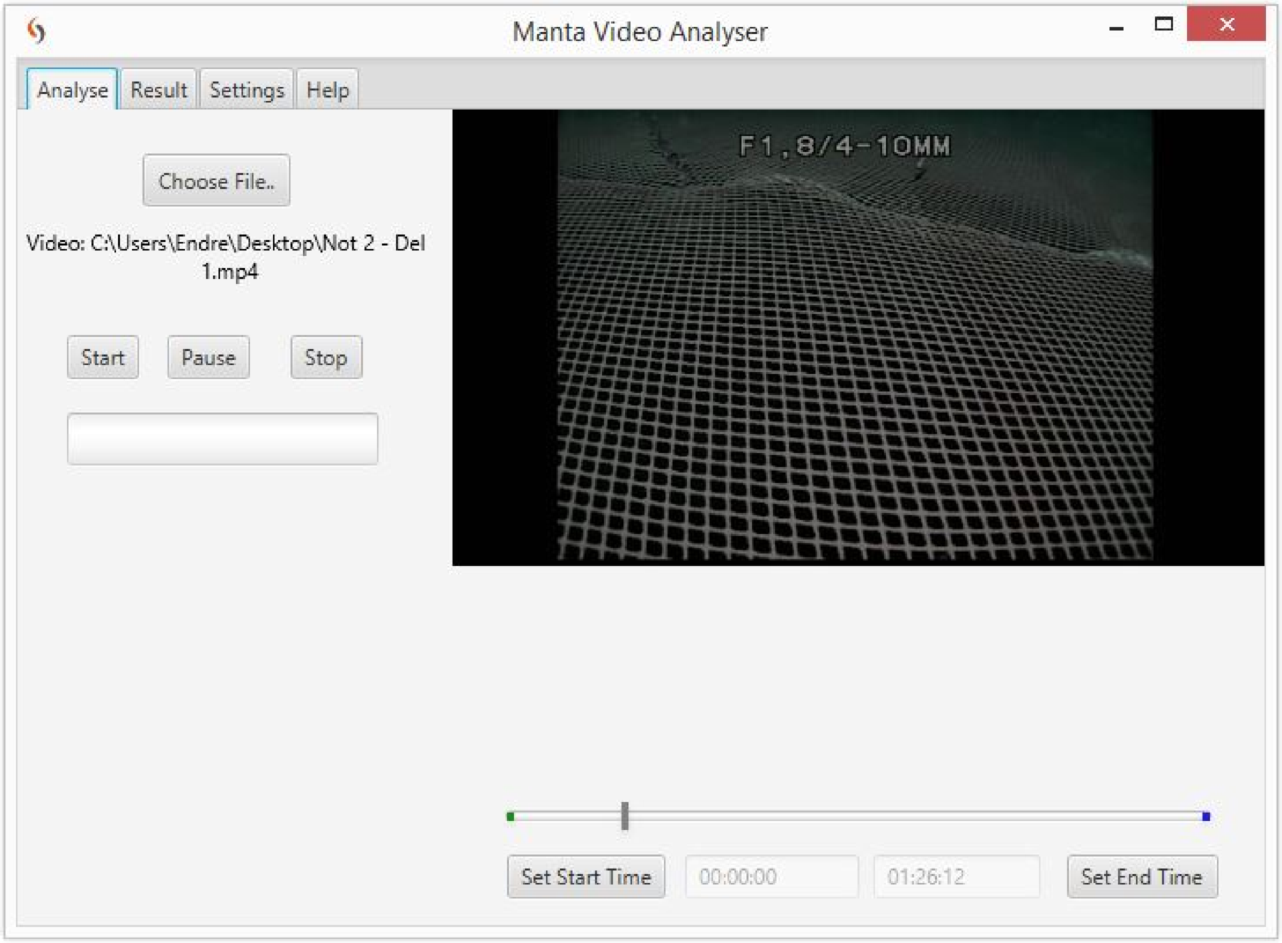
Deliveries from this project should include the software that is being developed. In addition, the group will also provide documentation in the form of a pre-project report and a bachelor's report when the project is completed. A good user guide will also follow the program.

3.4.2 Planned progress

The first part of the assignment was to thoroughly study the task and plan procedures for solving the assignments. In this process, a pre-project report was made (see appendix). The next step in the process was to thoroughly study various image processing methods that were found to be potentially relevant. Since the group had previous experience with the use of MATLAB, this program was planned to be used largely during testing. Further, the plan was to test and experiment with different combinations of relevant imaging methods to arrive at one or more aggregate methods that where able to detect deviations in the network. Further, the plan was to build a framework with functionality that picks up video and divides it into frames that is handled through the analysis methods. A user-friendly GUI should also be made and connected to the application.

Chapter 4

Result



Video: C:\Users\Endre\Desktop\Not 2 - Del 1.mp4

00:00:00

01:26:12

Figure 4.1: Picture showing the analysis application

The group's work on this assignment has resulted in a software that is able to analyse video files from the Manta net-cleaner with regard to detecting defects in the net. The software uses image processing to separate images with too poor quality for the analysis. Three different analyses have been developed by the group and added to the software to carry out the actual damage detection in the net. The software includes a functional and user friendly GUI (Figure: 4.1). Multithreading is used to achieve high speed analysis. The results from an analysis is presented in the GUI, and optionally as a PDF report.

4.1 Image Processing

The main part of the group's assignment is to detect damages in nets using image processing techniques. Through extensive research, testing, and calibration, the group has come up with the following use of image processing methods.

4.1.1 Pre-processing

Before running a frame through the hole detection algorithms, it has to be pre-processed. This process aims to remove unnecessary information and break the frame down into data easily manipulated by the computer. The video received from the manta is in colour, but for this project the group is mainly interested in background and foreground information, or a binary representation of the images. To achieve this, multiple experiments were conducted to find out which combination of methods best preserved the critical information when converting frames to binary images.

Step 1: Colour to grayscale

The first pre-proccessing step is to convert the colour image to a grayscale image. This is achieved using OpenCV's cvtColor method with the COLOR_BGR2GRAY flag. This converts the image into an 8-bit grayscale image as described in the theory section 2.1.1.

Step 2: Smoothing

The next step is to smooth the image. This aims to reduce noise while preserving the information needed for further processing. As described in the theory section there are multiple ways of achieving this 2.1.2. During the experiments, the group tested gaussian and bilateral filters, and using no filters at all.

To illustrate the impact of the noise removal, the same image has been run through the entire Large Masks Analysis (as described below) with different noise removal techniques:

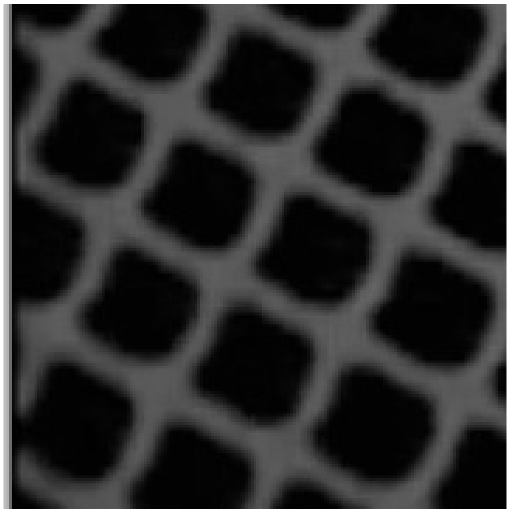


Figure 4.2: Picture showing the original image before it is run through the large mask analysis

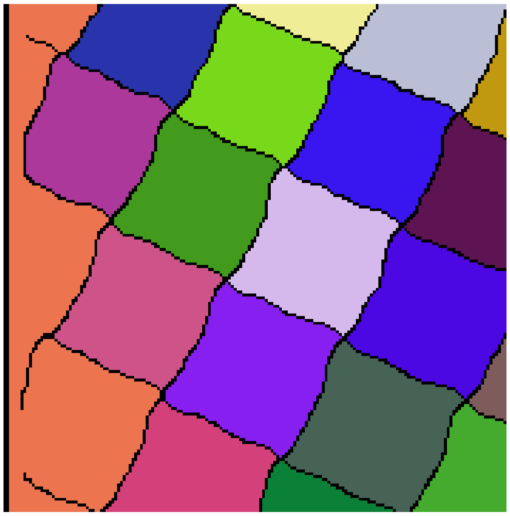


Figure 4.3: Picture showing the test image after it is run through the large mask analysis without any smoothing

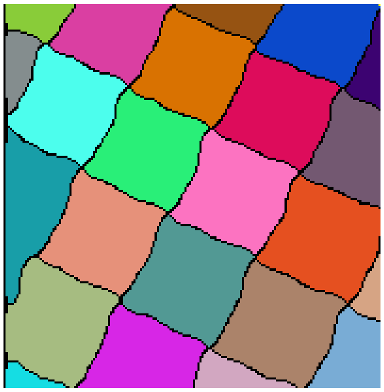


Figure 4.4: Picture showing the test image after it is run through the large mask analysis using Gaussian blur - 3x3 mask

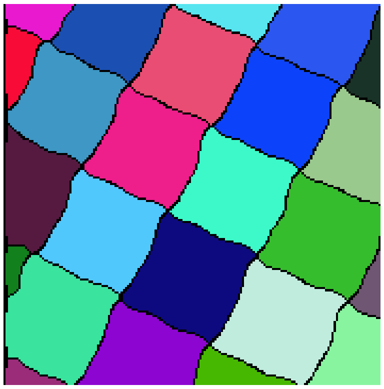


Figure 4.5: Picture showing the test image after it is run through the large mask analysis using Gaussian blur - 5x5 mask

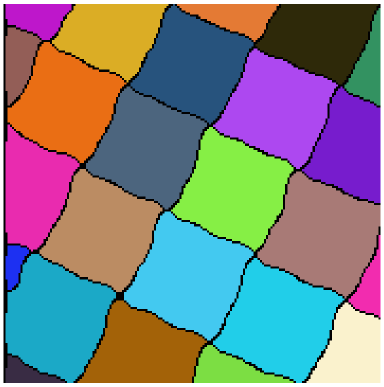


Figure 4.6: Picture showing the test image after it is run through the large mask analysis using Gaussian blur - 7x7 mask

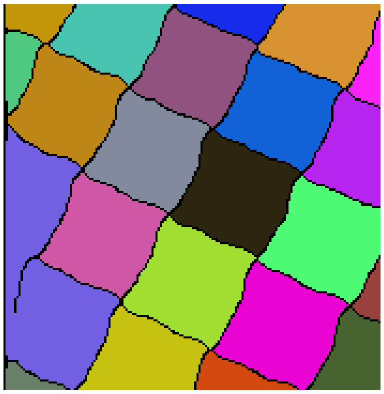


Figure 4.7: Picture showing the test image after it is run through the large mask analysis using Bilateral blur - 3x3 mask

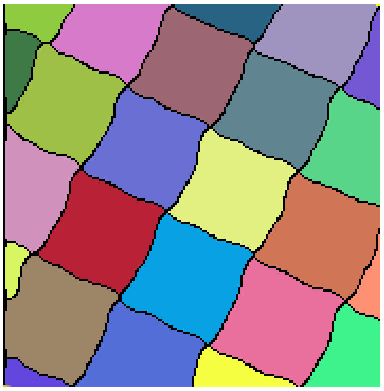


Figure 4.8: Picture showing the test image after it is run through the large mask analysis using Bilateral blur - 5x5 mask

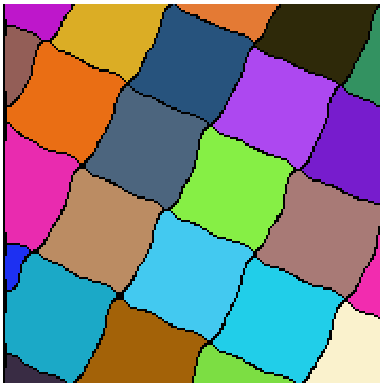


Figure 4.9: Picture showing the test image after it is run through the large mask analysis using Bilateral blur - 7x7 mask

The figures 4.3, 4.4 and 4.7 show that using no smoothing or small filtering masks can result in masks blending into each-other on the edges of the image. The figures 4.5, 4.6, 4.8 and 4.9 show that using larger mask sizes works better when trying to avoid this problem. The figures also show that there is no appreciable difference between gaussian and bilateral filters with larger masks. To determine which filter to use, the group ran a speed test to check if there were any appreciable speed difference between the methods.

Filter	**Time per Image (ms)**
Gaussian 3x3	11.6
Gaussian 5x5	11.6
Gaussian 7x7	11.9
Bilateral 3x3	13.8
Bilateral 5x5	16.3
Bilateral 7x7	20.5

Table 4.1: Speed-test results of different smoothing-masks

The speed test shows that bilateral filtering uses more resources. The absolute difference is small, but it does add up. In the tests there were no appreciable difference between the results, so the group ended up using a gaussian 5x5 filter.

Step 3: Thresholding

The next step is to threshold the grayscale image. The goal is to segment the image into background and foreground objects; water and fish net. As described in the theory section, there are various methods to achieve this 2.1.5. One of the experiments conducted was to see if a fixed threshold value could be used. Holes in the net are expected to be large and dark regions, so in theory only the darkest regions of the image should be of interest to us.

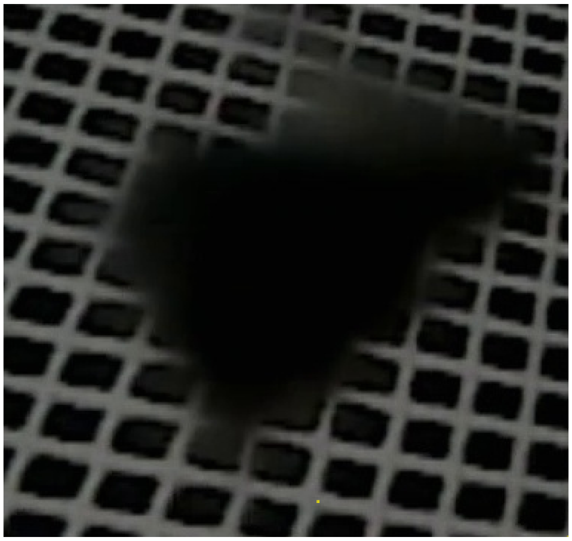


Figure 4.10: Picture showing an original test-image of a net with potential damage

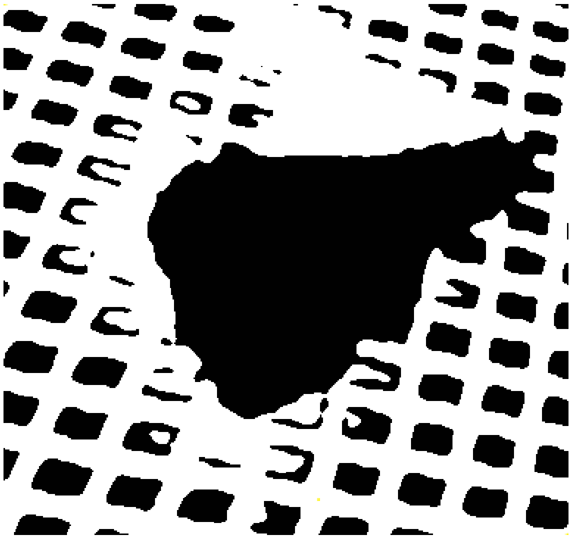


Figure 4.11: Picture showing the thresholded image

Figures 4.10 and 4.11 show that the hole is clearly detected. However, on images with brighter backgrounds we see the problem with this approach. In 4.12 hardly anything of the image is detected. In the image there is no hole, but any holes would have the same background colour as in the image, so would not show up as an object.

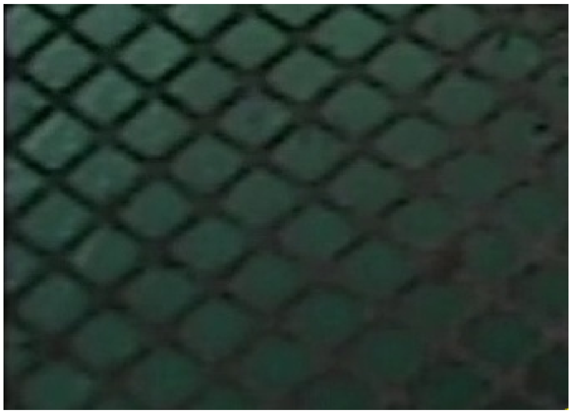


Figure 4.12: Picture showing an original test-image of a net with a bright background

Figure 4.13: Picture showing the thresholded image

The group moved on to testing methods to individually threshold each image, Otsu's method and adaptive thresholding were tested. As described in the theory section, threshold values can be calculated and used globally for the entire image or locally for smaller sections of the image

2.1.5. To test whether global or local thresholding worked best for this project multiple images were run through both methods:

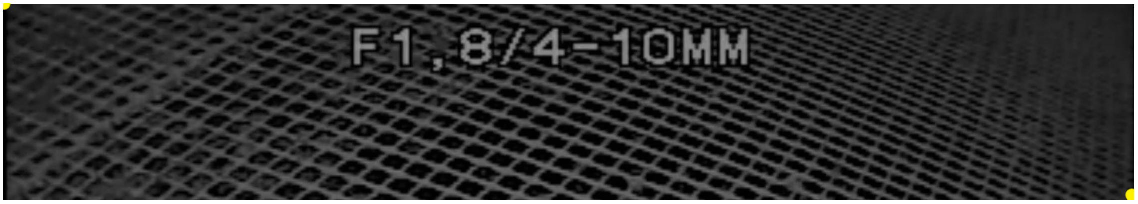


Figure 4.14: Picture showing a Grayscale test-image

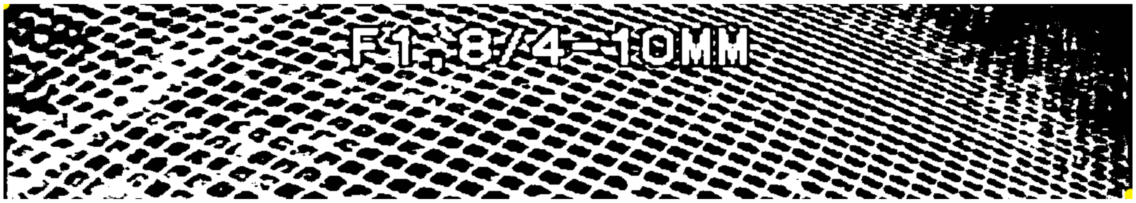


Figure 4.15: Picture showing global thresholding, Otsu's method performed on the picture from figure 4.14

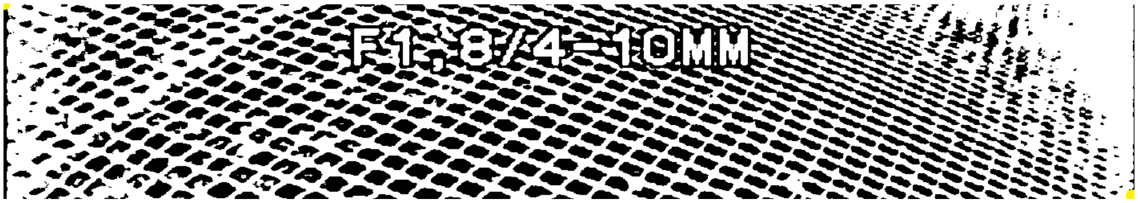


Figure 4.16: Picture showing adaptive thresholding, gaussian blur, mask size 335x335 performed on the picture from figure 4.14

From this, the group found that the middle section ends up practically the same with both methods, but that the top corners are black using global thresholding, white using adaptive thresholding. As seen in 4.14 these corners are darkened and the net masks are so close to each other that most of the contrast is taken out of the regions. Using global thresholding these areas tend to fall below the threshold value, thus ending up black. Using adaptive thresholding these areas are compared to the neighbouring area, so they end up being above the local threshold area. This is important for the project as the analysing algorithms looks for large black regions in the images.

One of the drawbacks of adaptive thresholding is that, if the mask size is too small, large areas of similar pixel values can be falsely segmented into two regions. To find out what mask size to use, another experiment was set up. Below is an example showing the findings:

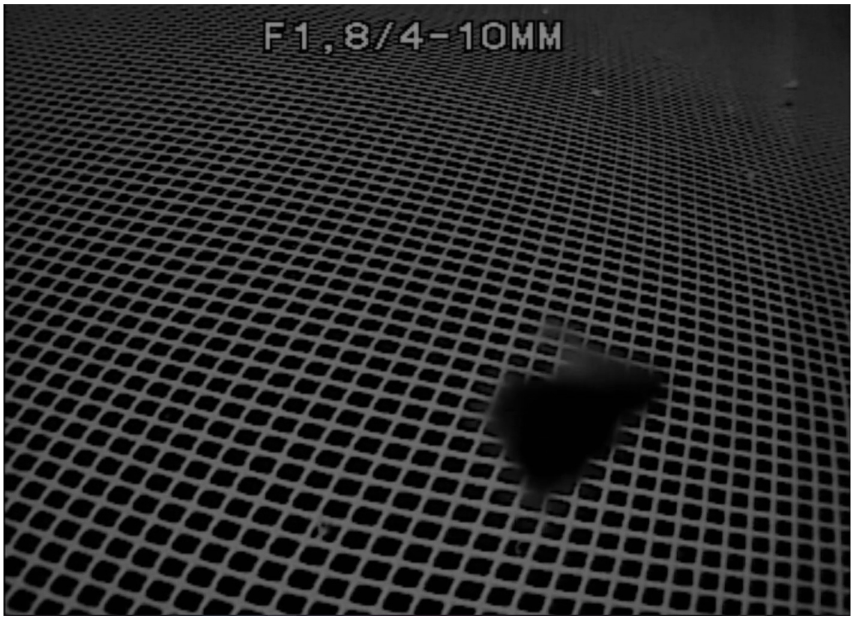


Figure 4.17: Picture showing a grayscale test-image

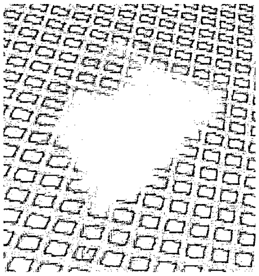


Figure 4.18: Picture showing adaptive threshold performed on the picture from fig. 4.17, using mask size: 5x5

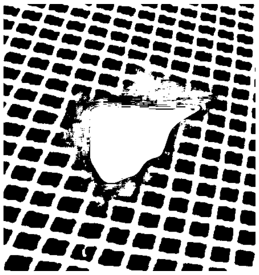


Figure 4.19: Picture showing adaptive threshold performed on the picture from fig. 4.17, using mask size: 55x55

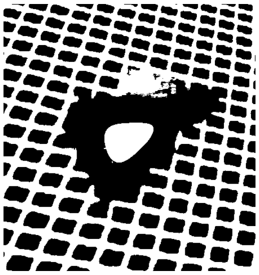


Figure 4.20: Picture showing adaptive threshold performed on the picture from fig. 4.17, using mask size: 175x175

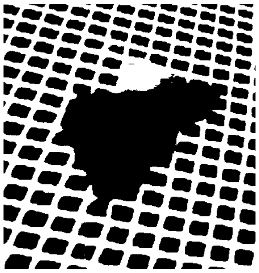


Figure 4.21: Picture showing adaptive threshold performed on the picture from fig. 4.17, using mask size: 335x335

From these figures it is clear that small mask sizes segments the black region into two objects. When using a 5x5 mask size, there are even white objects within each net mask. Using larger mask sizes, the net masks are correctly segmented. When using a large mask size (335x335), the object within the large black region gets entirely eliminated.

The right balance is to be struck between segmenting low-contrast areas accurately and not over-segmenting by creating objects where there shouldn't be any. From the testing, adaptive gaussian thresholding with a mask size of 335x335 seems to best fit these criteria.

Step 4: Skeletonization (thinning)

The final pre-processing step is to run the binary image through a thinning algorithm. According to the theory 2.1.4, this should give an image with black background and 1-pixel thick lines representing the net. To achieve this, simple morphological thinning 2.1.4 was tested, as shown below:

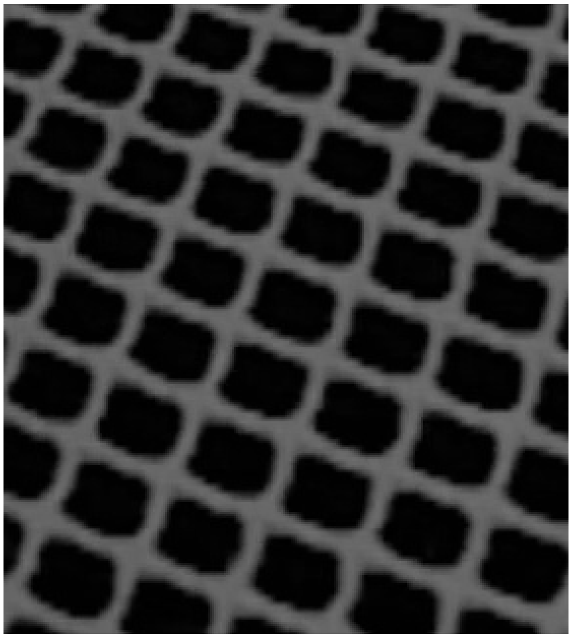


Figure 4.22: Picture showing a grayscale test-image

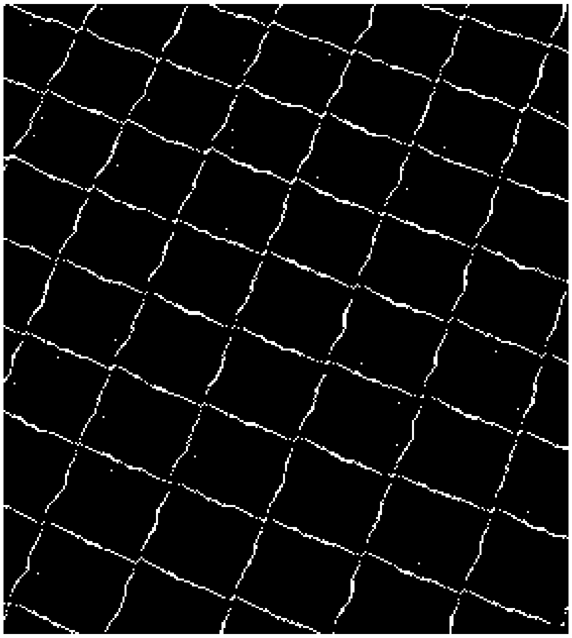


Figure 4.23: Picture showing morphological skeleton performed on the picture showed on figure 4.22

As seen in the results, this proved to be far too inaccurate for the group‘s needs. In fact, as can be seen on figure 4.23, hardly any line connects to another line. All the lines are clearly connected on the original image, so the idea of using simple morphological skeleton in favour of more advanced methods were scrapped.

The group continued by testing OpenCV's thinning algorithm. The algorithm is available from OpenCV's extra modules, and is based on the Zhang-Suen thinning algorithm (2.1.4). The Zhang-Suen thinning algorithm is an iterative method, so the group decided to copy and rewrite this code in Java. This allowed us to keep using the standard OpenCV library, and also gave more control over the number of iterations. The picture below shows the Zhang-Suen skeleton performed on the picture from Figure 4.22:

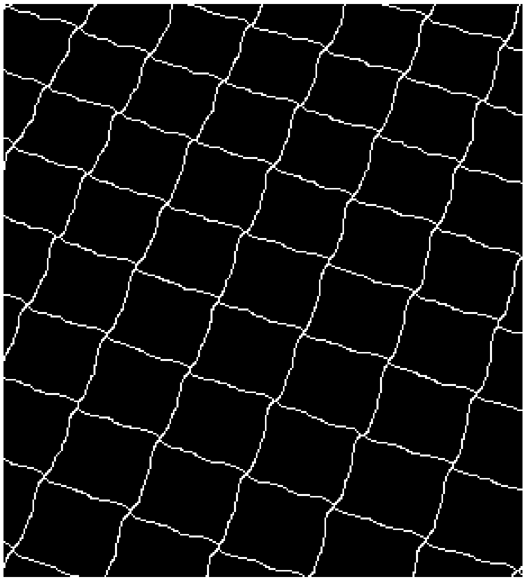


Figure 4.24: Picture showing Zhang-Suen skeleton performed on the picture from Figure 4.22

Figure 4.24 shows clearly that all the lines are connected, so this method was chosen above the others.

4.1.2 Quality Control

Due to the camera being mounted on top of the manta filming down at a set angle, one of the main issues to be solved is what to do with images where the masks can not be recognized in the net, either fully or partly. Below is an example of this problem.

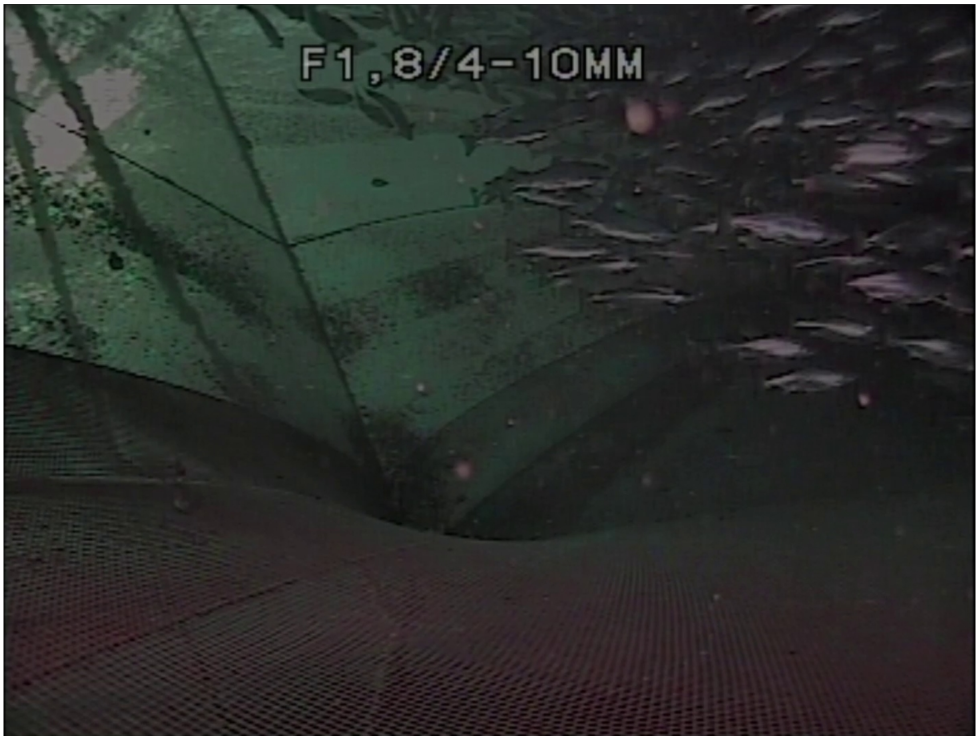


Figure 4.25: Example where it is not possible to recognize masks in the net

To find a way to discard the images of too poor quality, multiple methods have been tested.

Experiment 1: Finding peaks and valleys in the grayscale histogram to check for bimodal distribution of pixel values.

The first test conducted was to check if the grayscale histogram had a clear bimodal distribution 2.1.9. In theory, this should give an indication of whether or not there are two clear groups of pixels in the image, indicating background (water) and foreground (the net). The challenge when devising such a check is what to count as maximum and minimum points (peaks and valleys) in the histogram. Our solution was to go through each value and compare it to its close neighbours. If all neighbours were either below or above (with the possibility of some being the same value), the point is designated a peak or valley point. To prevent two values close to each-other counting as two extrema, each potential point is compared to the previous point. If it is too close in value or index place it is not counted. Where to set this limit, and how many neighbour values to count, can be changed in the code.

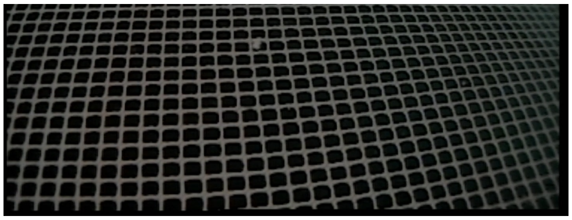


Figure 4.26: Picture showing a good quality image

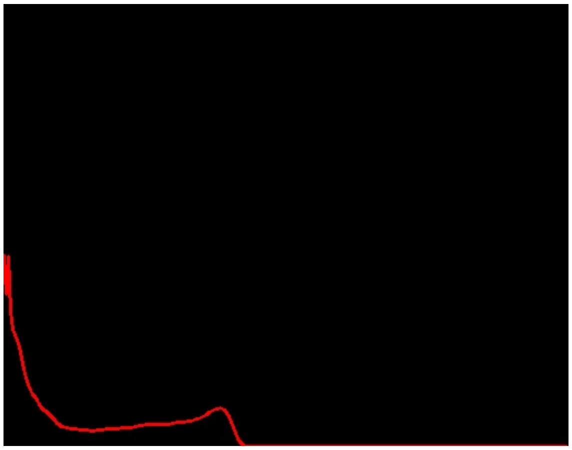


Figure 4.27: Picture showing the histogram of the image on figure 4.26

In figure 4.27 there is a clear peak of dark pixels, and a smaller, but still clear, peak of white pixels. Contrast that to figure 4.29, where the distribution is much closer to a unimodal distribution around a dark gray value.

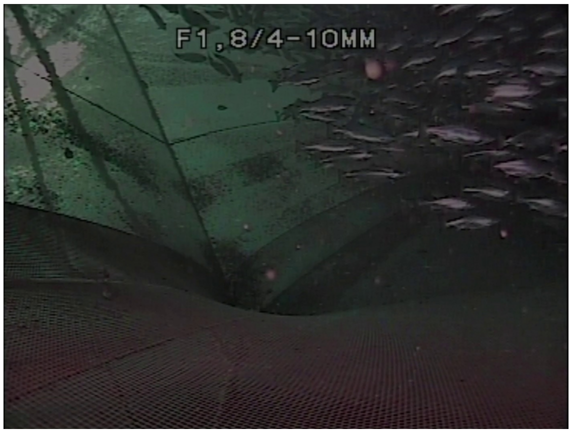


Figure 4.28: Picture showing a bad quality image

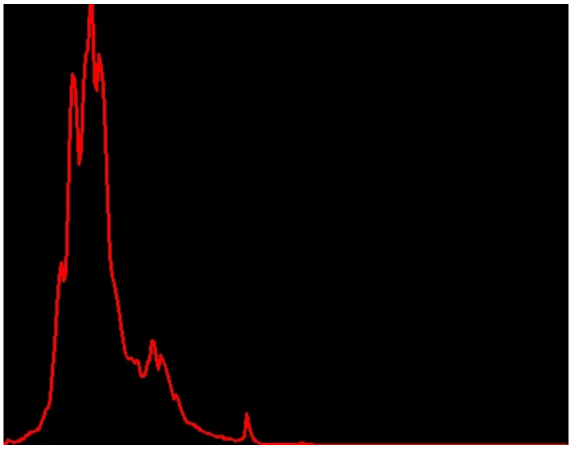


Figure 4.29: Picture showing the histogram of the image on figure 4.28

Experiment 2: Comparing histograms with known good images

When working on the bimodal distribution, it was noticed that most of the good images had similar histograms. Some experiments were conducted to test if this could be used to discard images of too poor quality. During our tests we had decent results using this method, but it was also clear that this approach would not work well on different videos unless the histograms were found for each video. We did not proceed further with this experiment.

Experiment 3: Finding horizontal and vertical edges, then checking if the number of edges in each direction are similar.

The group found that images of poor quality often had masks that have distinct lines in one direction, but blurred lines in the other directions. For example, horizontal lines can often be clearly found, while vertical lines are very indistinct. This can cause many long and narrow objects in the analysis. Good images do not tend to have this difference. Based on this, the group experimented with a method that creates vertical and horizontal edge images 2.1.8, and compares the number of edge pixels in each image.

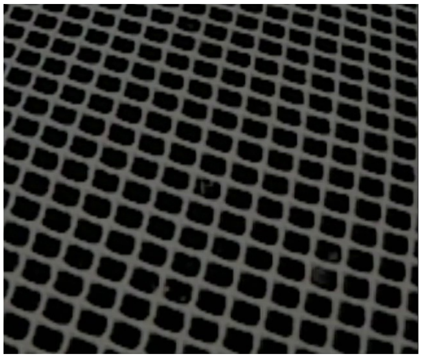


Figure 4.30: Picture showing a good quality image

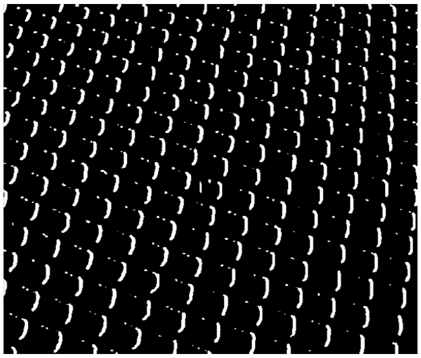


Figure 4.31: Picture showing the vertical edges

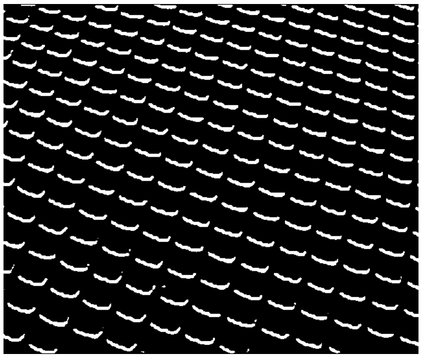


Figure 4.32: Picture showing the horizontal edges

Figure 4.30 shows a good quality image. Figures 4.31 and 4.32 show horizontal and vertical edges in the image. The pictures shows that it is possible to detect approximately as many distinct horizontal as vertical edges.

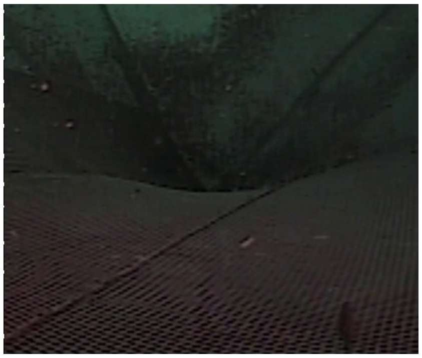


Figure 4.33: Picture showing a poor quality image

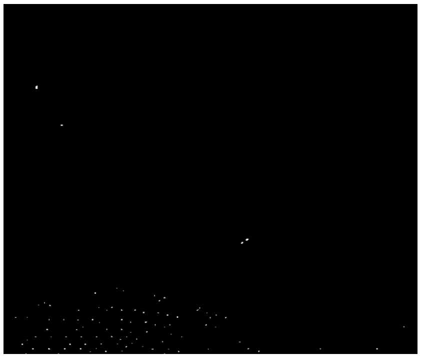


Figure 4.34: Picture showing the vertical edges

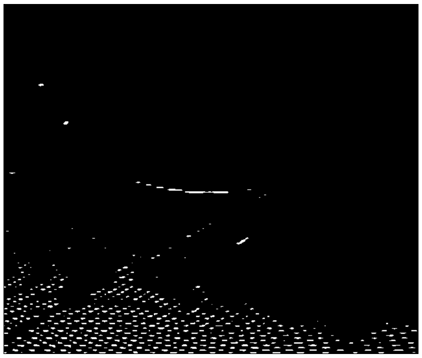


Figure 4.35: Picture showing the horizontal edges

Figure 4.33 shows an example of a poor quality image. As can be seen in figures 4.34 and 4.35, there are hardly any vertical edges, but quite a few horizontal edges. This indicates that the image is too poor to be used in further analysis.

Experiment 4: Black / White pixel ratio

The fourth experiment we conducted was using the binary image to count black and white pixels. In an image of good quality the amount of black background pixels should be higher than the thinner lines of white foreground pixels.

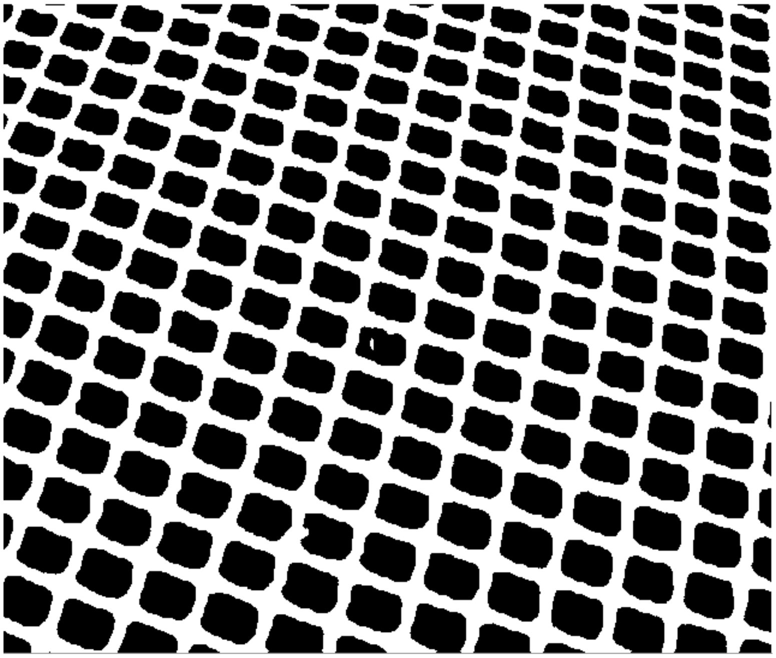


Figure 4.36: Picture showing the binary image of a good quality image

In figure 4.36 the number of white pixels is 151 299, number of black pixels 194 301. There are more black pixels than white, as expected. Looking at an image of poor quality:

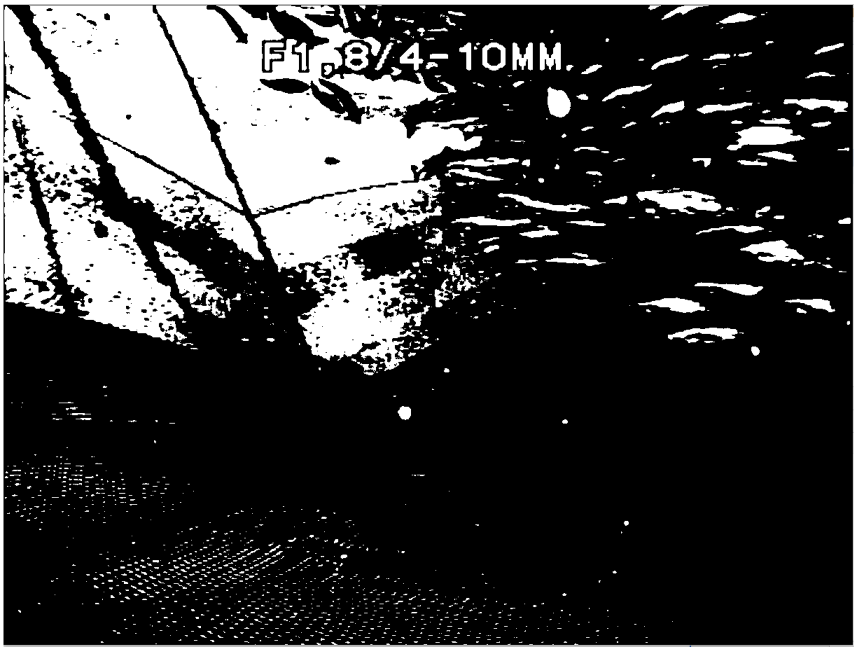


Figure 4.37: Picture showing the binary image of a poor quality image

In figure 4.37 the number of white pixels is 238 689, number of black pixels 870 907. Despite the image being of poor quality, it still results in more black pixels than white pixels, just like an image of good quality. One could try to check if the limit is within a certain limit, but during our testing we found the results to be too "random" to justify further testing.

Final solution

Based on the experiment results, we ended up combining the results from Experiment 1 (4.1.2) and Experiment 3 (4.1.2). For a section of an image to be determined good enough for analysis it has to pass both quality checks.

4.1.3 Hole detection algorithms

During the development, the group has devised three different hole detection algorithms.

- Large Mask Analysis, which searches for large masks in the net.

- Large Object Analysis, which searches for large objects in the net.
- Knot Density Analysis, which searches for areas with low knot density.

Each of the analyses is described in detail in the following sections.

Large Mask Analysis

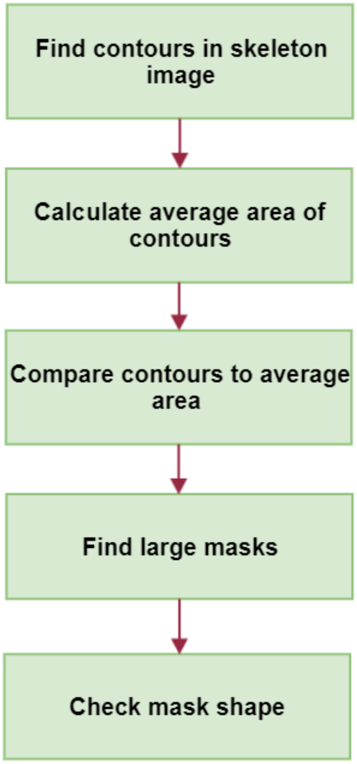

Find contours in skeleton image
Calculate average area of contours
Compare contours to average area
Find large masks
Check mask shape

Figure 4.38: Diagram showing main tasks in the Large Mask Analysis

The large mask analysis algorithm attempts to find hits by looking at the average mask area of all the net masks, and then checking if any of the masks are too large to be a regular mask. Figure 4.38 shows the main steps that the large mask analysis performs. The algorithm takes the skeleton image prepared by the pre-processes and finds all the contours of the masks. The mask contours are found using the OpenCV "findCountours()" function as described in the theory section 2.1.7. A contour-list is made, and all the contours that is found, is added to this list.

Further, the average mask area is found from the contour-list by adding the area of all contours and dividing by the total number of contours. To determine if any of the masks is a hit, each mask is then checked again. This is done by iterating through the list, with each area being compared to the average area. How much bigger an area has to be than the average area, can be adjusted by the user. During testing, a settings value somewhere in the range of 3-5 gave the best results. This means that any mask which is more than 3-5 times larger than the average area, is considered a hit.

During testing, the group often found that masks far away from the camera and at some special angles would end up as long, most often horizontal masks. This could potentially give false hits in the analysis. In addition, very long and thin holes rarely occur in fish nets, so the group chose to prevent such hits. To do this, any potential hit is compared to the smallest enclosing circle of the contour, found using OpenCV's minEclosingCircle() function. If the area of the contour is a lot smaller than the minimum enclosing circle, this tells us that the contour is a long and thin one. This helps removing false hits, but might also remove true hits, so this can be turned on/off in the settings.

To illustrate how the algorithm works on different images, the group have included three examples below.

Example 1

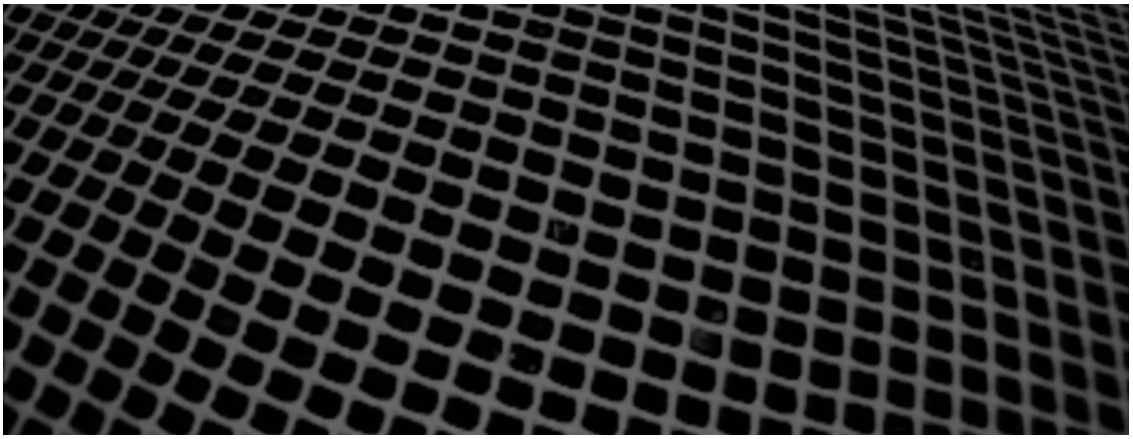


Figure 4.39: Example 1 - Original Image

Example 1 (4.39) is a typical image of the net without any holes. As seen on the picture, the net has shapes of regular sizes, and is at a slight angle.

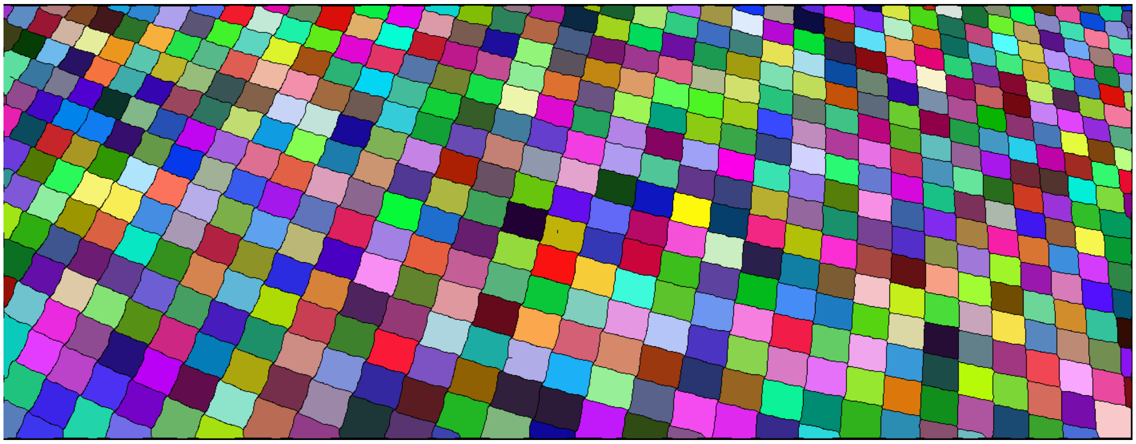


Figure 4.40: Example 1: Skeleton with objects randomly coloured

Figure 4.40 illustrates all the masks detected in the algorithm. Note that this picture is printed just for showing the contours. When the analysis runs, this information is stored in a list, so figure 4.40 is just printed for illustration.

Figure 4.41: Example 1: results

When the average area has been found, all the masks are compared to that area, the results of this can be seen below in figure 4.41.The figure shows that no hits were found in example one.

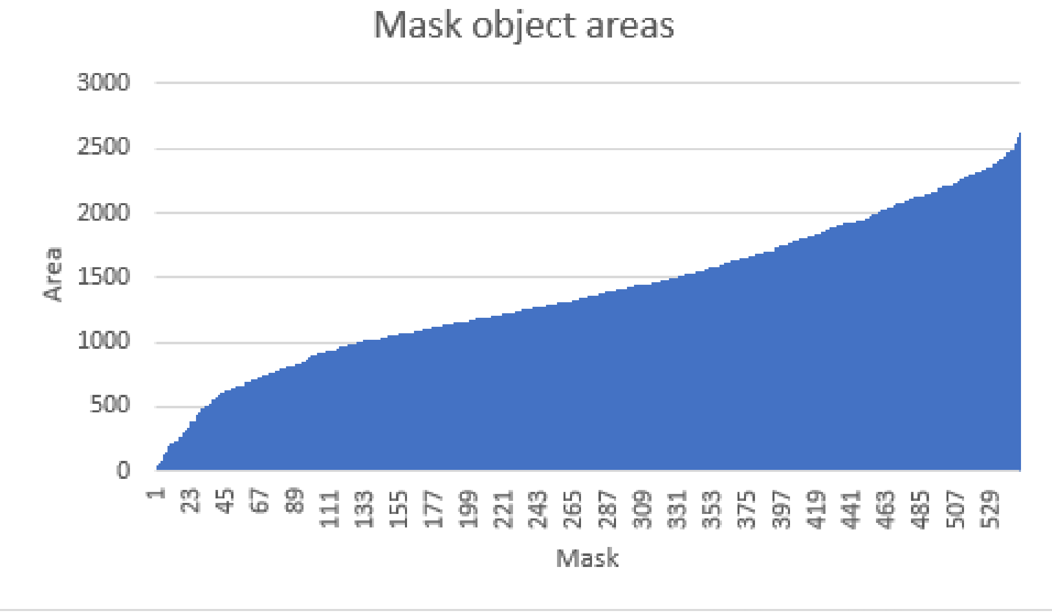


Figure 4.42: Example 1: Object areas in ascending order

Figure 4.42 shows that no holes are detected . Looking at the object area values in figure 4.42, they all fall within a range of 41 and 2627 pixels. The average value is 1386 pixels, so all objects are below the threshold value for a hole, which in this example is set to three times the average mask area.

Example 2

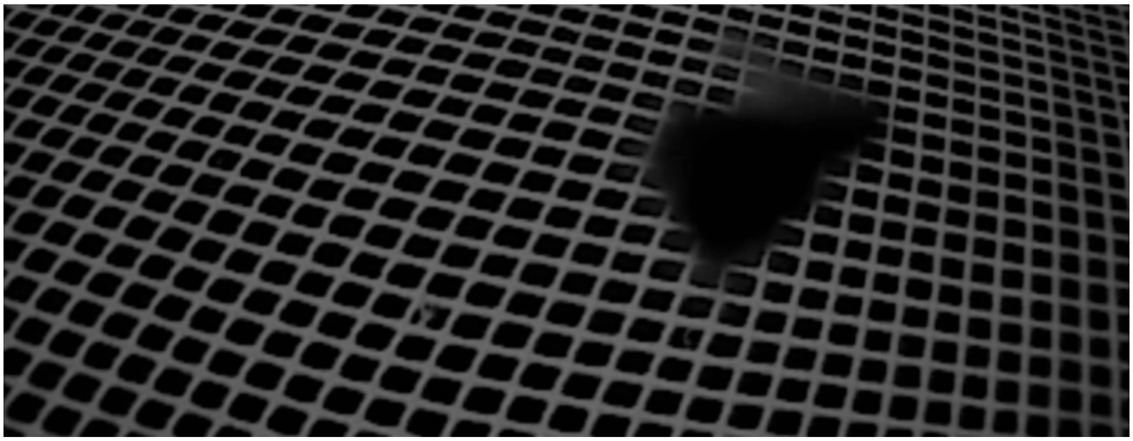


Figure 4.43: Example 2: large hole

Example 2 (4.43) is an image used to simulate a hole. It is not actually a hole, but rather a small piece of debris right in front of the camera that blocks out a small region of the image. A hole may look very similar to this picture, so the picture is a good example.

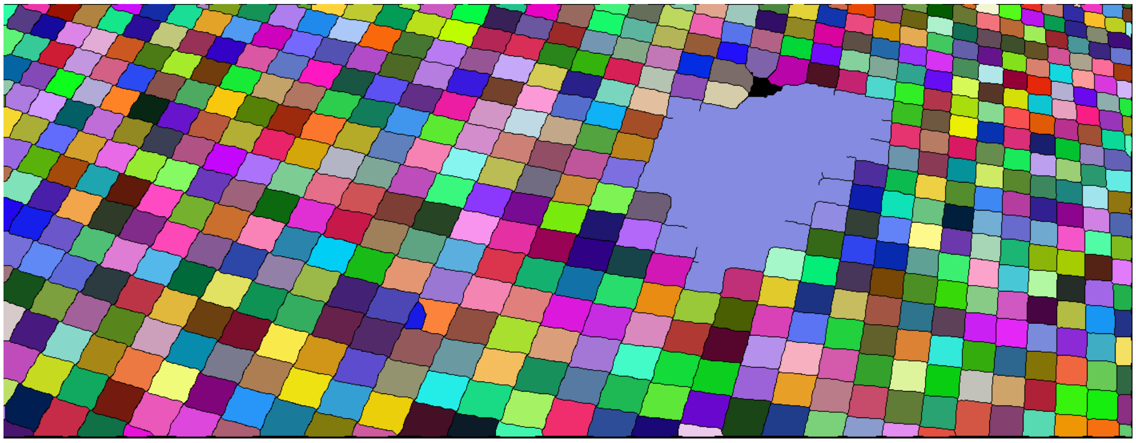


Figure 4.44: Example 2: Skeleton with objects randomly coloured

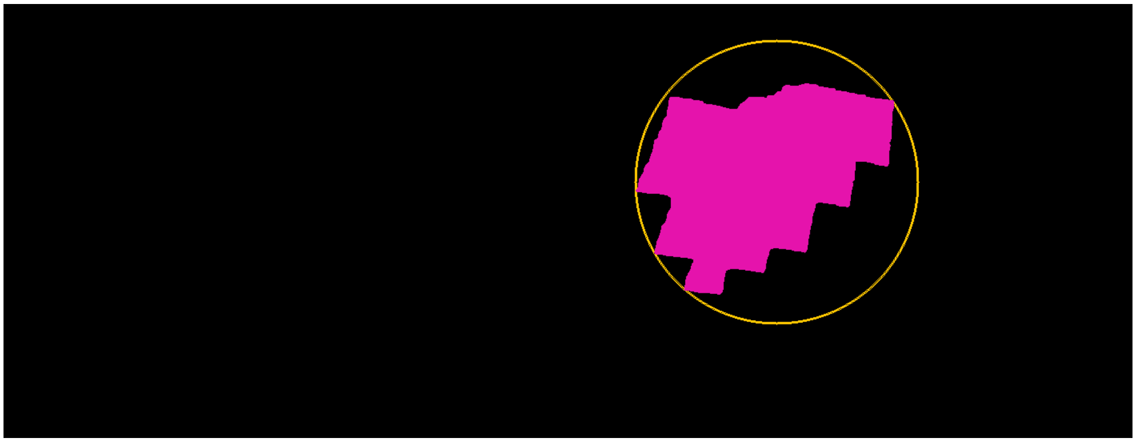


Figure 4.45: Example 2: results

Figure 4.45 shows that one hole is detected. Looking at the values, all objects fall within a range of 31 and 2452 pixels, with a single outlier at 51883 pixels. The average area is 1358 pixels, so again all the other objects fall well below the threshold value for a hole.

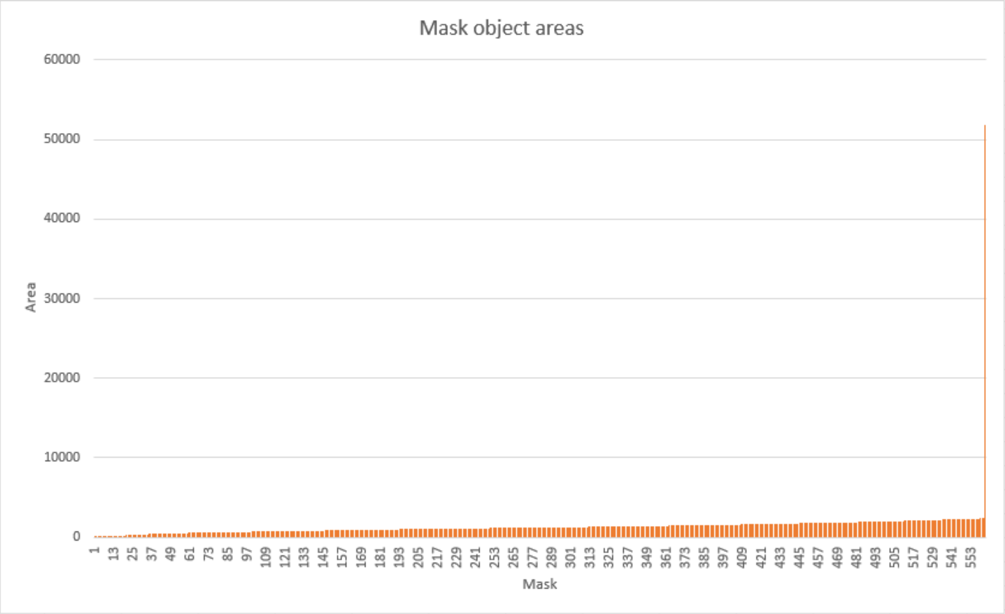


Figure 4.46: Example 2: Object areas in ascending order

Example 3

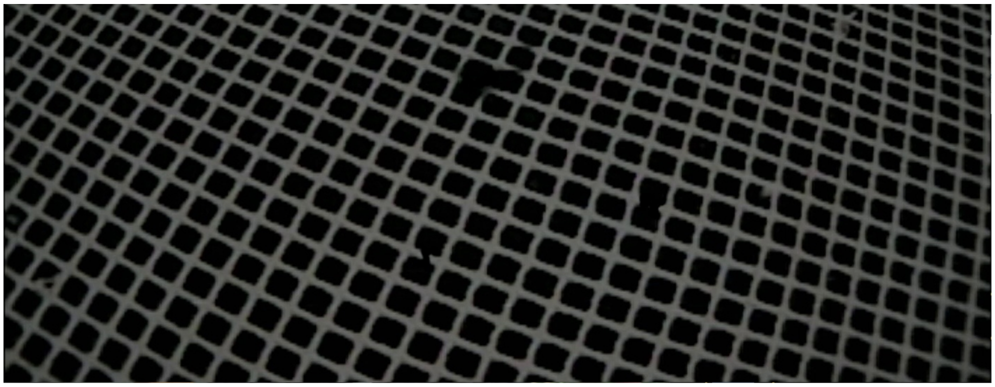


Figure 4.47: Example 3 - small holes

Example 3 (4.47) is a regular image of the net with 3 holes added by the group for testing purposes. Picture showing three defects in the net: one consisting of roughly three masks, one consisting of two masks with the net missing in between, and one consists of two masks where the line between has been "snipped".

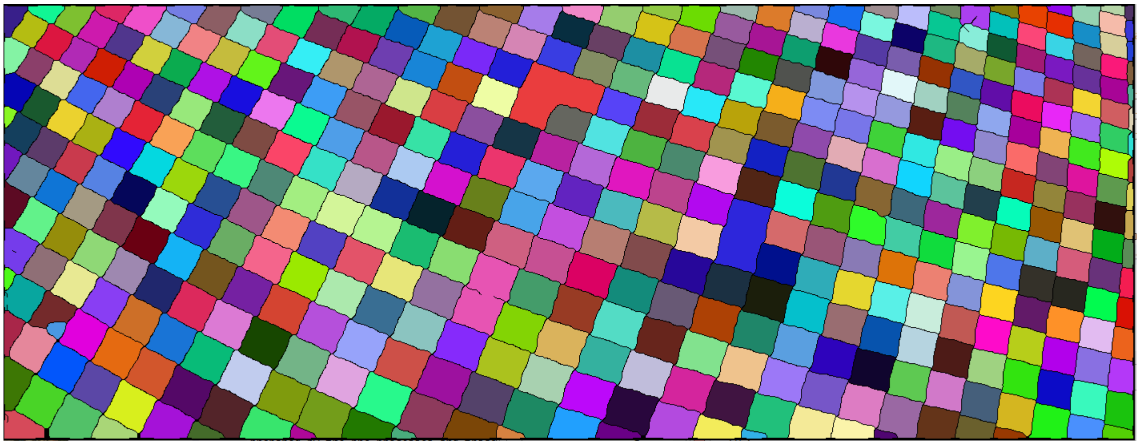


Figure 4.48: Example 3: Skeleton with objects randomly coloured

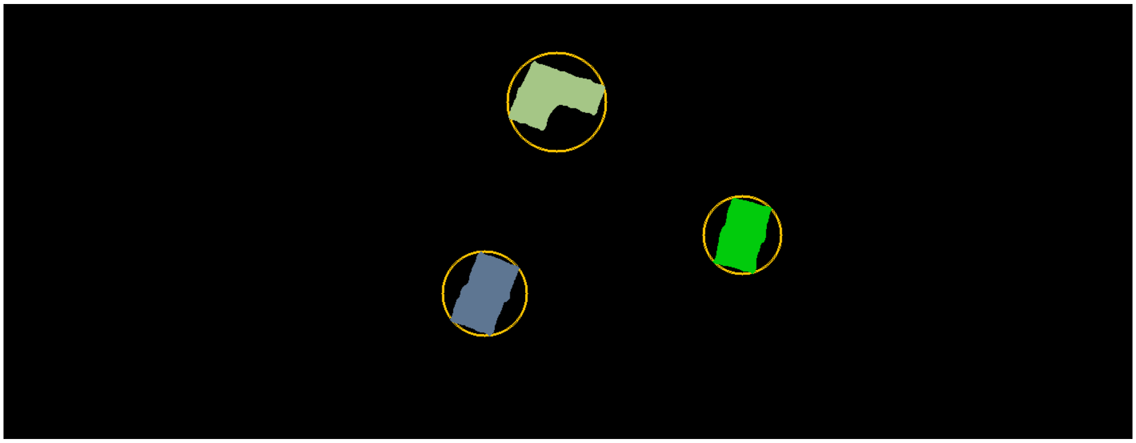


Figure 4.49: Example 3: results

Figure 4.49 shows that three holes are detected. Looking at the object area values, figure 4.50, they all fall within a range of 155 and 2932 pixels, with three outliers at 4329, 5026 and 5601. The average value is 1703 pixels, so all the other contours fall well within the threshold. This example shows that this test can detect with great accuracy on high quality images. In this example, the setting value for the analysis is set to 2. High image quality is required to be able to use such a low setting without getting many false hits.

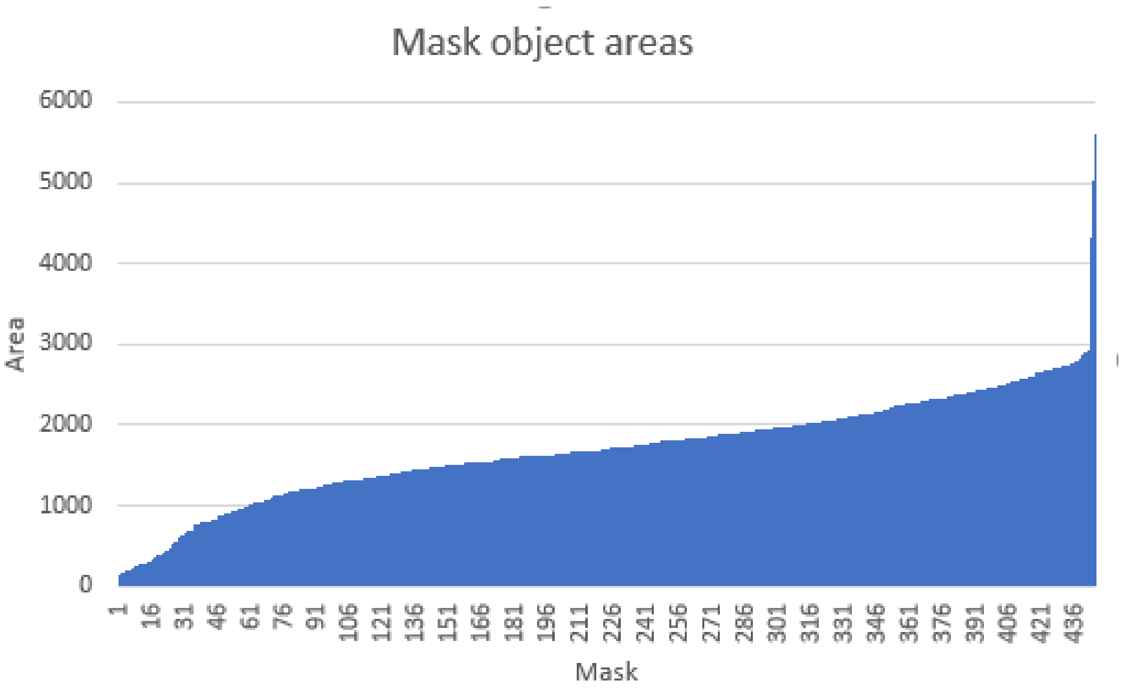


Figure 4.50: Example 3: Object areas in ascending order

Large Objects Analysis

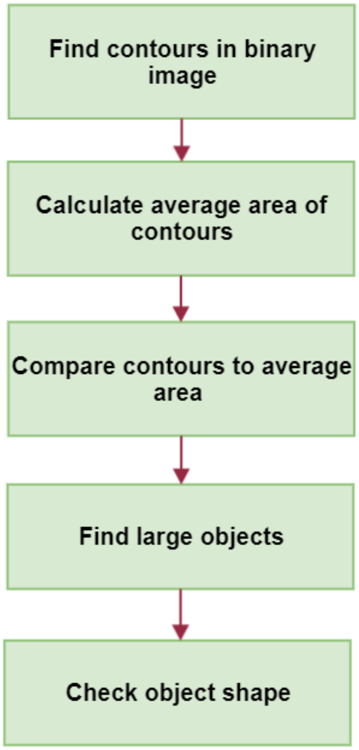

Find contours in binary image
Calculate average area of contours
Compare contours to average area
Find large objects
Check object shape

Figure 4.51: Diagram showing main tasks in the Large Objects Analysis

After developing the large mask analysis, the group decided to use the same principle on the binary image. This should detect all the large objects in the net, including defects. After running some tests, the group found out that it worked as intended, below results are shown using the same images as above. Figure 4.31 shows the main steps that the large objects analysis performs.

Example 1

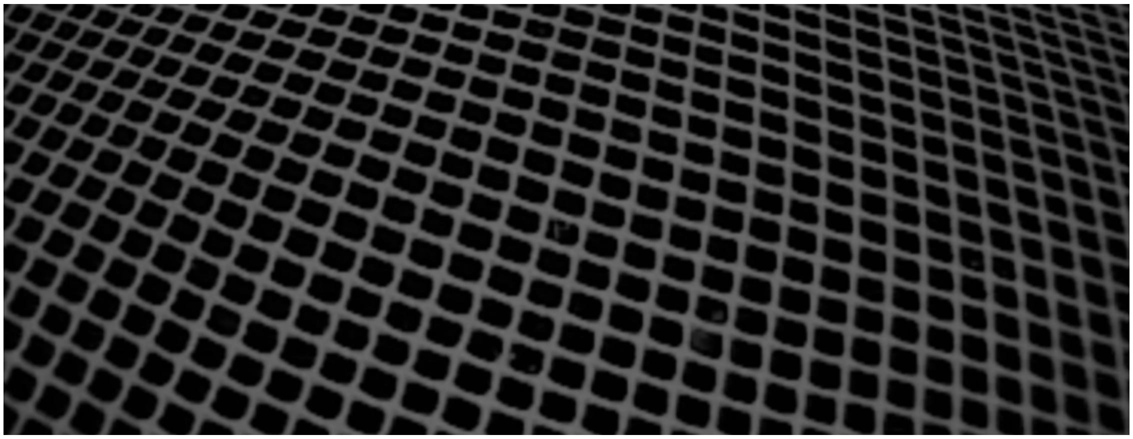


Figure 4.52: Example 1 - no holes

Example 1 (4.52), is a typical image of the net without any holes. As seen on the picture, the net has shapes of regular sizes, and is at a slight angle

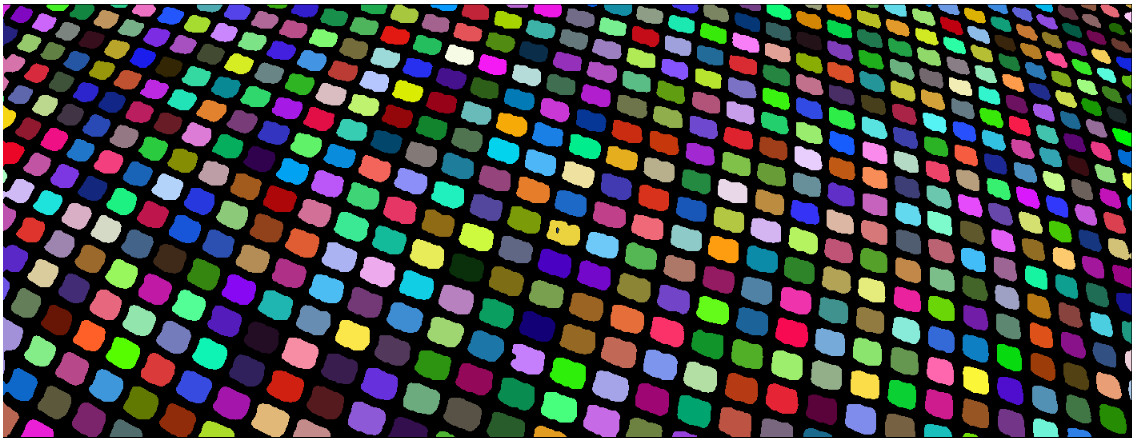


Figure 4.53: Example 1 - Objects randomly coloured

Figure 4.53 illustrates all the objects detected in the algorithm. Note that this picture is printed just for showing the contours. When the analysis runs this information is stored in a list, so figure 4.53 is just for illustration

Figure 4.54: Example 1 - no objects larger than the threshold

When the average area has been found, all the masks are compared to that area, the results of this can be seen in figure 4.54. The figure shows that no hits were found in example one.

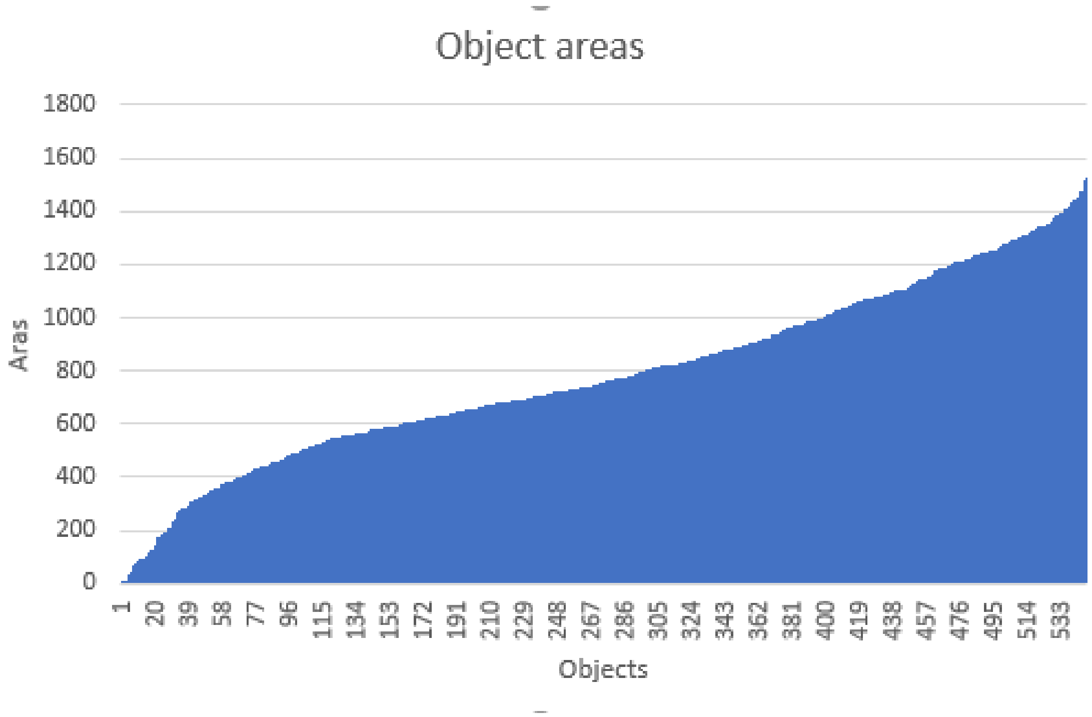


Figure 4.55: Example 1 - Object areas in ascending order

Figure 4.55 shows that no holes are detected . Looking at the object area values , they all fall within a range of 40 and 1527 pixels. The average value is 793 pixels, so all objects are below the threshold value for a hole, which in this example is set to three times the average object area.

Example 2

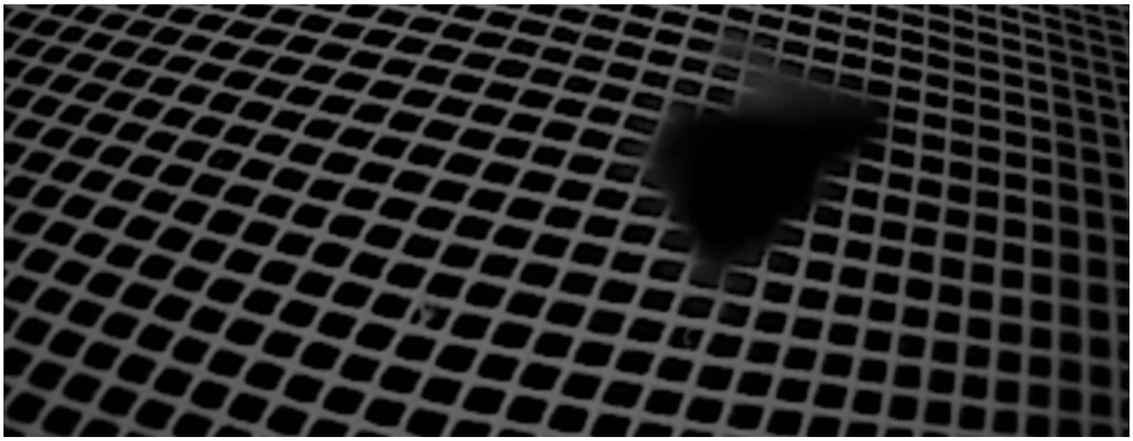


Figure 4.56: Example 2 - no holes

Example 2 (4.56) is an image used to simulate a hole. It is not actually a hole, but rather a small piece of debris right in front of the camera that blocks out a small region of the image. A hole may look very similar to this picture, so the picture is a good example.

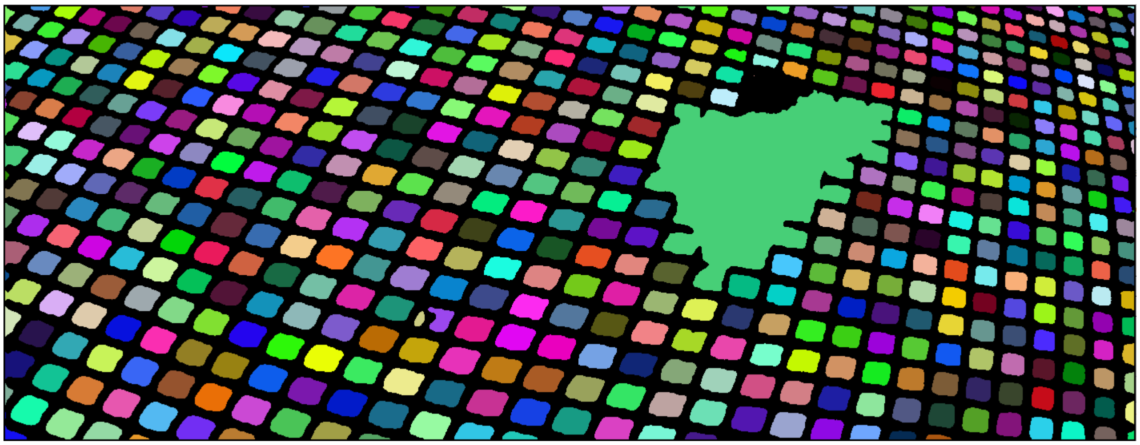


Figure 4.57: Example 2 - Objects randomly coloured

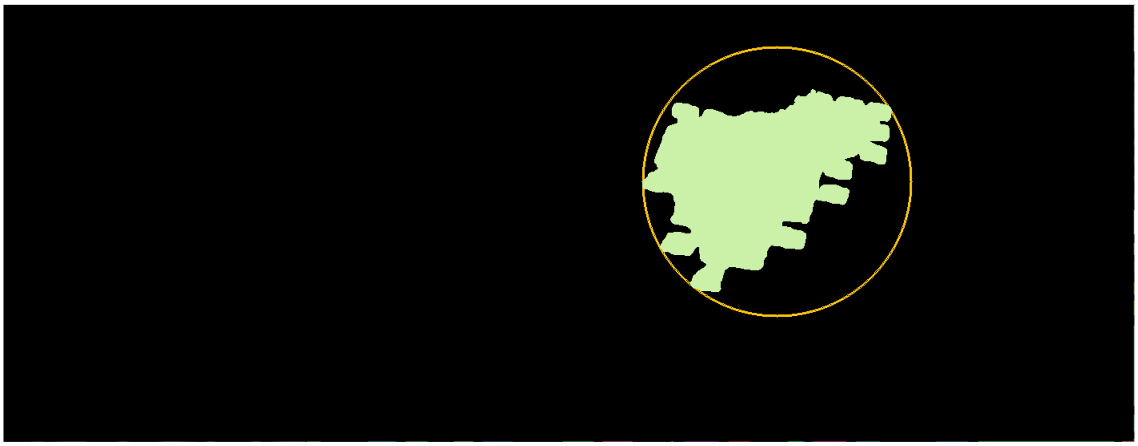


Figure 4.58: Example 2 - one region larger than the threshold

Figure 4.58 shows that one hole is detected. Looking at the values in figure 4.59, all objects fall within a range of 47 and 1431 pixels, with a single outlier at 43673 pixels. The average area is 782 pixels, so again all the other objects fall well below the threshold value for a hole.

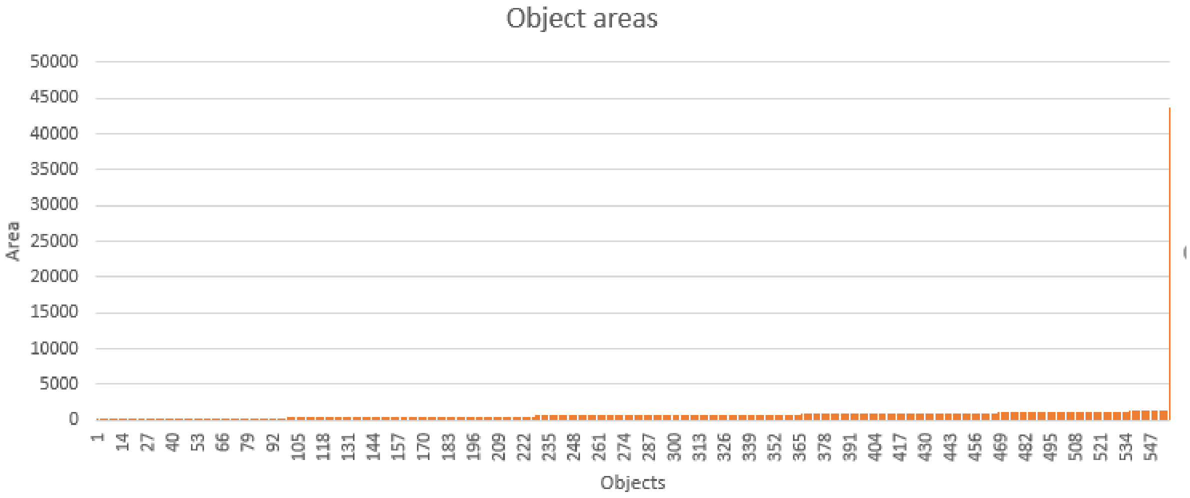


Figure 4.59: Example 2 - Object areas in ascending order

Example 3

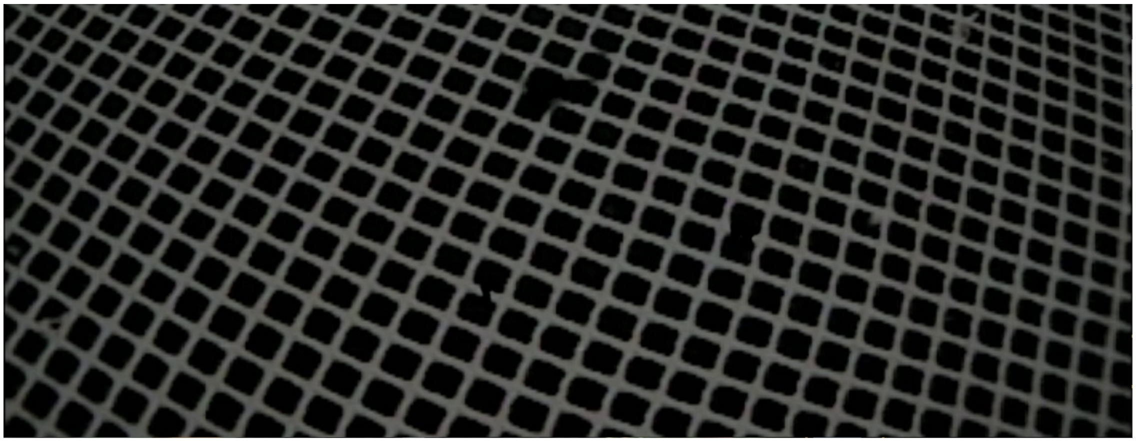


Figure 4.60: Example 3 - small holes

Example 3 (4.60) is a regular image of the net with 3 holes added by the group for testing purposes. Picture showing three defects in the net: one consisting of roughly three masks, one consisting of two masks with the net missing in between, and one consisting of two masks where the line between has been "snipped".

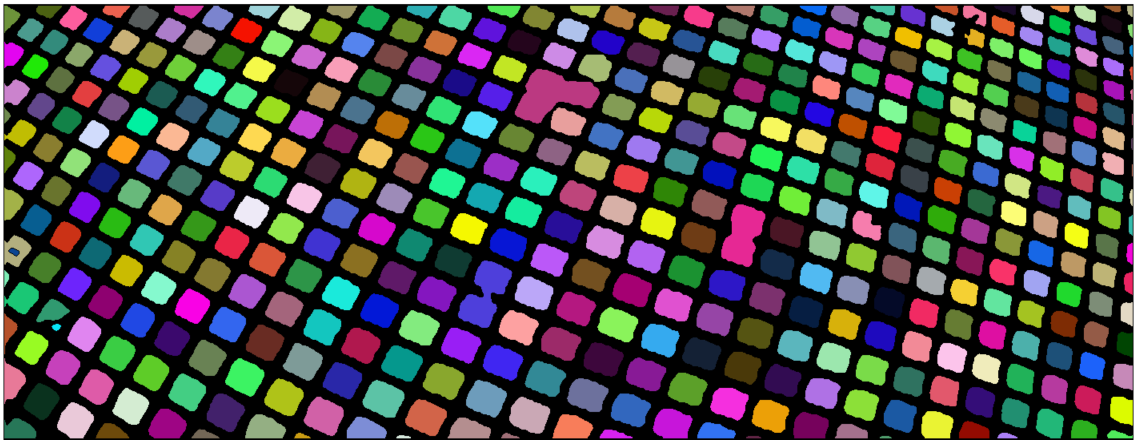


Figure 4.61: Example 3 - Objects randomly coloured

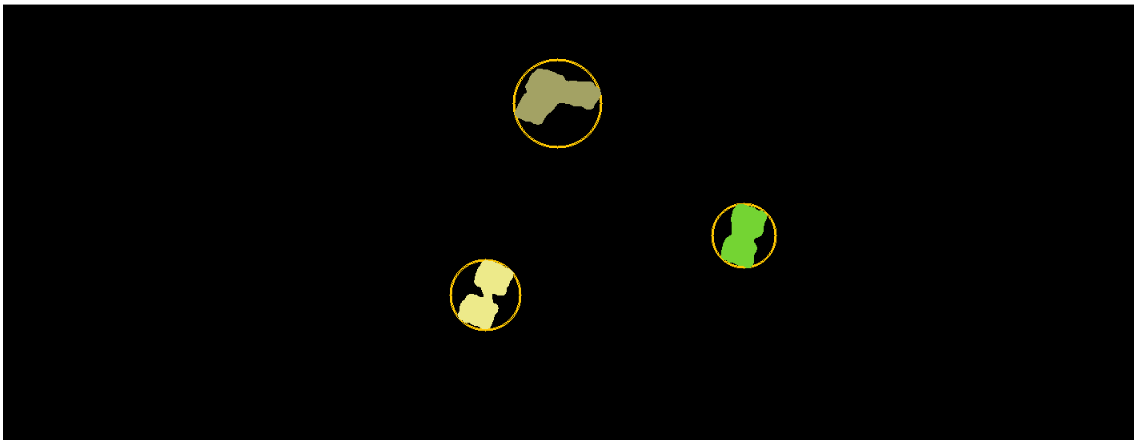


Figure 4.62: Example 3 - one region larger than the threshold

Figure 4.62 shows that three holes are detected. Looking at the object area values in figure 4.63, they all fall within a range of 42 and 1751 pixels, with three outliers at 2816, 3057 and 4068 pixels. The average value is 978 pixels. This example shows that this analysis can detect with great accuracy on high quality images. In this example, the setting value for the analyse is set to 2. High image quality is required to be able to use such a low setting without getting many false hits.

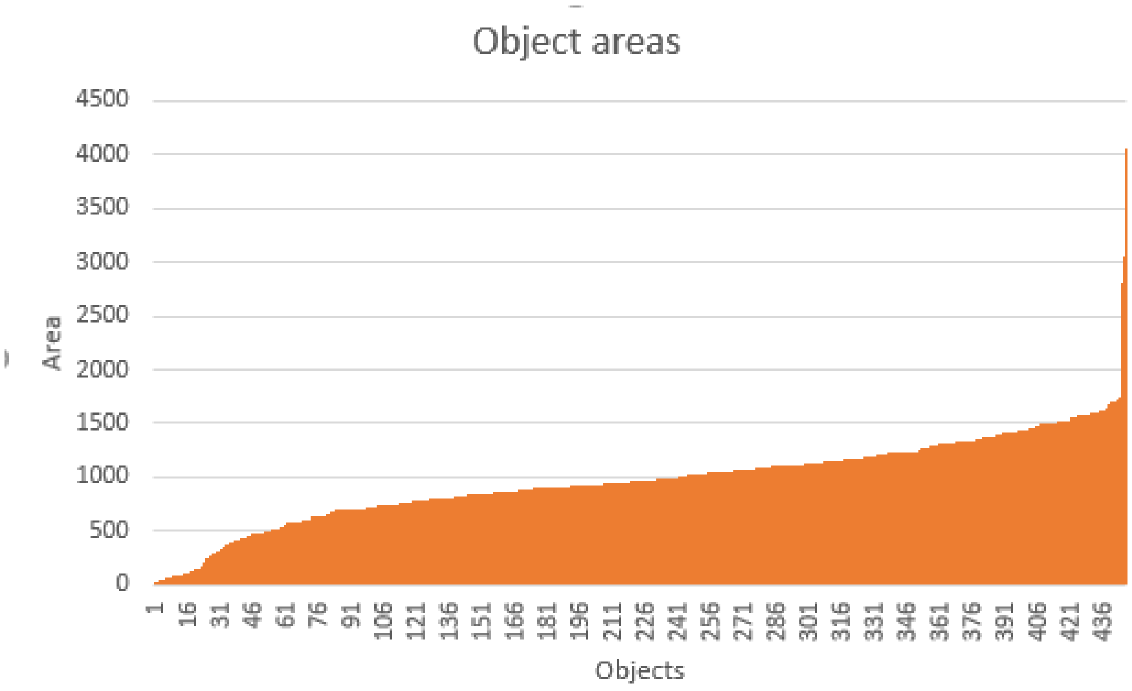


Figure 4.63: Example 3 - Object areas in ascending order

Knot Density Analysis

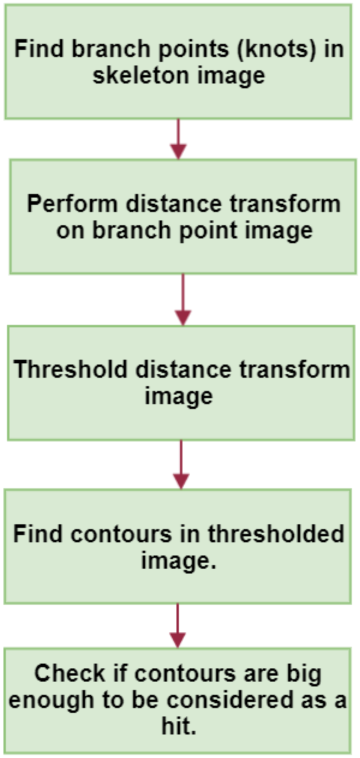

Find branch points (knots) in skeleton image
Perform distance transform on branch point image
Threshold distance transform image
Find contours in thresholded image.
Check if contours are big enough to be considered as a hit.

Figure 4.64: Diagram showing main tasks in the Knot Density Analysis

The knot density analysis algorithm uses the skeleton image to find all the branch points in the net, and then uses the distance transform to find the density of knots. Any areas with too low density is labeled a hit.

First the branch points are found and printed on an empty image of the same size as the skeleton image. The group had previous experience using MATLAB's branchpoint function, so the group first attempted implementing this method in Java. This looks for three or more 4-connected pixels around the middle pixel (2.1.4). The nets are often irregular and at a strange angle to the camera, so using a 3x3 neighborhood may not be accurate. The group experimented on making an own implementation and came up with a method using a 5x5 neighborhood method. In this method we searched an outer 5x5 ring for 0 to 1, or black to white, transitions. Below are some comparisons between the two:

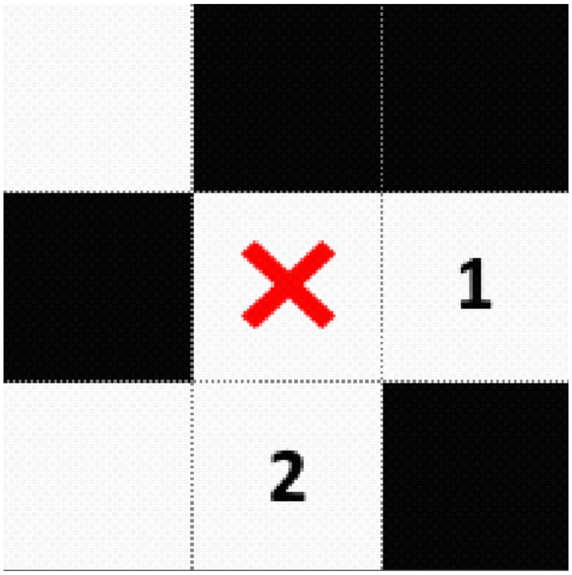


Figure 4.65: Picture showing a branchpoint example, attempted to detect using 3x3, 4-connected search

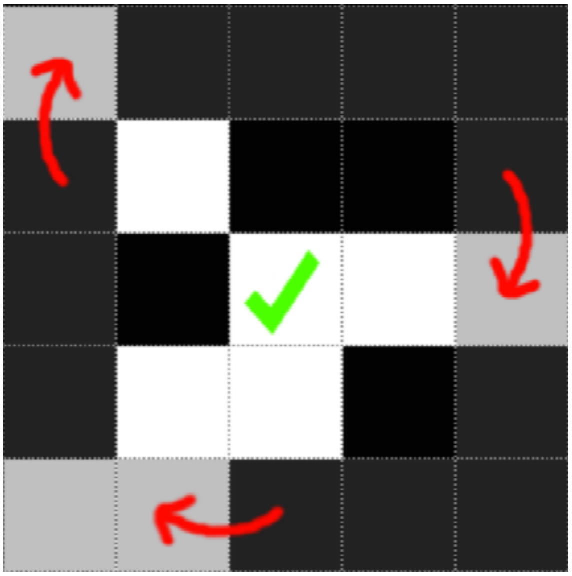


Figure 4.66: Picture showing a branchpoint example, attempted to detect using 5x5 outer ring, 0->1 transitions

Figure 4.65 shows that using a 3x3 4-connected search would fail in this example, whereas the 5x5 outer ring method (figure 4.66) which the group came up with, would correctly identify this as a branch point.

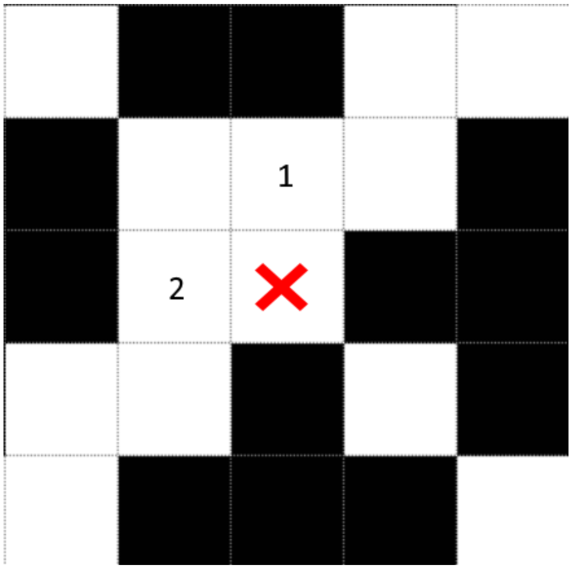


Figure 4.67: Picture showing a branchpoint example, attempted to detect using 3x3, 4-connected search

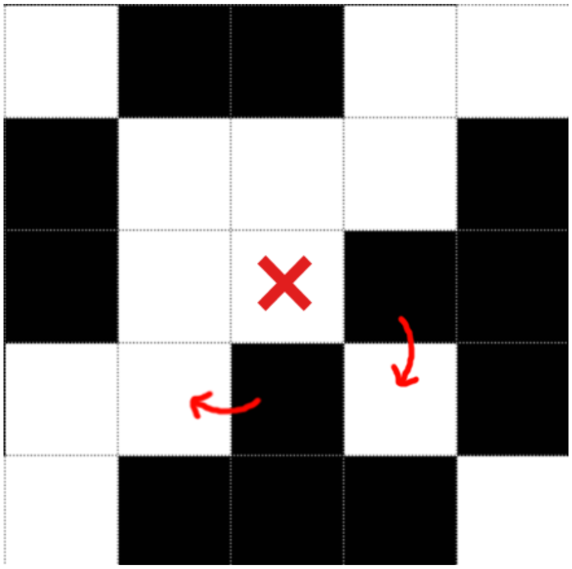


Figure 4.68: Picture showing a branchpoint example, attempted to detect using 3x3 ring, 0->1 transitions

The group also tested using the 0->1 transition method in a 3x3 ring, but as can be seen on figure 4.68 and figure 4.67, neither the 0->1 transition method or the 4-connected method succeed, but the self-made 5x5 outer ring 0->1 transition method would identify four 0->1 transitions in this example, and then correctly identifying it as a branch point.

After detecting all the branch points, they are printed on an empty image. Below are two example showing this.

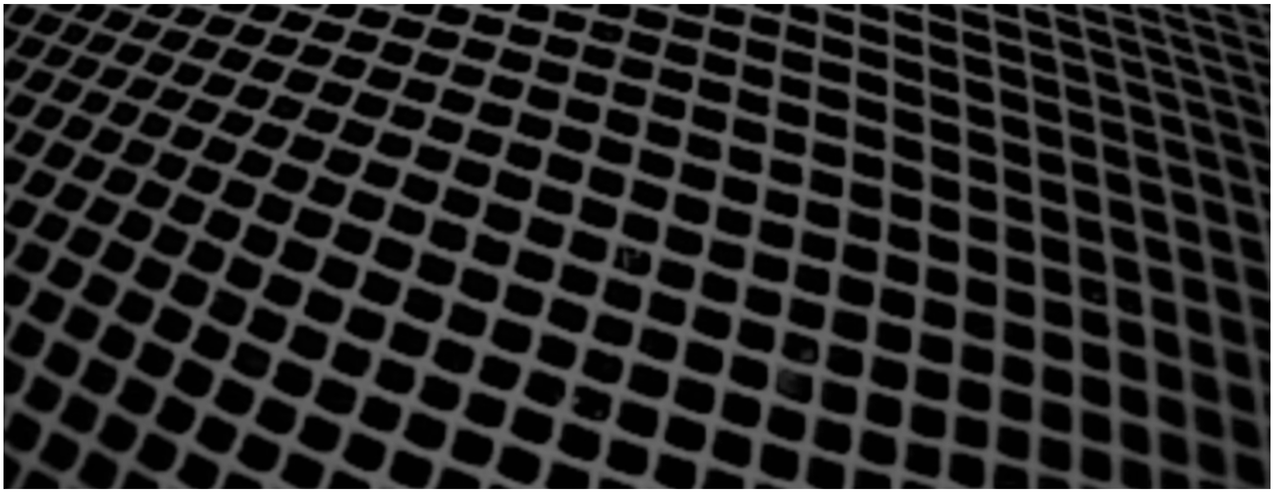


Figure 4.69: Picture showing knot density analysis example 1 - no holes

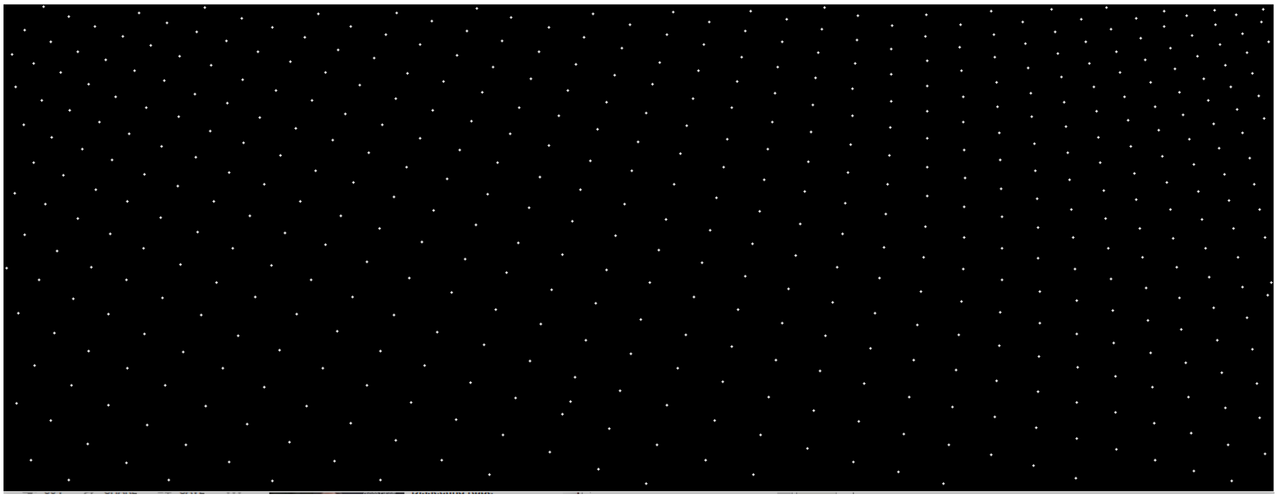


Figure 4.70: Picture showing knot density analysis example 1 - Branchpoints (knots)

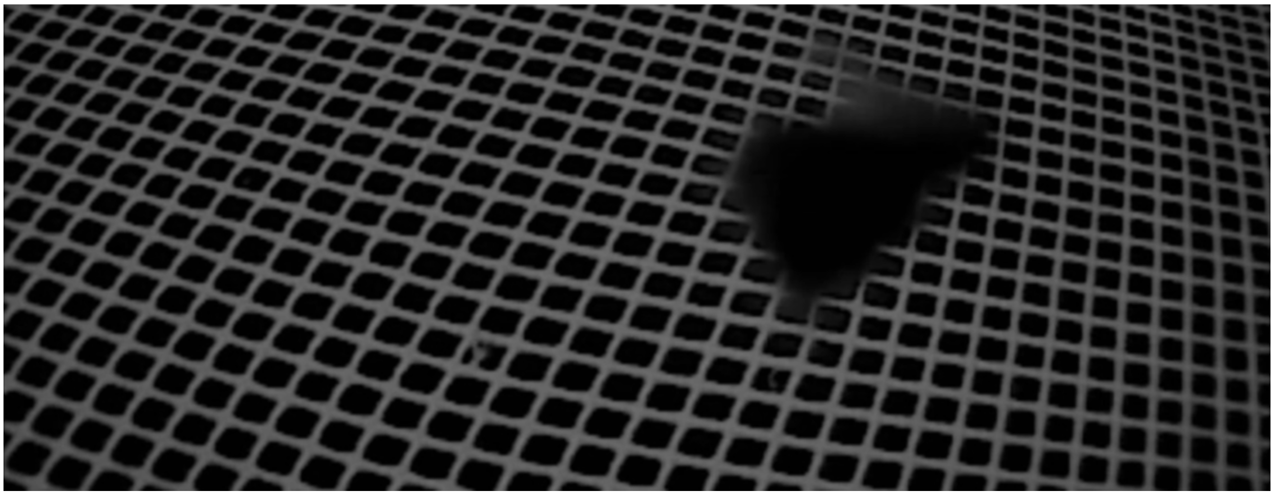


Figure 4.71: Picture showing knot density analysis example 2 - one large hole

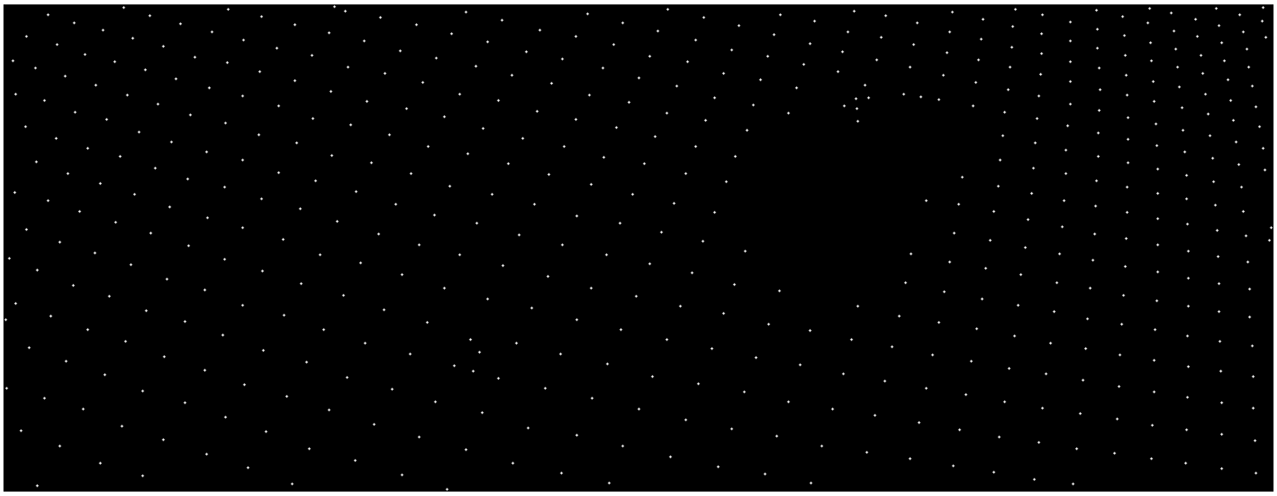


Figure 4.72: Picture showing knot density analysis example 2 - Branchpoints (knots)

After getting the branch points, OpenCV's distanceTransform() function was used to get a representation of the knot density as described in theory section 2.1.6.

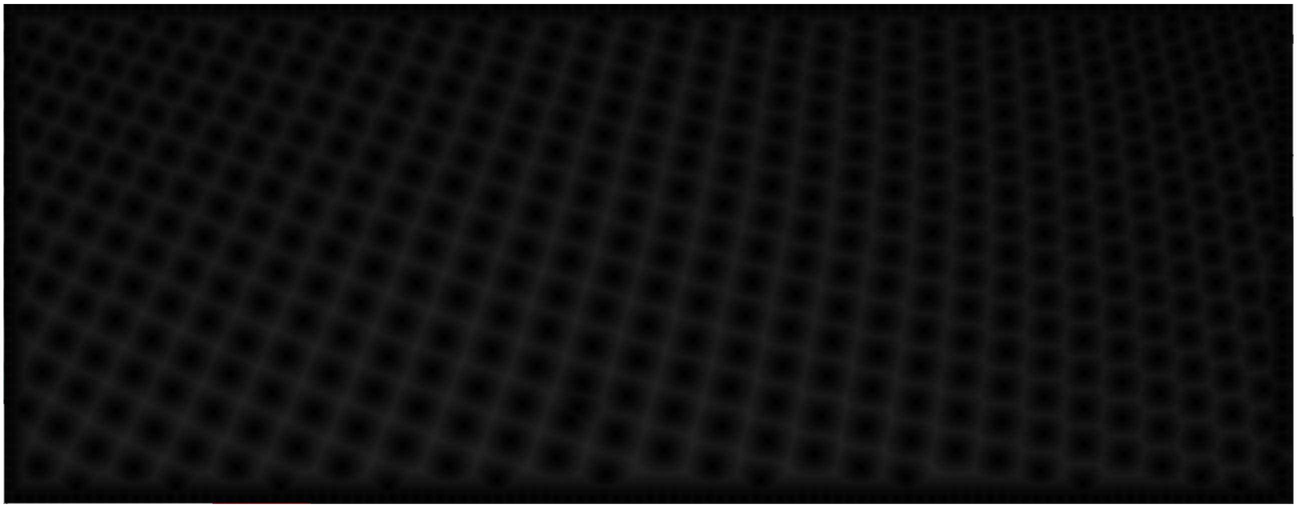


Figure 4.73: Picture showing knot density analysis example 1 - Distance transform

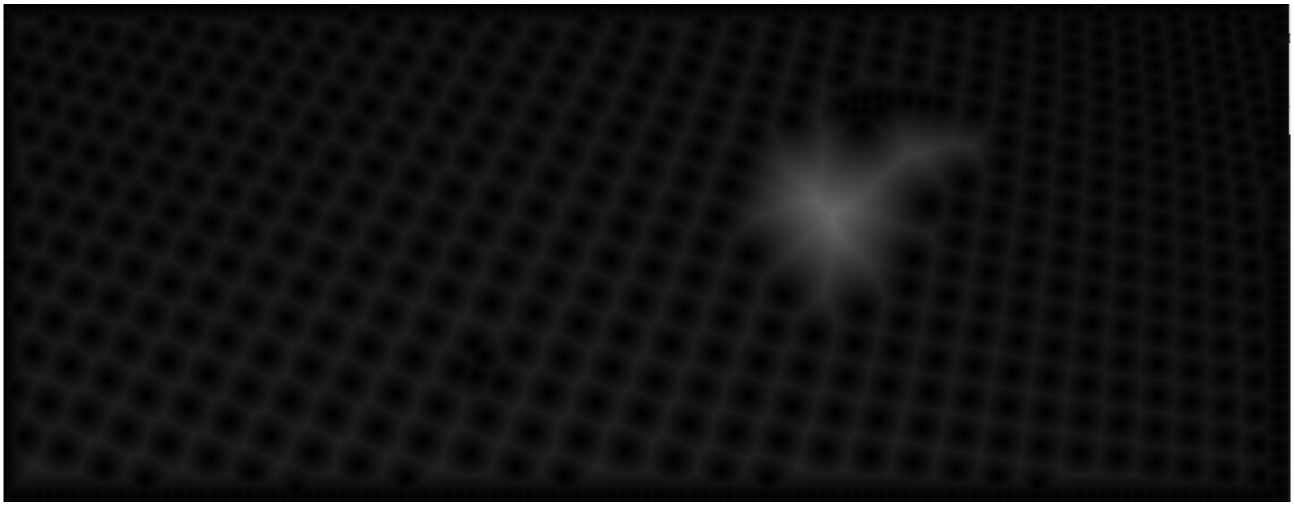


Figure 4.74: Picture showing knot density analysis example 2 - Distance transform

The group then used a binary threshold with a fixed threshold value to extract the brightest areas of the distance transform image 2.1.5. This results in an image with white objects, where size corresponds to distance from any branch point. Areas larger then a specified value are counted as hits. Below are the results from Example 1 (4.69) and Example 2 (4.71):

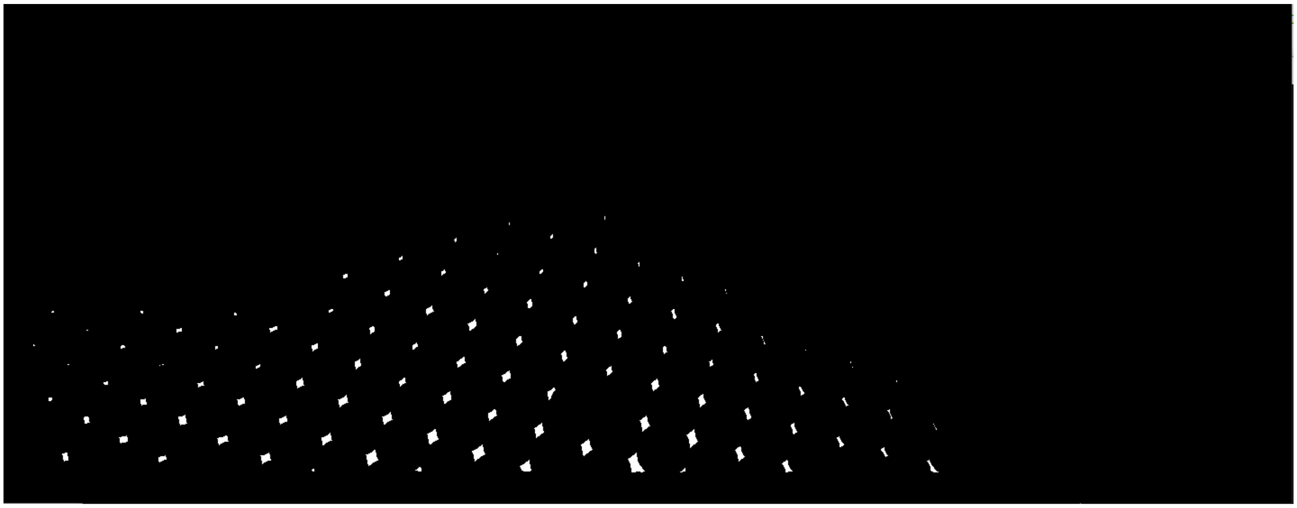


Figure 4.75: Picture showing knot density analysis example 1 - Results

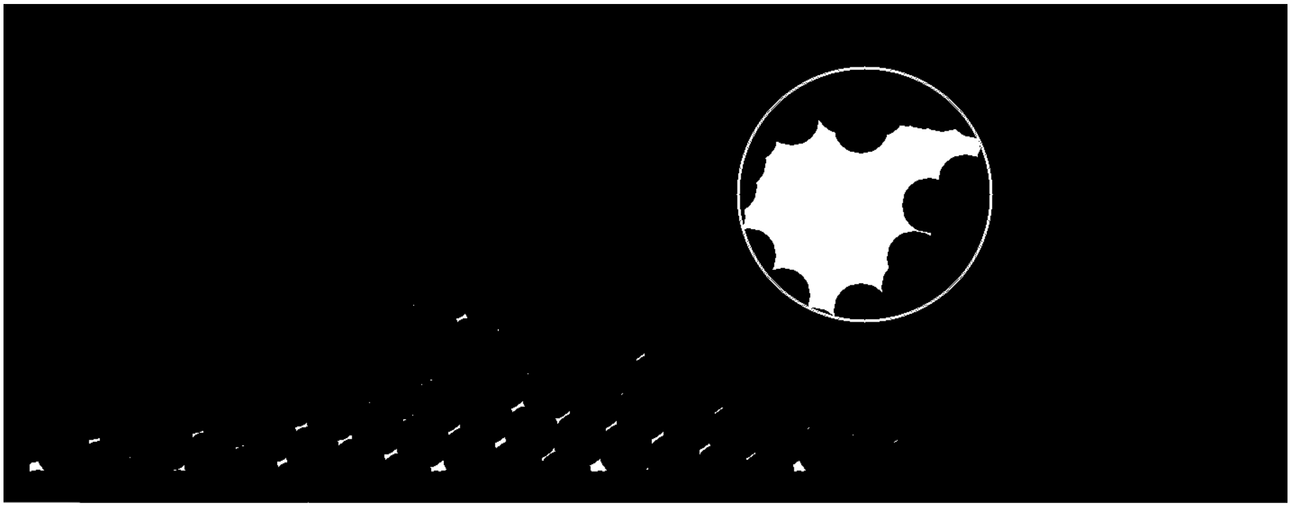


Figure 4.76: Picture showing knot density analysis example 2 - Results

Figure: 4.76 shows that one hit is detected and marked with a circle.

How to consider an object a hit

The Knots analysis produces a result image with a black background. Any hits appears as white objects in front of the black background. If the result image contains any white objects, the next and final part of the knot analysis is to consider whether these white objects are large enough to be considered a hit. To do this, the group has chosen to compare the area of each object with a "minimum object area" parameter (hereby referred to as “MOA”). For an object to be considered a hit, it must have a larger area value than the MOA parameter. The calculation of the MOA parameter is quite complex. This is because the parameter includes the possibility of manually adjusting sensitivity and automatic correction for both image resolution, mask size in the net and the distance from camera to net.

The MOA parameter is calculated using the following three individual parameters, in addition some hard-coded ratios is used to bring these three values together to form an appropriate MOA value.

- Total number of pixels in the analysed image.
- The density of knots found in the analysed image
- A multiplier value that the user can adjust manually

The number of total pixels in the analysed image must be included in the calculation to automatically correct for changes in the video-resolution. By taking this into account, it is possible to change the camera on the manta to one with a lower or higher resolution without making changes in the program.

The density of knots found in the analysed image must be taken into account to automatically correct for mask size in the grid and the distance from the camera to the grid. By utilizing the density of knots, the MOA parameter will change in relation to the average density of knots in the entire analysed image. If the camera moves further away from the net, the density of the knots will change and make the MOA variable decrease. Similarly, MOA will increase if the mask size in the net increases. The density of knots is calculated by using the ratio between the total number of pixels in the image and the number of knots found in the image. If only half of an image is analyzed (due to poor quality of the rest), this is also taken into account in the calculation of the knot density.

The multiplier value is used to give the user an opportunity to manually adjust the sensitivity of the knot test. This value is the one that is used in the settings tab of the GUI. This value is multiplied by a ratio that makes it fit to the adjustment scale between 1 and 10.

4.2 Test Run

The test runs have been done on a Windows 10 machine with the following specs:

- **Operating System:** Windows 10 Home, 64-bit
- **Processor:** Interl(R) Core(TM) i5-4670K CPU @ 3.40Ghz
- **Memory:** DDR3 12GB
- **Java:** 1.8.0

4.2.1 Speed Test

After implementing multithreading a series of test runs were done to test the speed improvement. A section of the video containing 1071 frames was tested, 3 times for each test to account for outside factors. On the test machine used, 4 proved to be the optimal number of threads, but on stronger computers this could be increased to speed up the process further.

Threading	**Frames**	**Time**	**Time per Frame (ms)**
Single	1071	12m43s	712ms
Single	1071	13m13s	740ms
Single	1071	13m27s	754ms
Multi (4)	1071	3m46s	211ms
Multi (4)	1071	4m13s	236ms
Multi (4)	1071	3m47s	212ms

Table 4.2: Speed-test results singlethread vs multithreaded software

4.2.2 Test Analysis

To test the software a section of the video was analysed. The portion is a 5 minute and 34 seconds long section of the video, with time between frames analysed set to 0.5 seconds. Of the 689 frames analysed, 157 were discarded by the quality checker. This shows one of the problems with the irregular nature of the video. Of the analysed frames, 41 hits were detected. Looking through the results, many can be explained by fouling casting shadows on the net. This can be seen in the first example.

Start time	00:15:02
End time	00:20:36
Number of analysed frames	689
Number of discarded frames	157
Percentage analysed of analysed frames	45%
Number of hits	41

Table 4.3: Statistics from the analysis test-run

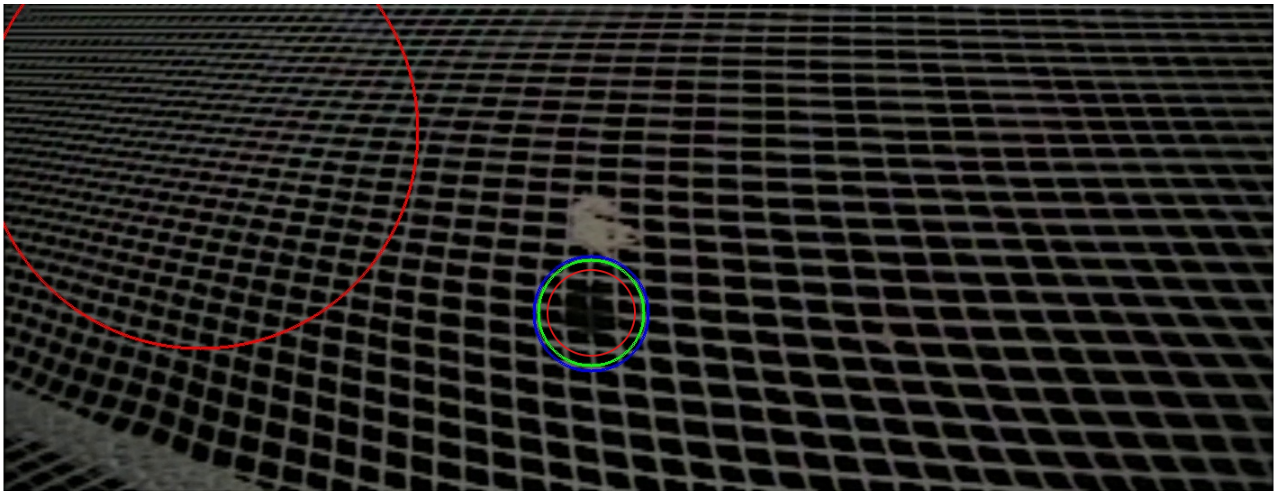


Figure 4.77: Example result from the test analyse

The shadowed part is detected by all three methods. This is a good hit, as this shadow looks similar to a real hole. The knot density analysis has also detected a large hole were clearly there are none. This can be explained by the nets in the mask being hard to distinguish, resulting in vertical lines not being detected. As can be seen in Figure 4.78, this results in long horizontal masks, with no branch points. In the knot density analysis this will therefore result in a large hole.

Figure 4.79 shows how the large mask analysis sees the shadow. This is clearly bigger than the other masks, thus resulting in a hit. Similarly, the object found by the large object analysis is also much bigger than the other objects.

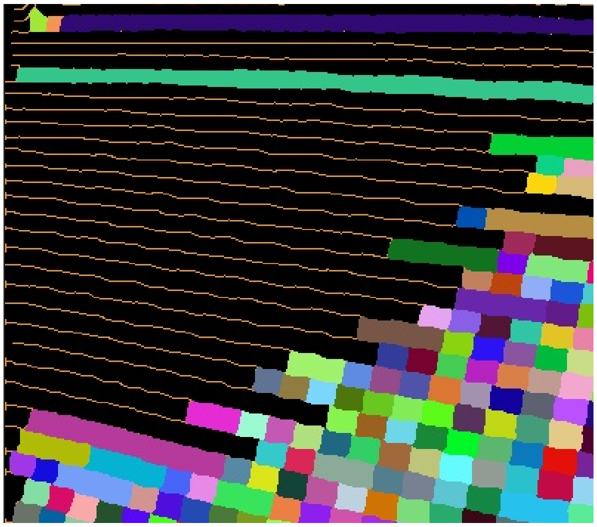


Figure 4.78: Showing upper right corner of figure 4.77

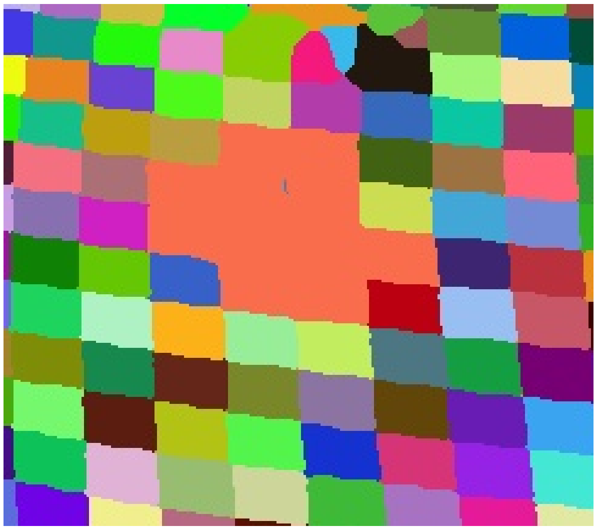


Figure 4.79: Showing the middle section of figure 4.77

Another example from the analysis is Figure 4.80. This image illustrates how small holes can be detected. It is unclear from the picture if this is caused by an actual hole, if it's a poor image or if there's a small shadow, but as the image is marked with a time stamp this can be double-checked against the video.

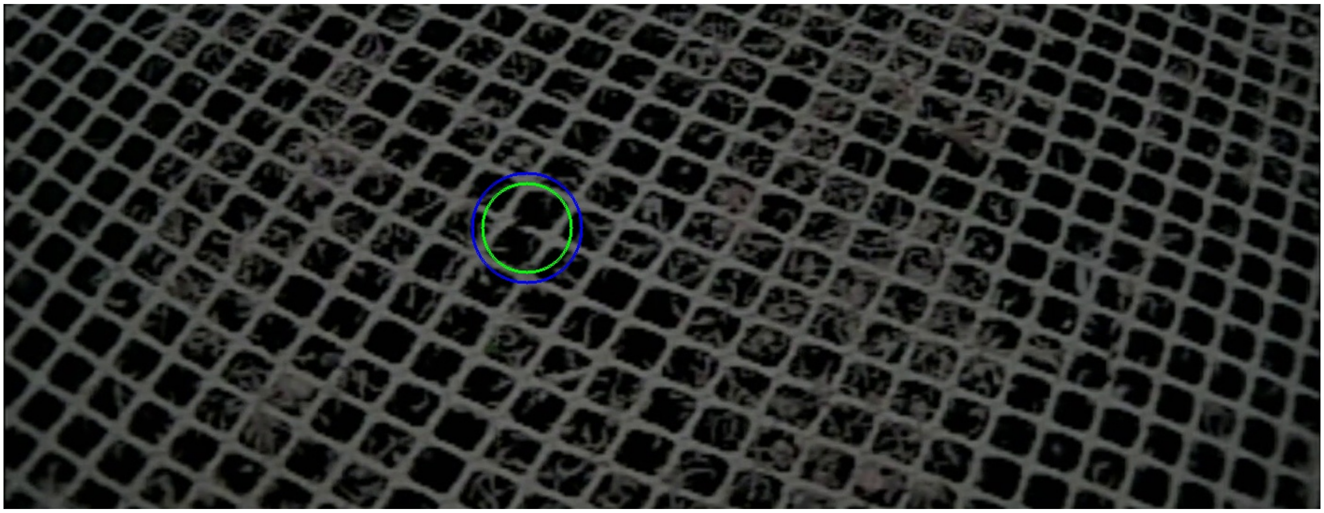


Figure 4.80: Example result

4.3 Software Architecture

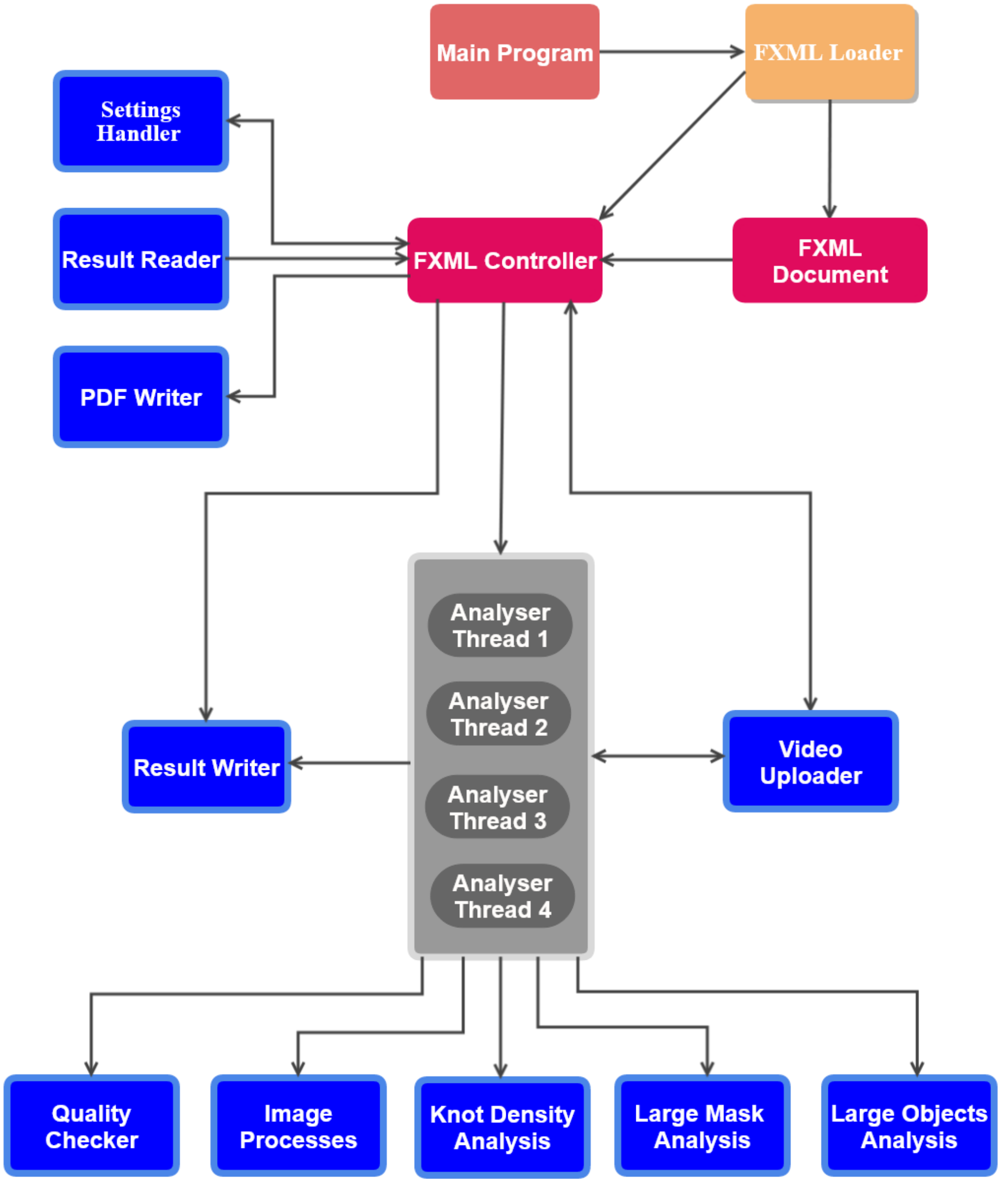


Figure 4.81: Class structure overview over the most important classes in the software.

4.3.1 Class overview

The java application has a total of 15 classes. More detailed description of each class, and the methods inside each class, is available in the javadoc.

- **Manta Program** class holds the main function that starts the application. The graphical user interface is created and presented to the user, and the Controller is created based on the FXML document.
- **FXML Document Controller** class handles all the user interactions by handling events from the GUI. This class is also responsible for updating the GUI.
- **Settings Handler** handles the XML file used for storing setting profiles. This includes taking care of all tasks related to creating, modifying, storing and retrieving profiles to and from the XML file. The class uses java JDOM Parser to handle the XML file.
- **Result Writer** class creates an XML file before each analysis and is responsible for writing each new hit to this file. In addition, it stores statistics and creates an XML file containing setting information and statistics after each analysis. The class uses java JDOM Parser to create the XML files.
- **Result Reader** class reads information from the XML files created by the Result Writer class. The class uses java JDOM Parser to read the XML files. This information is used to present the results in the GUI for the user. The PDF Writer class also uses this information to create the PDF report.
- **PDF Writer** class handles all tasks related to the production of the PDF analysis report, containing results and statistics from an analysis. The class uses Apache PDF Box 3.2.6. to write the PDF document.
- **Settings** class stores the current analysis settings. This class is used to send the setting information to the Analyser Threads, Result Writer and PDF Writer.
- **Video Uploader** class is responsible for uploading and holding the video data used during an analysis. It is responsible for dividing the video into frames that can be fetched by

the Analyser Threads during an analysis. The class is designed to allow multiple Analyser Threads to fetch images at the same time.

- **Analyser Thread** class is responsible for processing and analysing video frames. This involves fetching new frames from the Video Uploader, making sure that the different analysis steps are performed on each frame, and sending the results from each frame to the Result Writer. The class is designed to allow multiple instances to run concurrently.
- **Custom Image** class stores an image and its id. This is used to send images from the Video Uploader to the Analyser Threads.
- **Quality Checker** class checks the quality of each frames and returns information of the frame quality. This information is used to select frames, or part of frames that should not be analysed.
- **Image Processes** contains methods for performing image pre-processes used to prepare a frame for further analysis.
- **Knot Density Analysis** class contains methods for performing the knot density analysis and returning the analysis results.
- **Large Mask Analysis** class contains methods for performing the large mask analysis and returning the analysis results.
- **Large Objects Analysis** class contains methods for performing the large objects analysis and returning the analysis results.

4.3.2 The analysis process of a frame

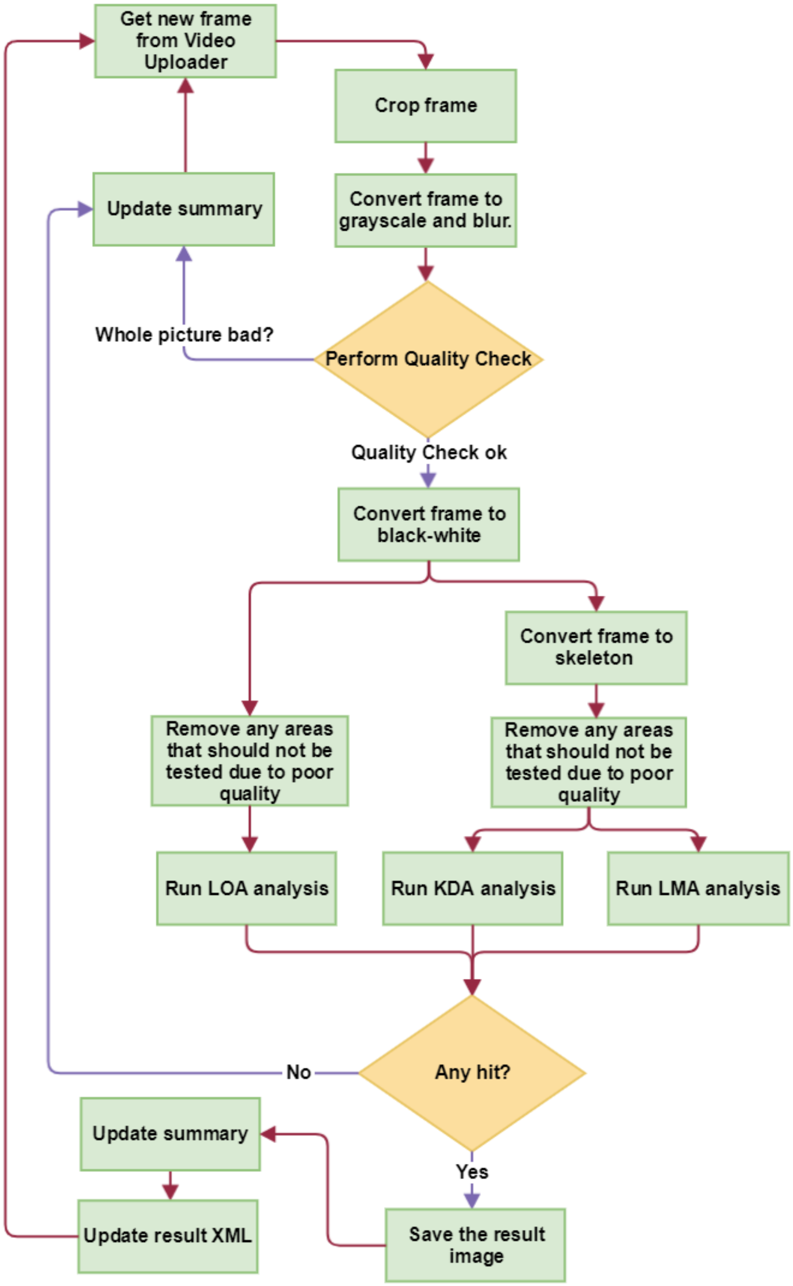

Get new frame from Video Uploader
Crop frame
Convert frame to grayscale and blur.
Update summary
Whole picture bad?
Perform Quality Check
Quality Check ok
Convert frame to black-white
Convert frame to skeleton
Remove any areas that should not be tested due to poor quality
Remove any areas that should not be tested due to poor quality
Run LOA analysis
Run KDA analysis
Run LMA analysis
No
Any hit?
Update summary
Yes
Update result XML
Save the result image

Figure 4.82: The analysis process inside a thread

When the java application is launched, the main class creates the FXML Loader and the FXML Document controller. The FXML Loader creates the GUI based on the FXML Document. The GUI is visible for the user and presents 4 tabs. The user can then upload a video and start an analysis.

When an analysis is started, the FXML Document Controller creates and starts four instances of the Analyser Thread class. The settings used are retrieved from the GUI. Each Analyser Thread then starts requesting frames to analyse from the Video Uploader.

Figure 4.82 shows a good overview of the analysis process inside each thread. As soon as the thread retrieves a frame, the frame is cropped. This is done because the camera angle on the manta is adjusted so that only the lowest part of the picture is relevant for this analysis. Next step in the analysis is to evaluate the quality of the frame. The Quality Check divides the frame into multiple (5) vertical sections. The Quality Check returns a boolean value for each of the sections, where "true" means the section should be analysed and "false" that it should be discarded.

After the quality check, functions inside the Image Processes class is used to pre-process the frame before the main analysis is done by the three analysis methods (Knots Detection Analysis, Large Mask Analysis, and Large Objects Analysis.) The results are sent to the ResultWriter class. If any hits are detected on a frame, the frame is saved in an image folder, and the analysis result is written to the result XML file, and included in the summary. If no hit is detected, the result is only included in the summary.

4.3.3 Communication between classes:

Loose coupling

To ensure a good software structure, the group has used known guidelines for object-oriented programming. To achieve loose coupling 2.3.1, the communication between classes has been carefully considered. This is clearly visible in the structure overview 4.81. In the overview you can see that all communication goes through a limited, neat and well-transparent pattern. For example, only the Analyzer Threads create and communicate with the five classes that perform the image processing (Quality Controller, Image Processes, Knots Detection Analysis, Large Mask Analysis, and Large Regions Analysis.) Similarly, the Settings Handler, Result Reader and PDF Writer only communicates with the FXML Controller. This makes the entire software structure very clear, which makes it easier for others to familiarize themselves with how the software works. It also makes it easy to make changes to a class without having to change the whole software. Only the classes a class communicates with needs to be considered when making a change.

Large degree of Encapsulation

Large degree of encapsulation 2.3.3 is used as a tool for reducing coupling. In the software all communication, such as distribution of information between classes, takes place through calls to methods that return the requested information. Thus, private fields can only be used within a class. This contributes to looser links, but also helps to make the software safer to use. For example, the chances of fields being changed by a mistake are significantly reduced. This also makes it possible to change where the information inside a class is stored without having to make changes in other classes.

4.3.4 Distribution of tasks between classes

High degree of cohesion

The division of the classes is also well thought out. Each class is responsible for their own carefully limited tasks, in other words they have a high degree of cohesion 2.3.2. The software could have been written using fewer classes. For example, several of the image processing methods could be located in one single class. However, this would have made the program less flexible. By dividing the different image processing methods into several classes, the program becomes more modular. This has many benefits, for example that it becomes very easy to copy a class for use in another code. Another advantage is that the program becomes easier to change. For example, one can easily make changes to the Knot Density Analysis class or one of the other image processing classes without affecting the rest of the program. How the rest of the software is built up is not important to the change made, as all other classes only relate to the methods within the Knot Density Analysis class.

All the classes also exhibit a high degree of cohesion. All the tasks within a class are limited and distributed among the classes' methods. For example, one can look at the Image Processes class that performs image pre-processing methods on frames. The responsibility of this class is to prepare frames for the three analysis algorithms, which all require different starting frames. Therefore, the tasks within the Image Processes class are carefully divided into methods. In this way, one can customize the processing of frames for each of the three analysis methods just by calling different methods. In this way the code can be re-use to avoid code duplication. Each of the image processing functions used by Image Processes is only listed once.

Distribution of tasks by responsibility

The tasks are distributed among the different classes on the basis of each class's specific area of responsibility. 2.3.4 For example, the Settings Handler class has responsibility for managing the settings stored in an XML file. Therefore, the software is designed so that only the Settings Handler class handles the XML file containing settings. Although many other classes use the settings information, only the Settings Handler class can retrieve and edit this information from the XML document. All the other classes that need to handle this information must use the settings from the Settings Handler class. This is also a great advantage if the program is to be modified or rebuilt. If there are many classes that edit the XML document, it would provide a higher degree of coupling and more to take into account if a change should be done.

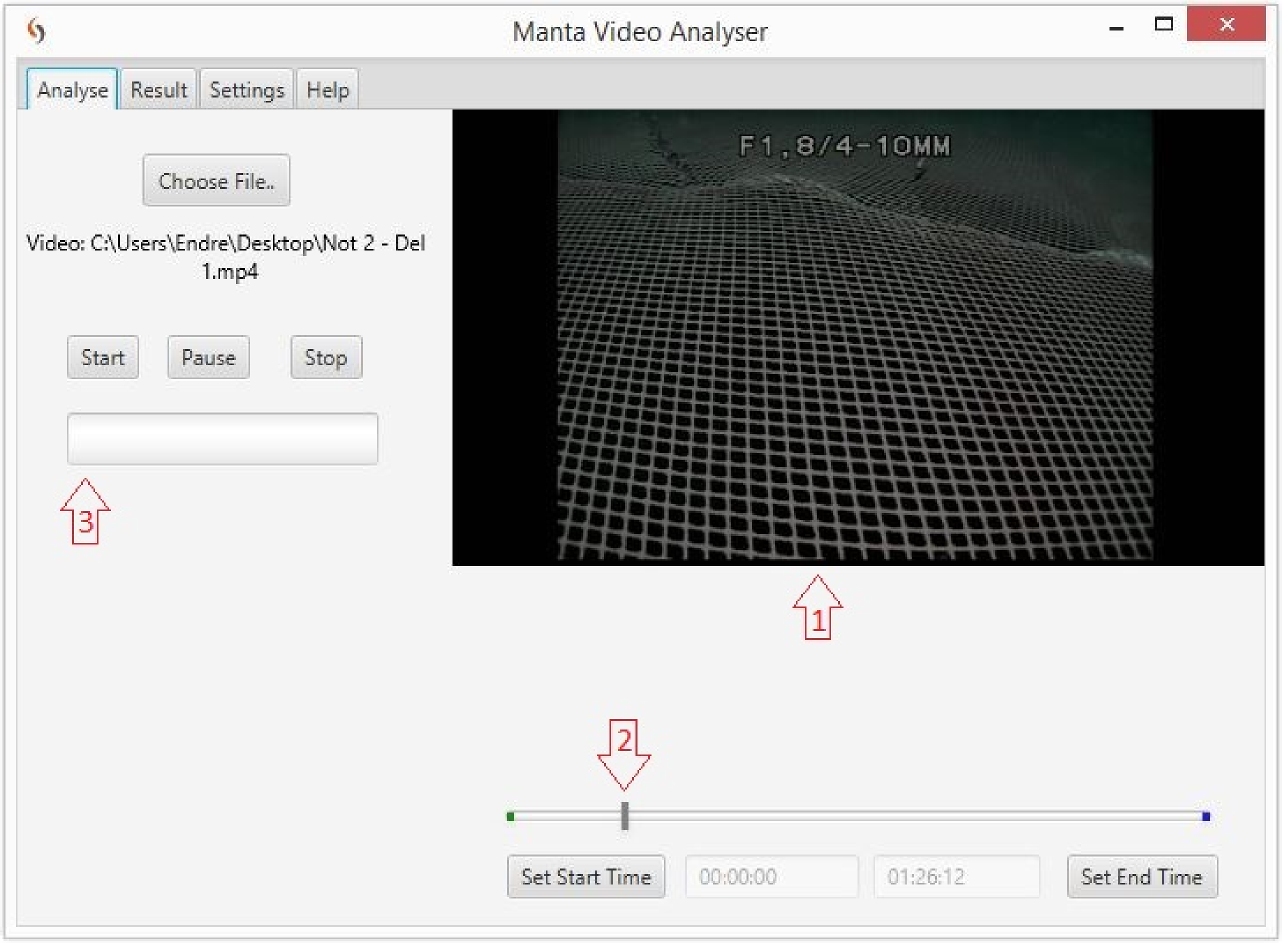
4.3.5 Multithreading

To increase the speed at which the program processes images, a multithreading approach has been chosen. Due to the fact that each image can be processed and analysed independently, the group decided to create a single class (AnalyserThread) to process and analyse an image and create multiple instances of this class. Each of the instances grabs images from a shared VideoUploader instance. To do this in a thread-safe way the getNextImage()-method in the VideoUploader is synchronized. This ensures that only one of the threads can ask for an image at the same time, which again insures that each image retrieved is a new image. Additionally, the results are all sent to another shared instance. This is also made thread-safe using synchronization.

All GUI updates has to be done on the main JavaFX thread, so to prevent the GUI from going unresponsive during the analysis, the AnalyserThreads are implemented using the JavaFX Task class. This ensures that the threads are run in the background, allowing GUI updates to be done on the JavaFX thread.

4.4 Graphical User Interface

4.4.1 Analysis Tab



Video: C:\Users\Endre\Desktop\Not 2 - Del 1.mp4

00:00:00

01:26:12

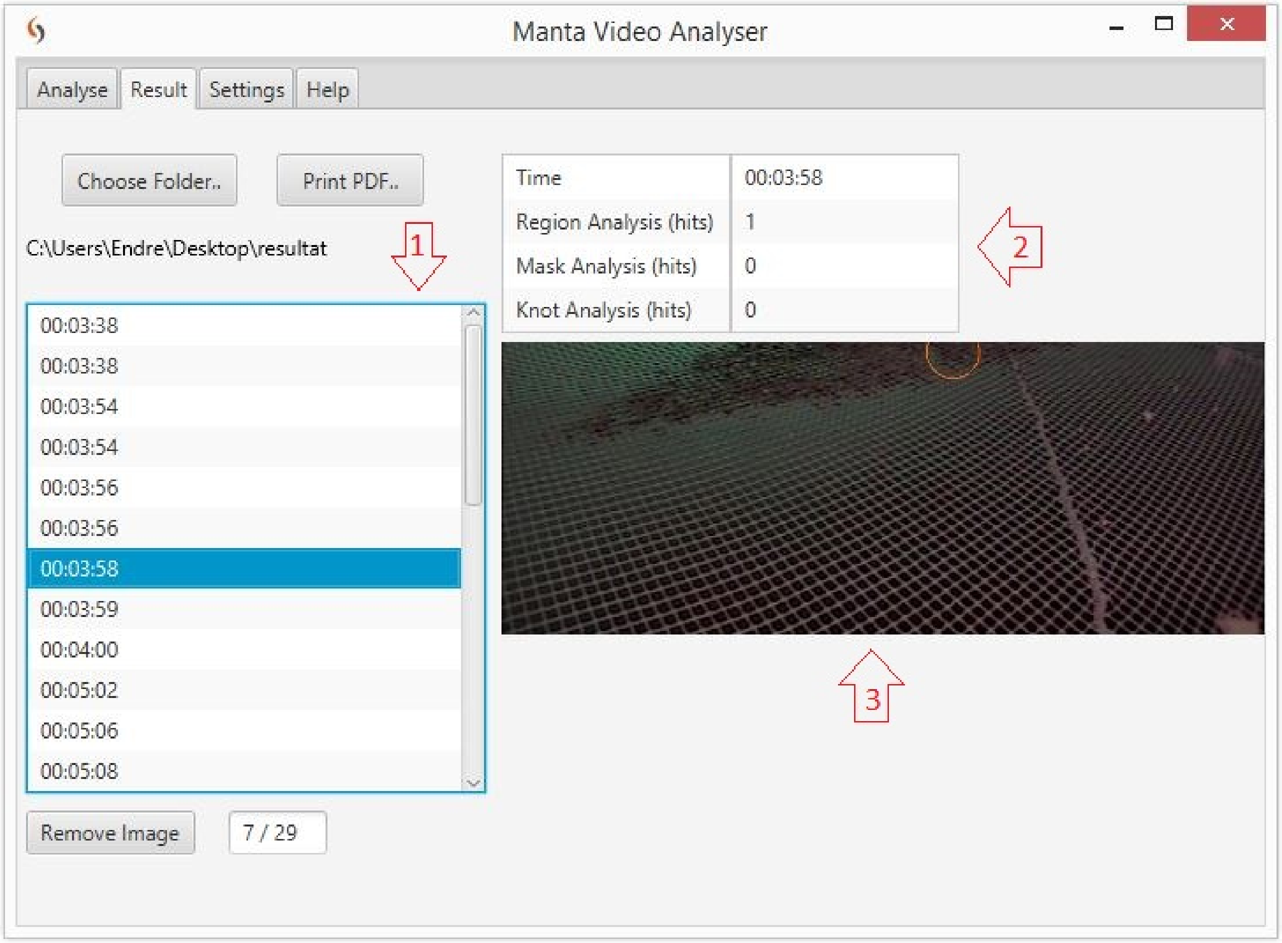
Figure 4.83: Picture showing the analyse tab in the GUI

The analysing tab is is the tab that is used to start and control an ongoing analysis. Before a video is chosen, the “Choose file” button is the only functionality that is available in the analysis tab. By clicking the “Choose file” button, the user gets the opportunity to choose a video-file to upload. As soon as a video-file is successfully uploaded, all the other functionality on the page is activated.

Before the user starts an analysis, there is an opportunity to adjust which part of the video to be analysed. As default the whole video is chosen, but this can easily be adjusted by using the time-slider (pointer nr.2, figure: 4.83) and the set-time buttons. As the time-slider is moved, the preview window (pointer nr.1, figure: 4.83) will update the preview with pictures from the corresponding time in the video. This makes it easier for the user to find a special time in the video. When the user has moved the time slider to the wanted start-time, he can set the start-time by clicking the “Set Start Time” button. The start-time indicator will then be updated with the new start-time. The end time is set in the same way, by moving the time-slider to wanted end time, and using the “Set End Time” button to update end-time indicator.

The “Start” button is used to start an analysis. When the “Start” button is pressed, a popup-window lets the user choose a destination folder as file location for the results. The results should be stored in an empty folder (for example a new folder in a folder containing previous results), so the user is informed if previous results are detected in the destination folder. The user can then choose to overwrite previous results, or choose a new destination folder. As long as the program is analysing, the progress is displayed in the progress-bar (pointer nr.3, figure: 4.83), and the preview window shows samples from the video being analyzed. The program also displays time used and estimated remaining time for the analysis to complete. This information shows up text-based below the progressbar when the analysis is started. The user also has the opportunity to pause or stop the analysis by using the “Pause” and “Stop” button. When the analysis is finished, a popup-window asks the user if he wants to upload the results directly to the results tab. If this is declined, the user has to choose and upload the result file to the result-tab manually. Previous results can also be selected manually.

4.4.2 Results Tab



C:\Users\Endre\Desktop\resultat

Time	00:03:58
Region Analysis (hits)	1
Mask Analysis (hits)	0
Knot Analysis (hits)	0

```
00:03:38
00:03:38
00:03:54
00:03:54
00:03:56
00:03:56
00:03:58
00:03:59
00:04:00
00:05:02
00:05:06
00:05:08
```

7 / 29

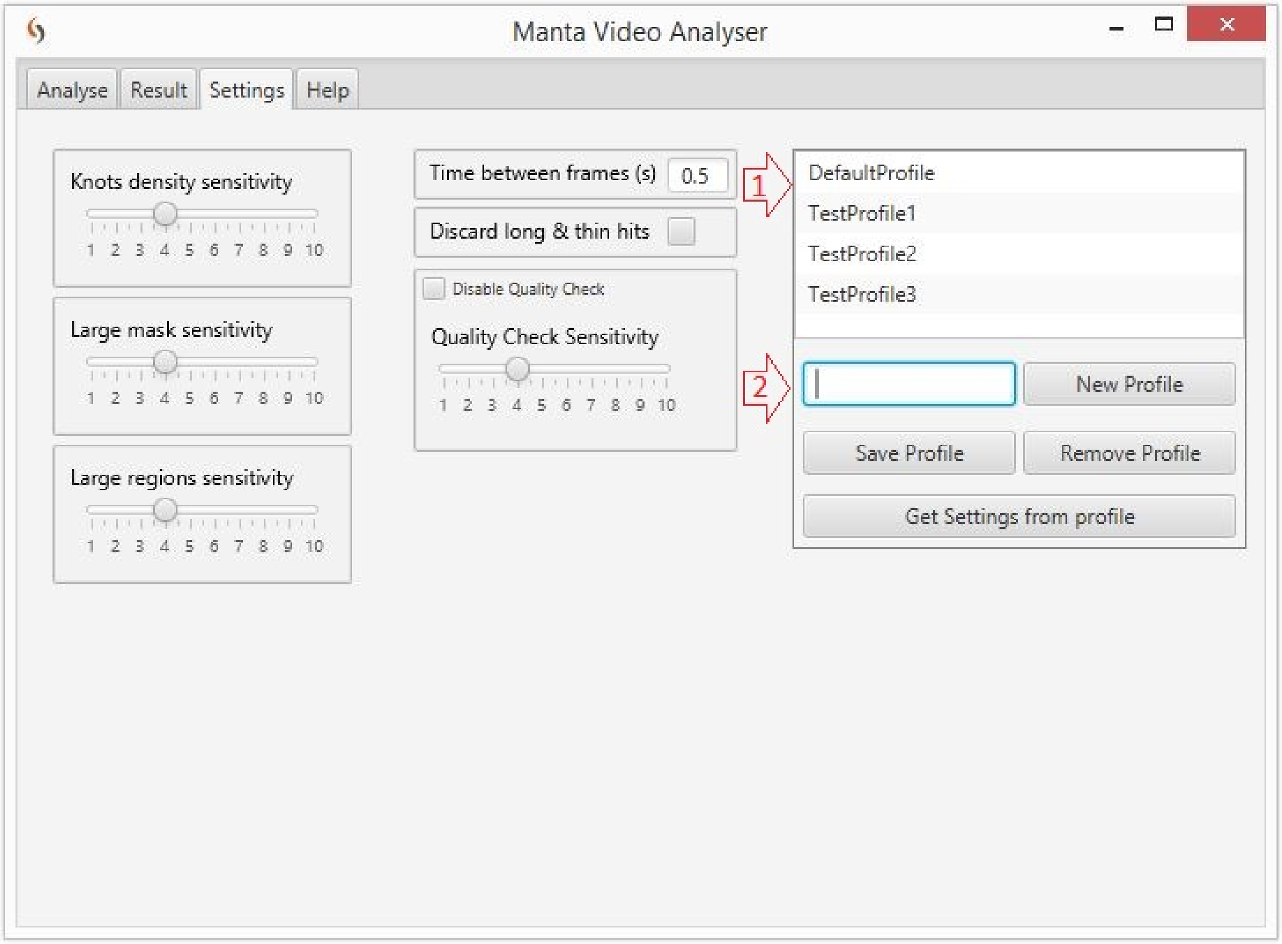
Figure 4.84: Picture showing the result tab in the GUI

The result tab offers the user of the program a chance to study the results of analyses. Before a result-folder is chosen, the “Choose folder” button is the only functionality that is available in the result tab (unless the user has loaded results directly after performing an analysis). By clicking the “Choose folder” button, the user gets the opportunity to choose a result folder to upload. As soon as a folder is successfully chosen, all the other functionality on the page is activated.

The frame-list (pointer nr.1 figure: 4.84) is filled as soon as the result folder is chosen and opened. In this list, the user can see all the frames from the current analysis that contains any hit. By clicking through the frame-list, the user can manually inspect each frame by watching the preview window (pointer nr.3, figure: 4.84) where the frame is presented. Each hit in the frame is marked by a circle which is placed exactly on the position where the hit is done. The circles uses color-code to tell which of the three analysis has done the hit. The information-list (pointer nr.2, figure: 4.84) shows the actual frames time in the video, and how many hits each of the analyses has detected in the frame.

Eventual incorrect hits can be removed by using the “Remove image” button. This button removes the hit, so that the frame is not included when the PDF file is created. The “Print PDF” button can be used to create a PDF document that contain all the information about the analysis. When the “Print PDF” button is pressed, a popup-window gives the user opportunity to add a comment that will be added in the PDF file. The user also has to type in a name and a location for the document to be saved in the popup-window.

4.4.3 Settings Tab



Knots density sensitivity: 4 (1 2 3 4 5 6 7 8 9 10)

Large mask sensitivity: 4 (1 2 3 4 5 6 7 8 9 10)

Large regions sensitivity: 4 (1 2 3 4 5 6 7 8 9 10)

Time between frames (s) 0.5

Discard long & thin hits □

□ Disable Quality Check

Quality Check Sensitivity: 4 (1 2 3 4 5 6 7 8 9 10)

DefaultProfile
TestProfile1
TestProfile2
TestProfile3

Figure 4.85: Picture showing the settings tab in the GUI

In the setting tab, it is possible to fine-tune the analysis. Each of the three analysis functions in the program has a sensitivity setting that can be individually adjusted. The quality check has its own sensitivity setting. It is also possible to activate/ deactivate the entire quality check. It is possible to activate / deactivate the function that discards long and thin hits (because such hits are usually false hits). In addition, it is possible to regulate how long period there should be between each of the frames analysed. (The Manta does not move too quickly when washing, so it is not necessary to analyse every frame in the video to cover the entire net)

In order to easily change these settings, the group has created functionality to create and store entire profiles with settings. In an easy way, it is possible to create and manage settings profiles with a desired name. When the program is adjusted with desired settings values, it is possible to save these settings to the desired profile. The next time the program is started to do an analysis, one can easily retrieve the desired settings from the saved profiles. The different settings-profiles is shown in the profile list (pointer nr.1, figure: 4.85)

To create a new settings profile, the user has to write wanted profile name in the textfield (pointer nr.2, figure: 4.85), and press the "New Profile" button. Then the new profile will appear in the settings-profile list. If the user is trying to add a new profile with the same name as an existing profile, the new profile will replace the old one. The user can click on a profile in the profile list to make the profile active. The "Get Settings from profile" button updates the current settings according to the settings saved in the active profile. The "Remove Profile" button removes the active profile and all its settings completely. By clicking the "Save Profile" button, the active settings-profile is updated according to the current settings.

The advantage of making such a solution with profiles is that it is very quick and easy to change settings. The group has experienced during the testing that it is a great advantage to be able to adapt the settings to different videos. For example, a video from a cage with a lot of grouse can require a more sensitive quality check, and less sensitive analysis methods than a clean one. Light conditions and camera focus are also parameters that will affect which settings are most ideal. By using settings profiles, one can easily switch from a set of settings that suit one type of videos, to another profile that fits a different type of videos. This makes the program very user-friendly.

4.4.4 Help Tab

The help tab contains information about how to use the program, and how the three analyses work. The help tab displays a html document with a user guide, included as an appendix.

4.5 PDF Report

The group wanted to be able to present test results in a good and user-friendly manner. Presenting results in the results tab of the analysis program itself is excellent for the user of the program. But to present the results to others, a more portable solution is required. For example, one will probably present any damage findings in the cages to the owner or person in charge at the fish-farm.

To do this, the group has built in functionality in the program that enables the production of a report, in form of a PDF document based on analysis results. The user has the opportunity to open any test results in the results tab and check the results manually. If the user finds some findings wrong, it is possible to remove this before the PDF document is produced. The user can also add a comment to the results. To make the PDF document, the user can press the "Write PDF" button, and the PDF document will be made.

The PDF file has a front-page which contains video name, used settings, comment and general statistics of the entire analysis, such as image quality, and number of images with hits. After the first page, a list of links to all the images is set up based on time in the video. The reader is able to click on each of the links to jump to the info page of the actual image. In the info page, the reader can see the current picture, where in the picture there are hits, and which tests have made the hits. One can also read the time of the image to easily check the frame against the video.

4.6 File structure

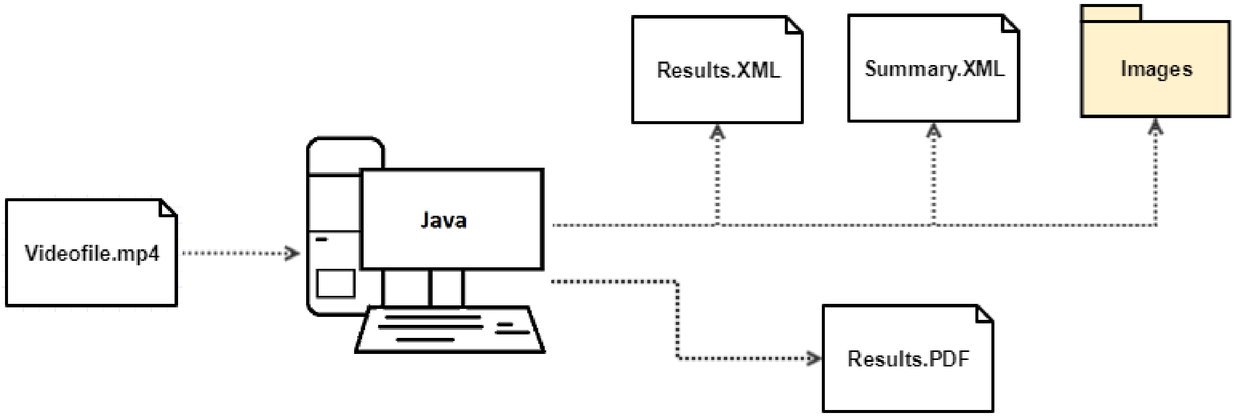


Figure 4.86: Picture showing the filestructure.

The file structure overview 4.86 shows which result files the program produces. In addition to the start and result files on figure 4.86, an internal XML file (settings.XML) is used to store settings profiles. Before starting an analysis, a video file must be selected for uploading. When the analysis runs, the results are stored as follows:

- **Results.XML** stores the analysis results for each frame with hit. It includes data such as frame number and time in the video, result from the quality control, and result from the three analyses.
- **Summary.XML** stores statistics of the analysis as a whole, in addition to the settings that were used during the analysis.
- **Images** Inside this folder, all the frames with hits are saved. Each hit is marked on the image by a circle.
- **Results.PDF** This document can be created by the user after each analysis if desired. It contains all the important information about the analysis, including each image with a hit.

Chapter 5

Discussion

5.1 Discussion of the technical results

5.1.1 Image Processing

Quality Check

With the current camera position on the Manta, it is necessary to use a quality control to separate frames and discard frames or parts of frames where the net masks cannot be recognized. If the quality control is not used, many of the frames would result in false positive results. Examples of this is when the camera is too distant or at a steep angle so that the masks can't be recognized, or when the camera looks too far up from the net.

The advantage of using the quality control is that you avoid many incorrect hits. The disadvantage of using the quality control as it is today, is that it may remove frames that should be analysed. This means that the chances of not being able to detect damages that should be detected is increased. In order to be sure that there are no missed holes that should have been detected, it is necessary to manually inspect all the pictures that the quality check removes. Today, this will be a very time-consuming job, because much of the video files are not suitable for image analysis. Too many images do not show the net clearly, as figure 4.25 shows.

Large mask analysis

The large mask analysis is a very accurate analysis of frames with the net clearly visible. This analysis is able to detect holes all the way down to a single line being broken, as shown on figure 4.80. However, due to the fact that it is based on the skeleton image, it depends on an accurate skeleton representation of the frame. If the lines in the net are vague or too close to each-other, or not visible at all, it is possible to get a false skeleton representation of that part. In some cases, this leads to false hits being created by the skeletonization algorithm. Because of this, the large mask analysis does not work well on parts of a frame where the net is not clearly distinguishable from the background. When the skeleton image is good, the large mask analysis has a very good accuracy.

Large object analysis

Large object analysis works better than the large mask analysis on frames where the net is not always clearly distinguishable from the background. This analysis works well, without finding many false hits on unclear images. Large object analysis is not as sensitive or accurate as large mask analysis on high quality images.

When the Manta turns, the camera tends to focus far ahead. When the camera looks far ahead, it often sees areas with shadows because the light on the manta does not reach too far ahead of the manta. When this happens, large object analysis tends to detect shadows as a hole. This is a weakness with this analysis.

Knot density analysis

The Knot detection analysis is quite good on both good and less clear images. The disadvantage of this analysis is that it tends to overlook smaller holes in nets that have a lot of fouling. The analysis can sometimes detect knots in the fouling, this can contribute to failure.This happens especially if there is a lot of fouling in the area around small holes. Another source of error for this analysis is narrow holes. An example of this could be if masks are cut without the density of knots being seriously affected. In addition, the knot density analysis cannot detect small holes where the density of knots in the net is not affected by the hole. Therefore, this analysis will not be as accurate as the large object and large mask analyses to detect small damages. Figure 4.80 is a good example of this. In this example, the number of knots is not affected by the damage in the net, and thus the knots density analysis is not able to detect the damage.

5.1.2 Test results

The test analysis 4.2.2 shows that from the total 689 frames, 159 are completely removed by the quality check due to bad quality. This corresponds to 22.8 percent of the total number of frames. In addition, the quality control removes 46 percent from the remaining 532 frames that are not removed in their entirety. This means that the quality control totally removes 57.5 percent of the total video, and 42.5 percent of the selected video clip is analysed.

As figure 4.78 and the left corner of figure 4.77 shows, the quality check still does not remove enough. Other tests the group has made of the quality control have also shown that the quality control occasionally takes out pictures or parts of pictures that should not be removed. Most of the 41 hits made during the test analysis were very good. Most of them could have been real holes, as shown in figure 4.77 and 4.80.

5.1.3 Unused methods

LMA/LOA: Compare contour area to image area

During the development of the program, the group tested comparing the area of each contour to the total area of the image. An advantage of this is that the analysis is less susceptible to the average area being skewed, but it was quickly determined that this did not work as intended. Due to the manta moving up and down, the size of the masks vary widely, so it is more accurate to compare the area to the average area of the masks in an image.

Hugh Lines Transform

In the beginning of the project, one of the methods we tested was the Hugh Lines Transform 2.1.8. This method was natural to test as it can detect lines, and wires in a net can look like lines.

However, after testing the Hugh lines methods in OpenCV, it became clear that this method is not suitable for detecting lines in the net because the lines are not always straight. The net stretches and bends, which often makes the pattern slightly uneven. The Hugh lines method proved far too inaccurate for the project, so it was scrapped early on.

5.1.4 Software

GUI

The main goal of the GUI design was to make it as clear, user-friendly and functional as possible. To achieve this, it was decided to distribute the functionality over three tabs in addition to an information tab (analysis, results, settings and help tabs). The group thinks this is a natural way to build the GUI. By using good names for buttons, the GUI becomes as self-explanatory as possible. The GUI is adapted to the desired window size, which can be adjusted manually. The group very satisfied with the way the GUI ended up looking and functioning.

Multithreading

As shown in the speed test, the group saw a big speed improvement when running multiple Analyser Threads concurrently. By implementing proper concurrency principles this was done in a safe manner. For example, the use of synchronization makes sure that only one thread can fetch an image or update the results at the same time.

Due to the speed difference, as shown in 4.2.1, the group is now confident that the analysis can be run in real-time during the washing. On our test machine we were able to get close to analysing 5 images per second, but the machines the software is to be deployed on is even more powerful than our test machine.

5.2 Discussion of the project itself

5.2.1 Development method

Scrum

The use of Scrum as development method 3.4.1 has worked well. The group has experienced that this has been a good way to organize the work. By dividing the work into smaller concrete tasks, it has been easy to distribute the tasks and keep track of the progress. The use of specific work tasks that should be done by the end of the sprint, has secured good work effort. This has also contributed to the good progress in the project. The group has set high goals, and has not always been able to complete all tasks by the end of each sprint. The advantage of this is that there has always been something to strive for.

Jira and Confluence

The group has used both Jira and Confluence as a tool for project management. Confluence has been used to write status reports, summaries from the meetings, charts and dashboards with presentation of the group and project. Jira has been used to create sprints and handle tasks. In the beginning, it took some time to familiarize with these tools. The group members had not used these tools before this project. But as soon as the group got used to the tools, this worked well. It takes some time to post tasks for each sprint, but this is a great way to plan the sprint. Jira provides a good overview of the status of the various tasks, and thus the progress of the project. The benefits of these tools will be even better in bigger groups.

5.2.2 Progress in relation to prepared progress plan

The pilot project was carried out and completed according to the schedule. The preliminary study that involved studying and testing of image analysis functions and previous projects were not entirely completed in accordance with the progress plan. A project like this consists of a lot of experimentation and testing. This has lead to some unforeseen challenges, such as the quality control. This has meant that the part of the pre-project which involves studying image processing methods has gone parallel to the development phase. The development part itself was performed according to the plan, in time for the group to write the main report.

The group has prioritized work tasks and time with regard to an overall goal of being able to complete a functioning application. This priority has been successful and the result is a functioning and complete application. The result can be called complete in the sense that it contains the functionality that it was planned to provide in the preliminary project.

5.3 Further development

The group means that the deadline has been the greatest limit in this project. With some more time, there is a great potential for further development.

5.3.1 Image Processing

Improve Quality Check

The group sees potential for improving the current quality check. As it is, it splits the image into defined regions and checks each of these. Ideally, it should be possible to make a mask for each image that only blocks out the bad parts. We conducted some research into this, but were ultimately not able to come up with a solution within our time-frame. There are some new challenges with such an approach, for example areas with damage could be detected as unfit for further analysis, thus eliminating the purpose of the analysis. A balance has to be struck, which the group thinks is a great topic that can be looked at more in the future.

Fish detection

As the program is today, fish might create shadows that are detected as holes. This is a topic that can be studied and resolved in further development.

GPU acceleration

Another potential for further development is utulising the GPU to do image processing. The machines the software is to be deployed on have state of the art GPUs, so the potential to speed up the processing is huge.

5.3.2 Real-time analysis

The program has been set up for easy transition to running real-time analysis. The group envisages a new class with the task of grabbing frames from a live feed from the camera, and then sending them to the VideoUploader. With minimum change, the VideoUploader can be changed to accomodate this and distribute the frames to the Analyser Threads. Thus, with the object oriented programming principles in mind, we have prepared for an easy real-time implementation.

5.3.3 Video production on the manta

Camera location and angle

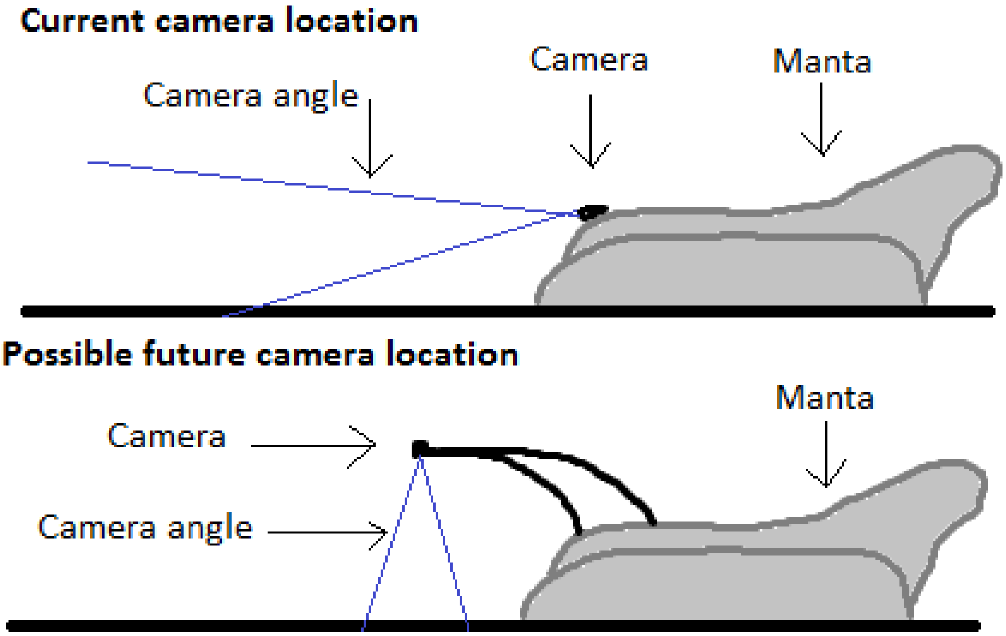


Figure 5.1: Existing and possible future camera position and angle.

The group believes that there is a great potential for improvement in the production of the video. The work of picking out images or parts of images that has too bad quality to analyse has been significant in this project. The biggest challenge with today's solution as the group sees it, is the camera angle towards the net. Current camera location and angle makes it sometimes difficult to detect masks in the net. This becomes extra challenging as the manta doesn't always follow the net closely, but tilts in different directions. This means that the net often disappears completely from the focus area of the camera. The group has devised a possible future camera location and camera angle. This is outlined in the figure 5.1.

Lighting conditions

Another challenge with today's solution is the lighting conditions. The manta has light that illuminates the grid. This lights up the net, while the background (sea) is dark. This makes it possible to separate the net from the background. The challenge is that sunlight is sometimes reflected from the water surface, and enters the camera's focus area. When this happens, much of the video pictures often turns white. This means in practice that we do not have pictures from that area. It also happens that sunlight makes the grid lighter than the background in parts of the camera's focus area. At the same time as the net is darker than the background in the other part of the camera‘s focus area. This makes the image processing challenging.

We believe that our proposal for future camera location will help to improve lighting conditions. Our proposal for future camera placement also allows for extra strong LED lighting. The group envisages that additional lighting can be mounted next to the camera and at the same angle as the camera. The group believes that this will be able to greatly improve the lighting of the network, which will probably improve the video images and thus the accuracy of the video analysis.

Chapter 6

Conclusion

The group has created a functional software to analyse video files from the Manta net-cleaner to detect defects in the fish-net. To perform the actual damage detection, the group has developed three different analyses, each with their own strengths and weaknesses. The group has created functionality to handle the quality control of an image, used to determine if the image can be analysed. A user-friendly GUI makes the software easy to use. By using multithreading, the group has achieved satisfactory speed in order to run the analyses in real time. The analysis result can be presented both in the GUI, and as a PDF report.

The program works well and meets the task requirements. The software has been tested on relevant videos produced by the Manta. The group experiences that the software works, and is able to detect many defects in the fish-net that could potentially be holes. However, some wrong hits are made, and the software has a good potential for further development.

During the work on this assignment, the group has acquired a lot of valuable new knowledge. Especially in programming, image analysis, and building a functional GUI. The group has been creative and conducted many experiments to find solutions to challenges along the way. The experience of working independently as a group over time, while having a dialogue with the supervisor and the company has also been valuable. The group has had good cooperation from start to finish, both internally, with supervisors and Stranda Prolog AS. Meeting notes and retrospects support this.

Bibliography

[1] Camilla Aadland. Slik vil optimar digitalisere fiskeri- og oppdrettsnæringen. URL https://fiskeribladet.no/tekfisk/nyheter/?artikkel=65702.

[2] Altassian. What is confluence?, . URL https://confluence.atlassian.com/confeval/confluence-evaluator-resources/confluence-what-is-confluence.

[3] Altassian. A brief overview of jira, . URL https://www.atlassian.com/software/jira/guides/getting-started/overview.

[4] Altassian. Bitbucket, . URL https://bitbucket.org.

[5] Image Analyst. How does bwmorph define a branch point? URL https://se.mathworks.com/matlabcentral/answers/292215-how-does-bwmorph-define-a-branch-point#answer_226798.

[6] Stranda Prolog AS. Stranda prolog as: Manta net cleaner, . URL https://www.stranda.net/no/produkter/manta-notvasker.

[7] Stranda Prolog AS. Stranda prolog as: About us, . URL https://www.stranda.net/no/om-oss.

[8] The book Learning OpenCV by Gary Bradski and Adrian Kaehler.

[9] Kate Chidzik. About overleaf. URL https://www.overleaf.com/about#who-we-are.

[10] Michael Kolling David J. Barnes. *Objects first with java, Sixth edition, ISBN: 978-0-13-447736-7.*

[11] The Tech Terms Dictionary. Pdf definition. URL https://techterms.com/definition/pdf.

[12] docs.oracle.com. About the java technology. URL https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html.

[13] Claire Drumound. What is scrum? URL https://www.atlassian.com/agile/scrum.

[14] The Apache Software Foundation. About apache pdfbox. URL https://pdfbox.apache.org.

[15] Stephanie Glen. Bimodal distribution: What is it? URL https://www.statisticshowto.datasciencecentral.com/what-is-a-bimodal-distribution/.

[16] Magnus Hellem. Pattern analysis of nets used in fish farming. URL https://forum.image.sc/t/pattern-analysis-of-nets-used-in-fish-farming/1032.

[17] https://www.tutorialspoint.com. Opencv - adaptive threshold. URL https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm.

[18] Java.com. What is java technology and why do i need it? URL https://www.java.com/en/download/faq/whatis_java.xml.

[19] Martijn Laan Jordan Russell.

[20] Fatih Kodak.

[21] W3C XML Activity Lead Liam Quin. Extensible markup language (xml). URL https://www.w3.org/XML/.

[22] Mathworks. Image processing toolbox, . URL https://se.mathworks.com/help/images/.

[23] Mathworks. Millions of engineers and scientists trust matlab, . URL https://se.mathworks.com/products/matlab.html.

[24] Mathworks. What is matlab, . URL https://se.mathworks.com/discovery/what-is-matlab.html.

[25] Miljødirektoratet. Fiskeoppdrett, en nering i vekst. URL https://www.miljostatus.no/tema/hav-og-kyst/fiskeoppdrett/Rapport.

[26] Oracle Monica Pawlan Technical Writer. What is javafx? URL https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm.

[27] University of Edinburgh. Bilateral filtering for gray and color images. URL http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html.

[28] OpenCV. Contours: Getting started, . URL https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html.

[29] OpenCV. Histograms - 1: Find, plot, analyze, . URL https://docs.opencv.org/3.1.0/d1/db7/tutorial_py_histogram_begins.html.

[30] OpenCV. Color conversions, . URL https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html.

[31] Oracle. About netbeans ide, . URL https://www.oracle.com/technetwork/developer-tools/netbeans/overview/index.html.

[32] Oracle. Javafx scene builder, . URL https://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html.

[33] Oracle. Concurrency in javafx, . URL https://docs.oracle.com/javase/8/javafx/interoperability-tutorial/concurrency.htm.

[34] Oracle. Handling events, . URL https://docs.oracle.com/javase/8/javafx/events-tutorial/events.htm.

[35] N Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076.

[36] A. Walker R. Fisher, S. Perkins and E. Wolfart. Distance transform. URL https://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm.

[37] Jason Long Scott Chacon. Git-distributed-even-if-your-workflow-isnt. URL https://git-scm.com.

[38] Joachim Seehusen. Oppdrettsnæringen ber om automatiserte løsninger. URL https://www.tu.no/artikler/oppdrettsnaeringen-ber-om-automatiserte-losninger/224625.

[39] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. URL https://www.sciencedirect.com/science/article/pii/0734189X85900167.

[40] Richard Szeliski and to LearningOpenCV. Computer vision: Algorithms and applications. URL https://docs.opencv.org/3.3.1/dc/dd3/tutorial_gausian_median_blur_bilateral_filter.html.

[41] OpenCV team. About opencv. URL https://opencv.org/about/.

[42] Inc The MathWorks. Analyzing images using image thresholding techniques, . URL https://www.mathworks.com/discovery/image-thresholding.html.

[43] Inc The MathWorks. bwmorph, . URL https://se.mathworks.com/help/images/ref/bwmorph.html.

[44] Scott E Umbaugh. *Digital Image Processing And Analysis, Third Edition, ISBN 978-1-4987-6602-9*. CRC Press, Boca, Raton, London, New York, 2018.

[45] Andy Wellings. *Concurrent and Real-Time Programming in Java, Third Edition, ISBN 978-0470844373*. John Wiley Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2004.

[46] Mike Williams.

[47] www.tutorialspoint.com. About java jdom parser. URL https://www.tutorialspoint.com/java_xml/java_jdom_parser.htm.

[48] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM,* 27(3):236–239, 1984. doi: http://www-prima.inrialpes.fr/perso/Tran/Draft/gateway.cfm.pdf.

Appendices

A PDF Report (example report from an analysis)

B Userguide for the application

C Preproject report

D Meeting report 11.01.2019

E Meeting report Stranda Prolog AS 16.01.19

F Meeting report 25.01.19

G Meeting report Stranda Prolog AS 11.02.19

H Meeting report 22.02.19

I Meeting report 11.03.19

J Meeting report Stranda Prolog AS 15.03.19

K Meeting report 25.03.19

L Meeting report Stranda Prolog AS 10.04.19

M Meeting report 10.04.19

N Meeting report 26.04.19

O Meeting report 09.05.19

P Retrospective 24.01.2019

Q Retrospective 07.02.2019

R Retrospective 21.02.2019

S Retrospective 10.03.2019

T Retrospective 25.03.2019

U Retrospective 09.04.2019

V Retrospective 25.04.2019

W Retrospective 08.05.2019

Filename of analyzed video:

S:\Manta-net (S)\Not 2 - Del 2.mp4

Comment:

This is a comment!

Result Overview:

Number of rejected frames:	157
Number of analyzed frames:	689
Percentage analyzed of analyzed frames:	54.0
Number of frames with hit(s):	41

Settings:

Start Time:	00:15:02
End Time:	00:20:36
Knot Density Multiplier:	4.0
Large Masks Multiplier:	4.0
Large Regions Multiplier:	4.0
Quality Check Multiplier:	4.0
Time Between Frames:	0.5
Disable Quality Check:	false
Use Long Thin Compensation:	true

Time: 00:16:11

Time: 00:16:25

Time: 00:16:26

Time: 00:16:26

Time: 00:16:26

Time: 00:16:27

Time: 00:16:28

Time: 00:16:29

Time: 00:16:29

Time: 00:16:31

Time: 00:16:34

Time: 00:16:34

Time: 00:16:35

Time: 00:16:37

Time: 00:16:38

Time: 00:16:41

Time: 00:16:42

Time: 00:16:52

Time: 00:16:53

Time: 00:17:30

Time: 00:17:30

Time: 00:17:31

Time: 00:17:32

Time: 00:17:39

Time: 00:17:44

Time: 00:17:45

Time: 00:17:52

Time: 00:18:07

Time: 00:18:07

Time: 00:18:08

Time: 00:18:08

Time: 00:18:09

Time: 00:18:12

Time: 00:19:13

Time: 00:19:13

Time: 00:19:14

Time: 00:20:23

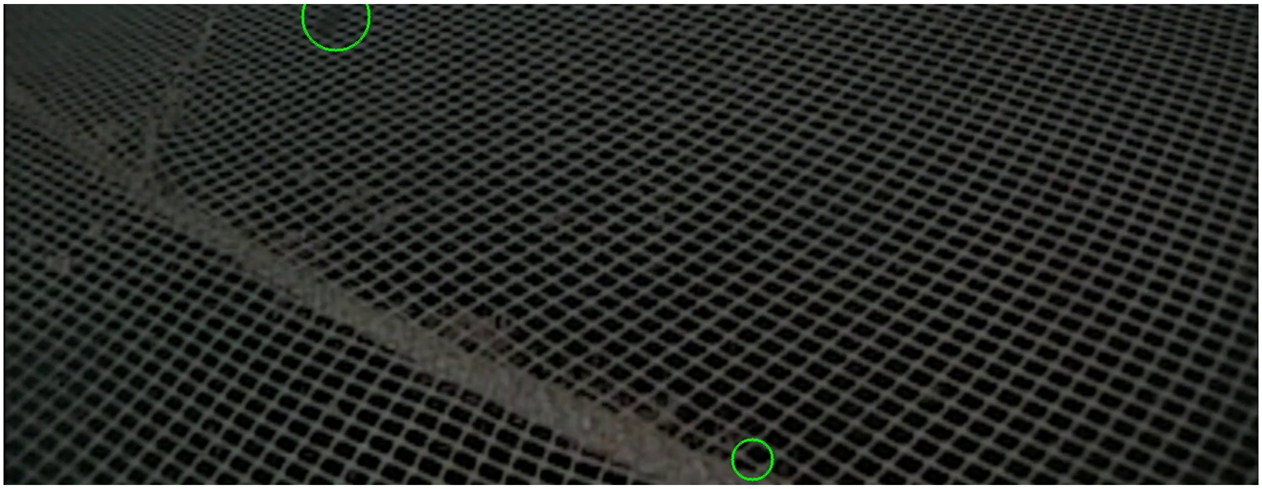
Time: 00:20:24

Time: 00:20:25

Time: 00:20:26

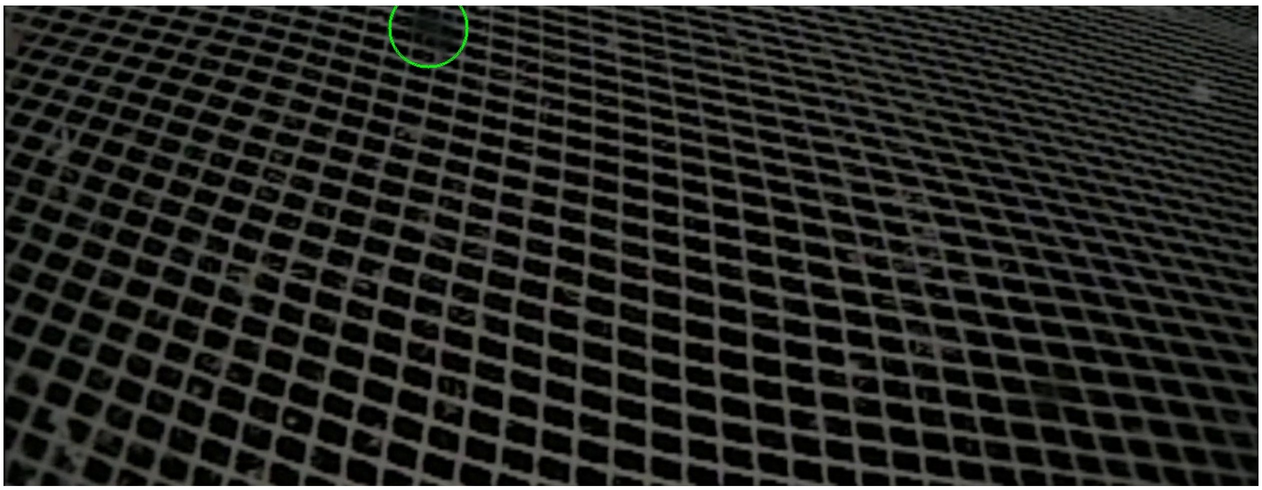
Time: 00:20:29

Back to top



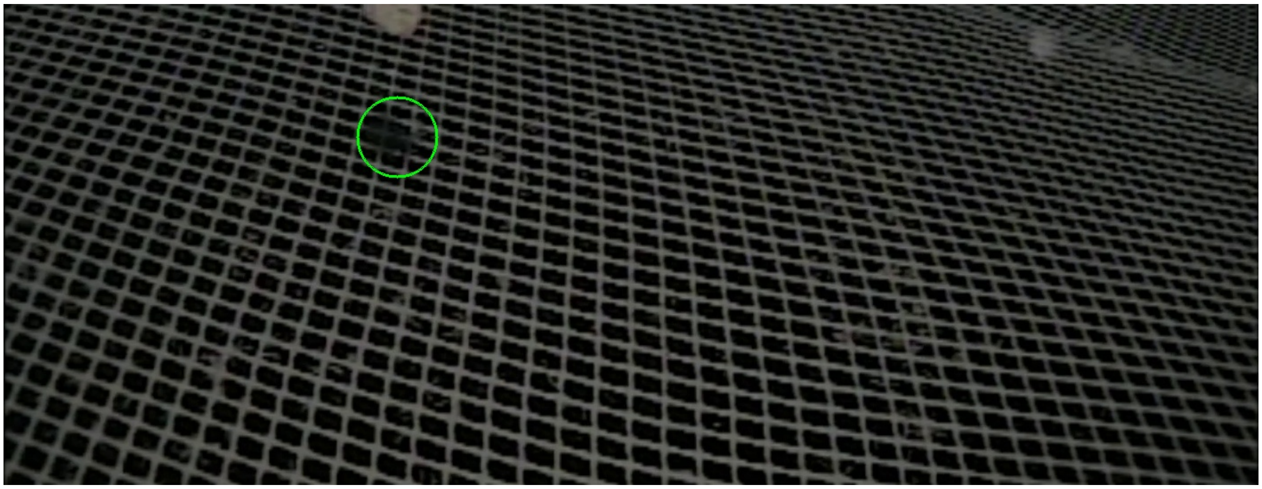
Frame Id	28174
Time	00:16:11
Percentage Analzed	80
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



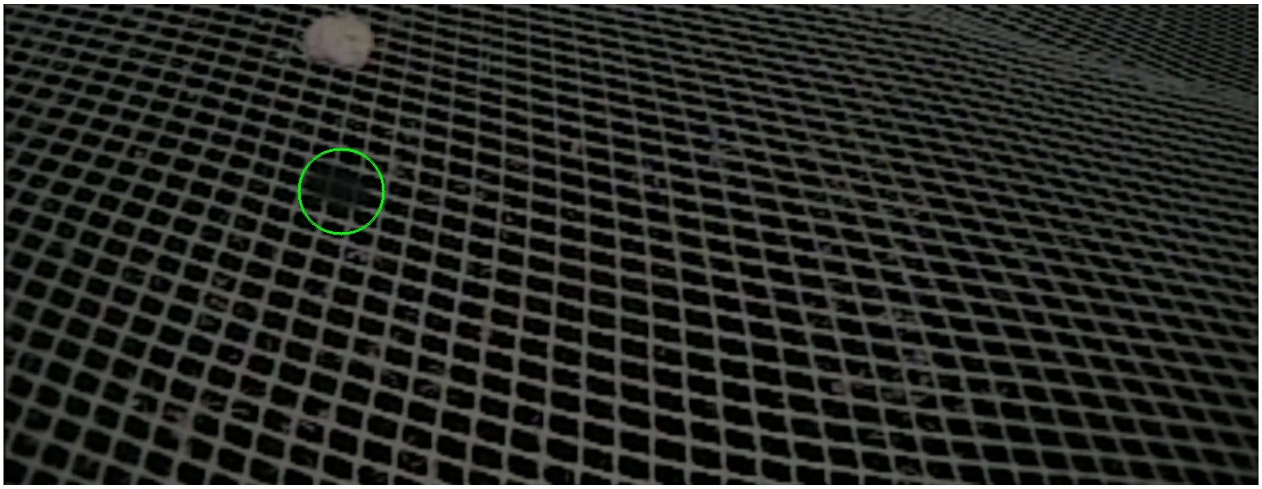
Frame Id	28580
Time	00:16:25
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



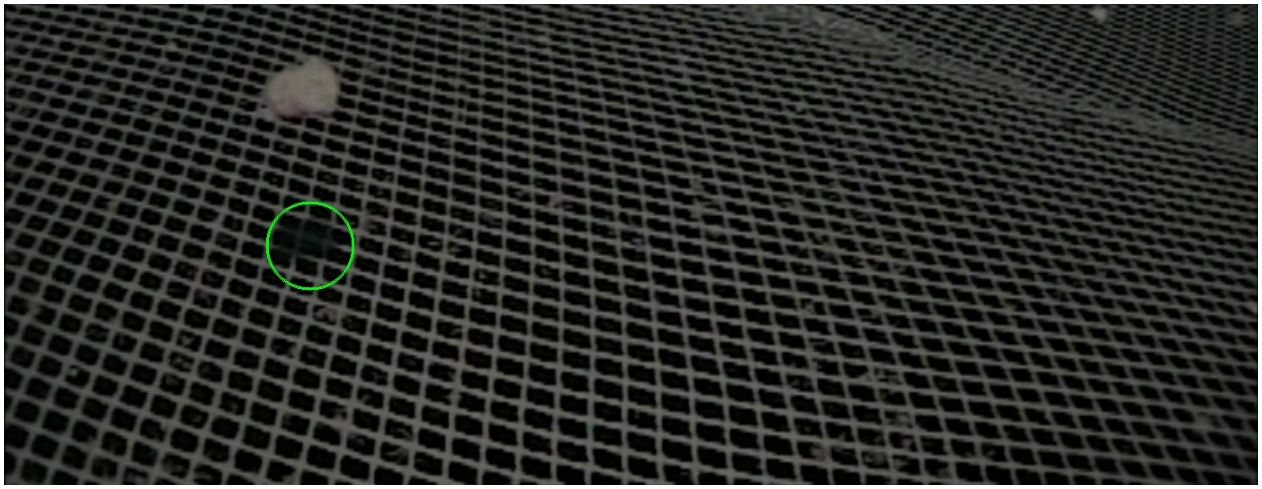
Frame Id	28594
Time	00:16:26
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



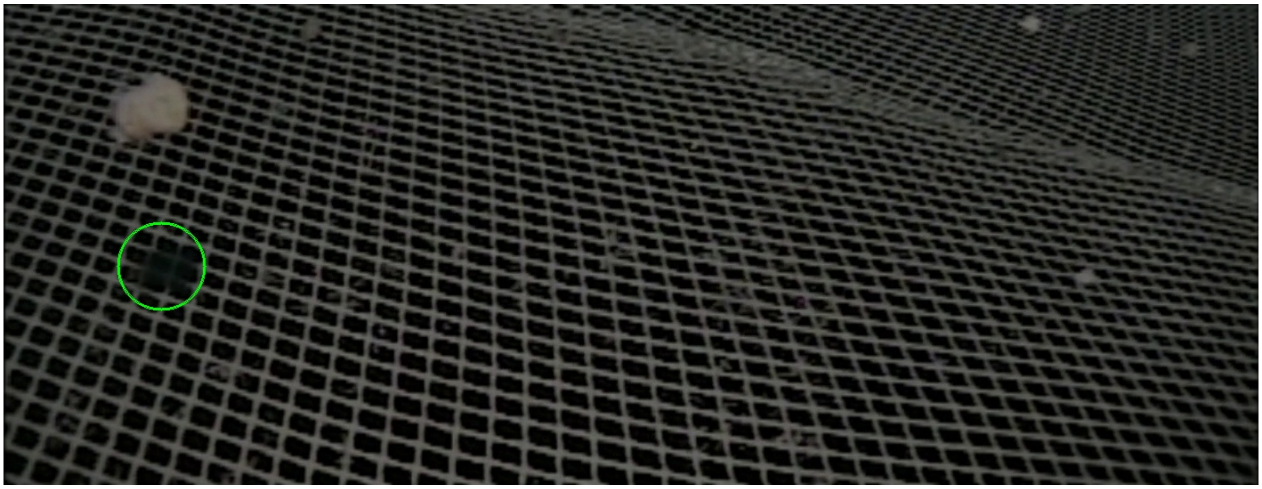
Frame Id	28608
Time	00:16:26
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



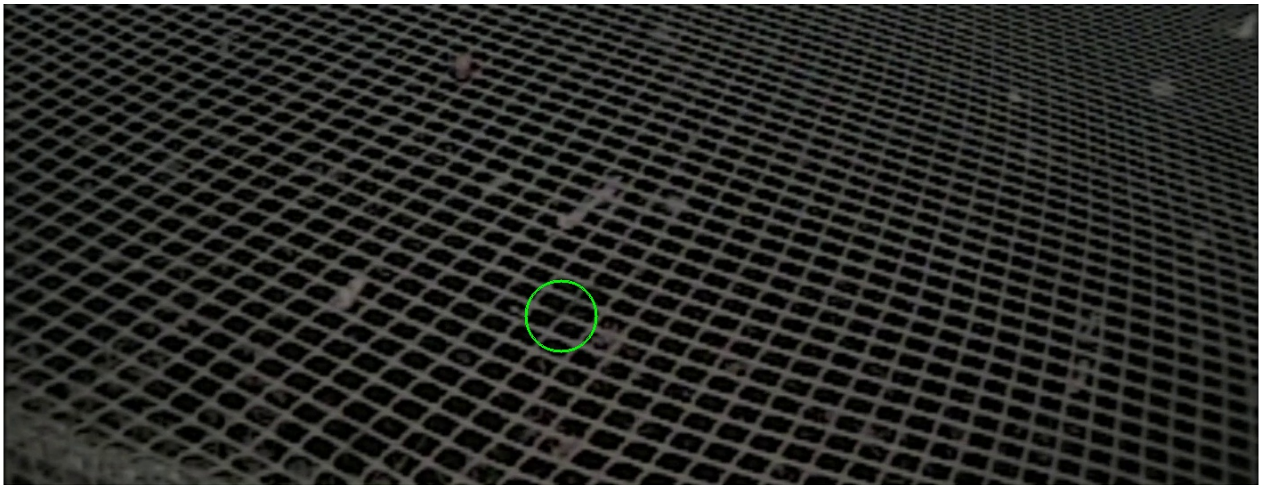
Frame Id	28622
Time	00:16:26
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



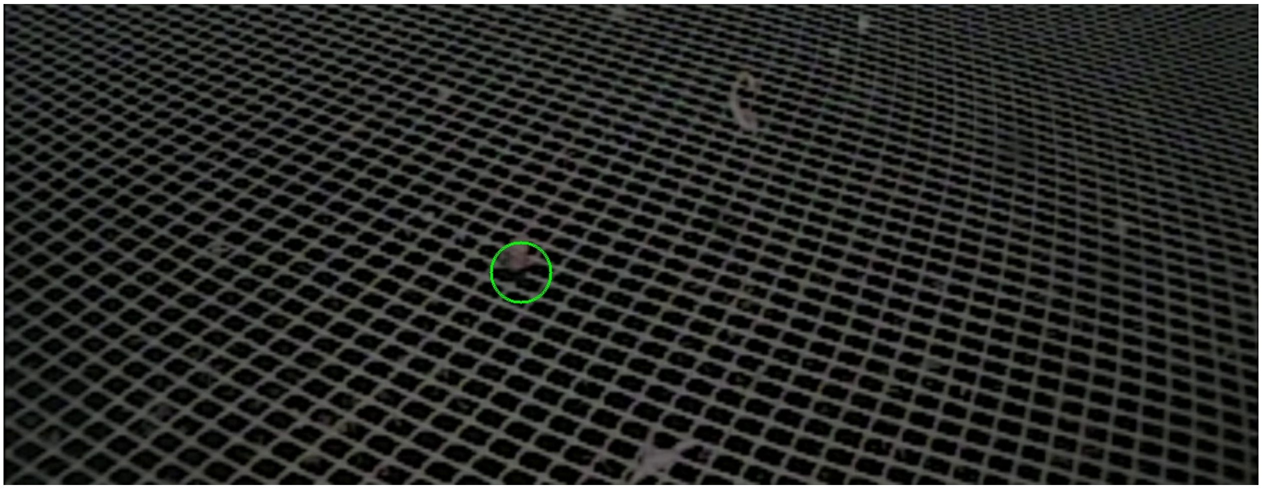
Frame Id	28636
Time	00:16:27
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



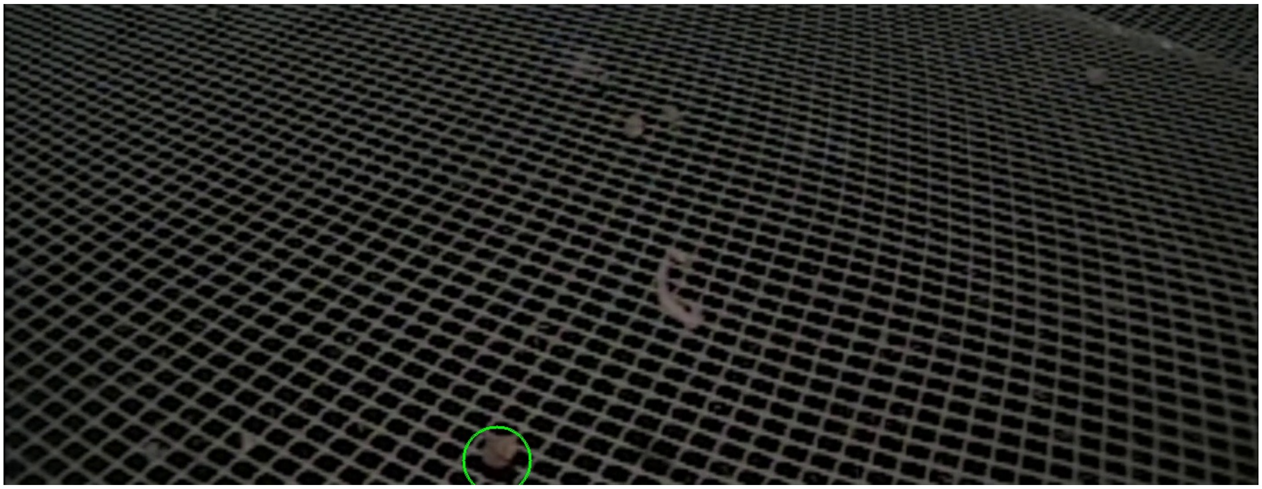
Frame Id	28678
Time	00:16:28
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



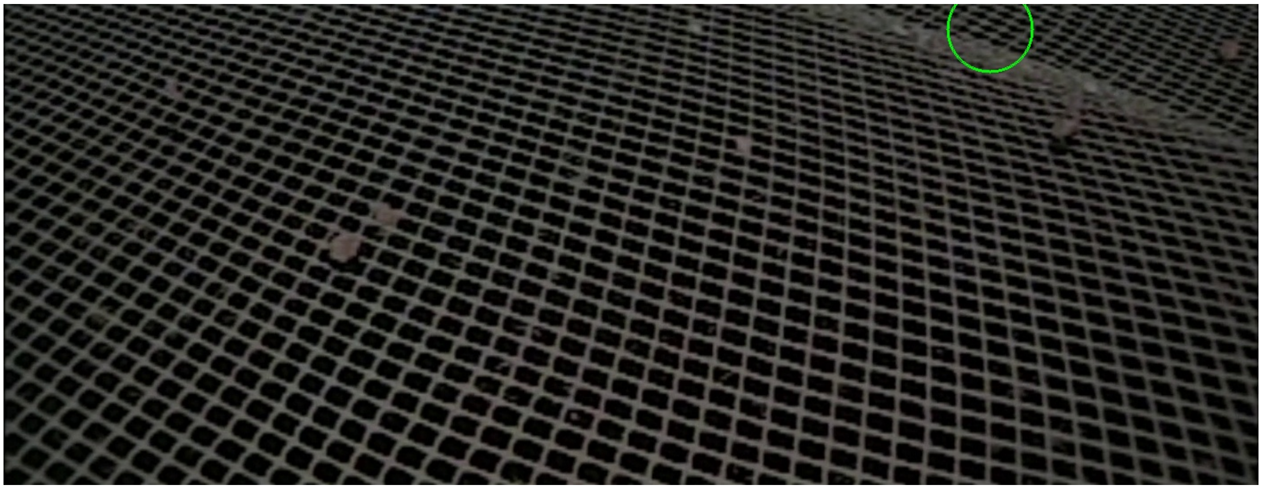
Frame Id	28692
Time	00:16:29
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



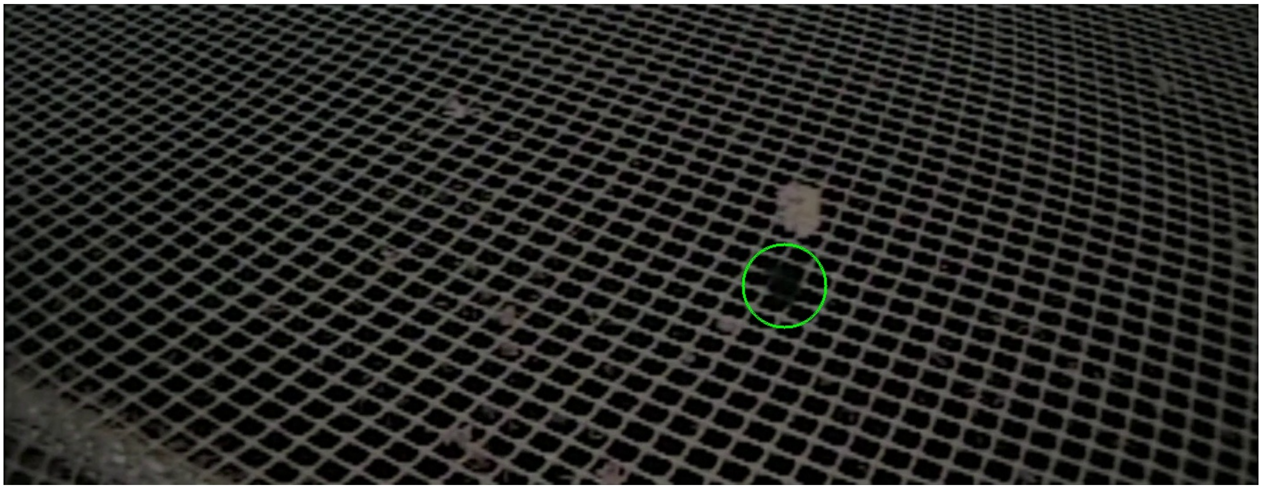
Frame Id	28706
Time	00:16:29
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



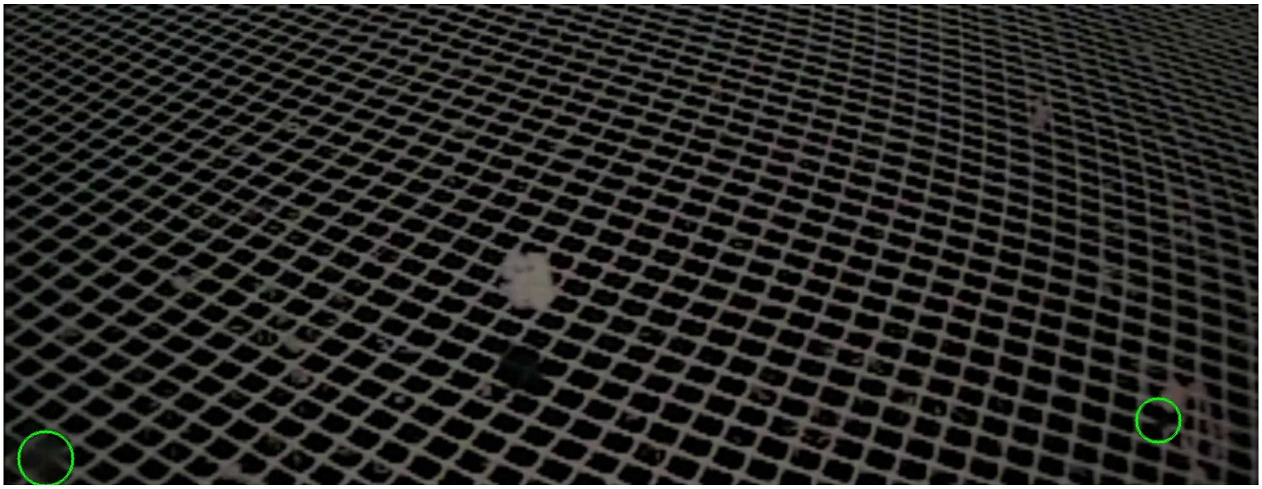
Frame Id	28762
Time	00:16:31
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



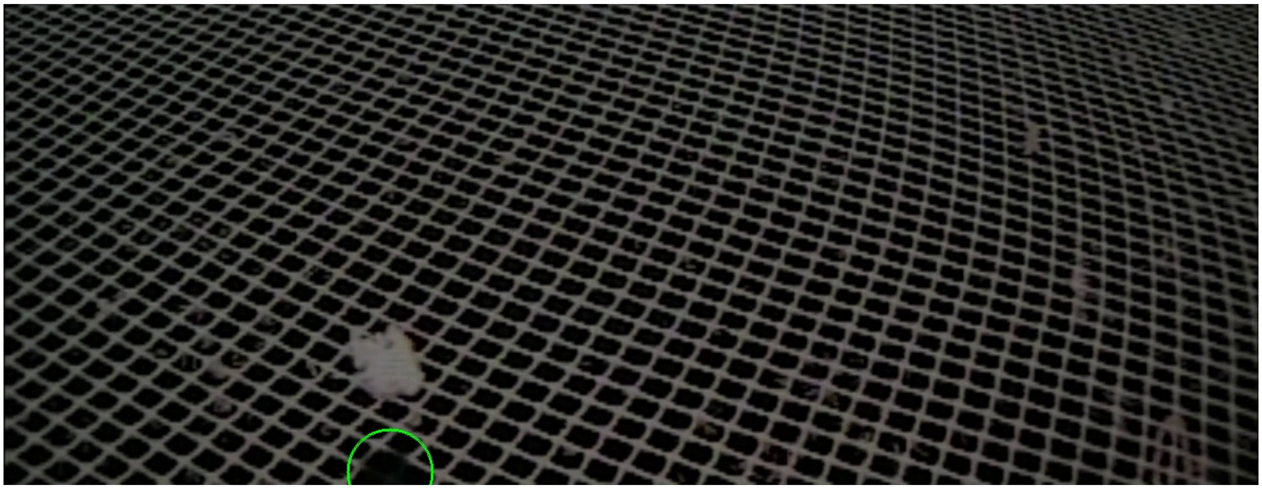
Frame Id	28832
Time	00:16:34
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



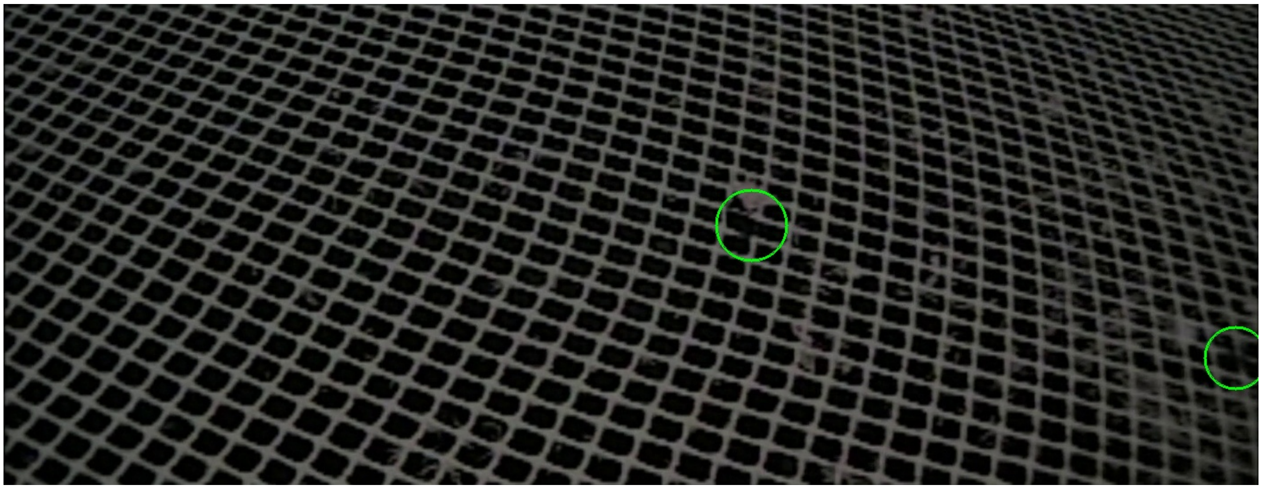
Frame Id	28846
Time	00:16:34
Percentage Analzed	60
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



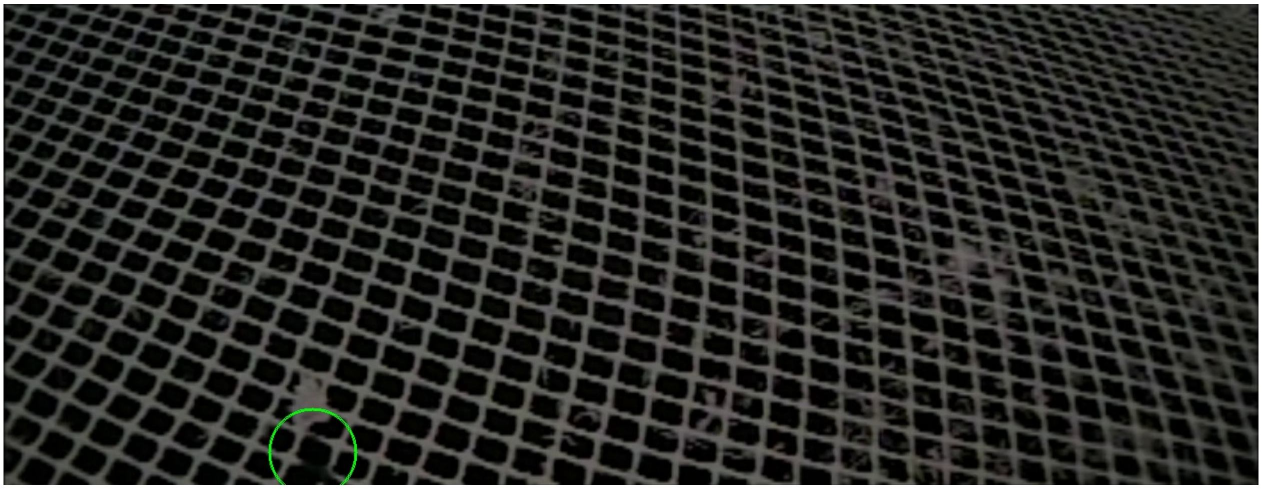
Frame Id	28860
Time	00:16:35
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



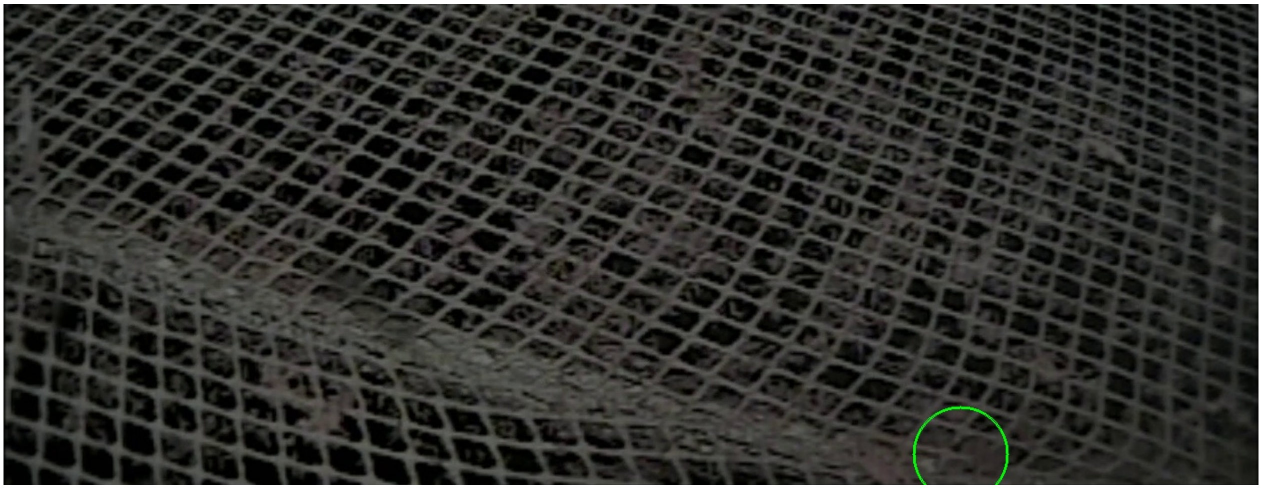
Frame Id	28916
Time	00:16:37
Percentage Analzed	100
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



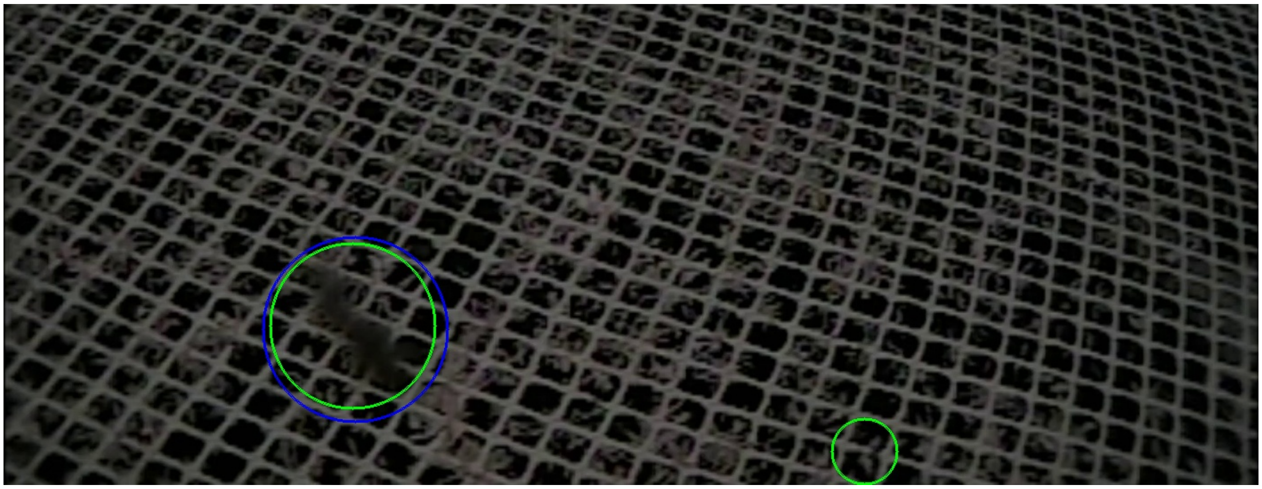
Frame Id	28944
Time	00:16:38
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



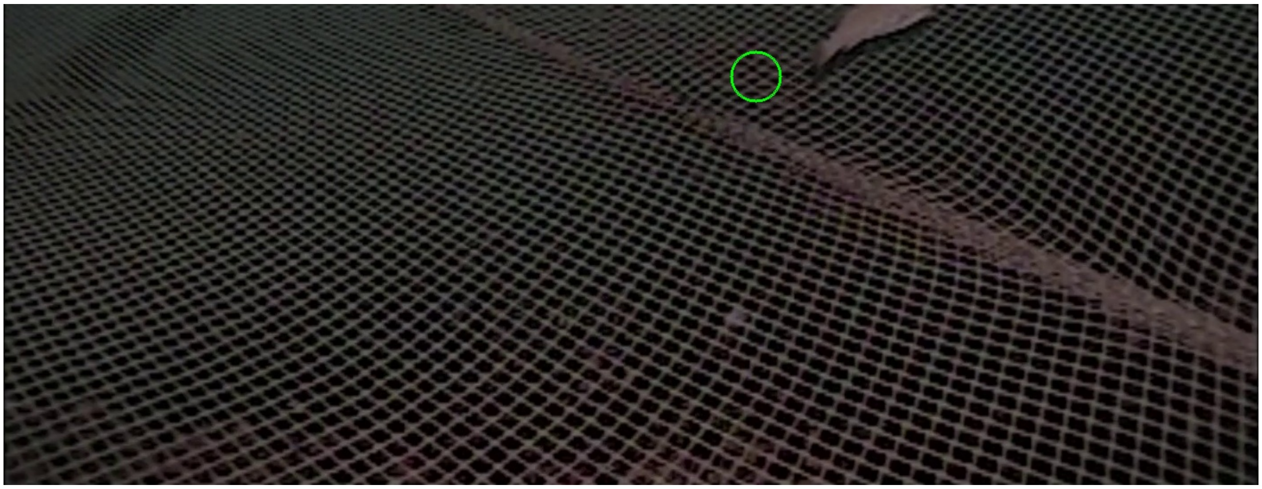
Frame Id	29042
Time	00:16:41
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



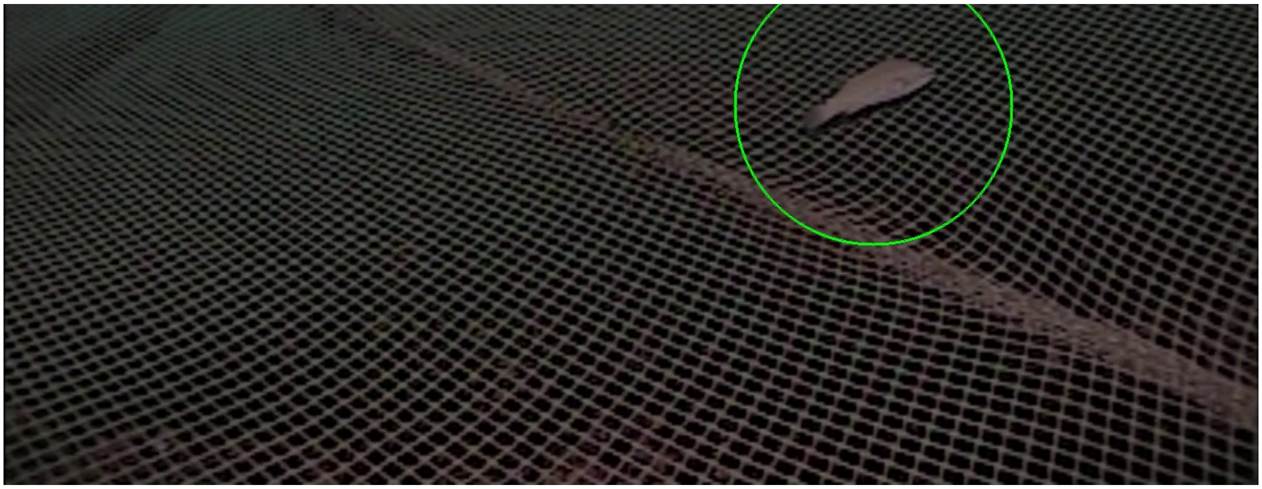
Frame Id	29070
Time	00:16:42
Percentage Analzed	80
Object Analysis (Hits)	2
Mask Analysis (Hits)	1
Knot Analysis (Hits)	0

Back to top



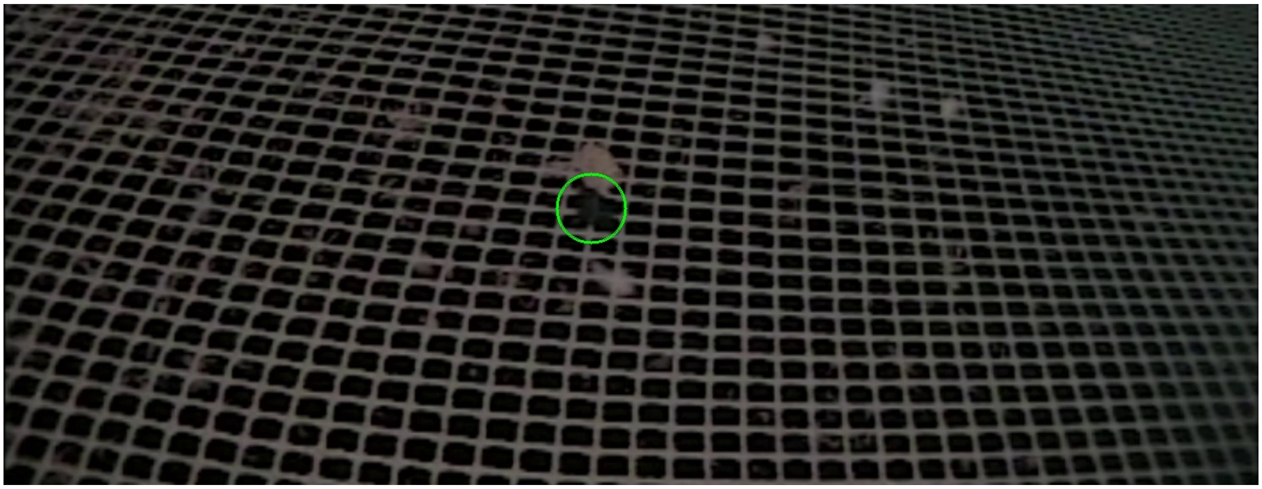
Frame Id	29350
Time	00:16:52
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



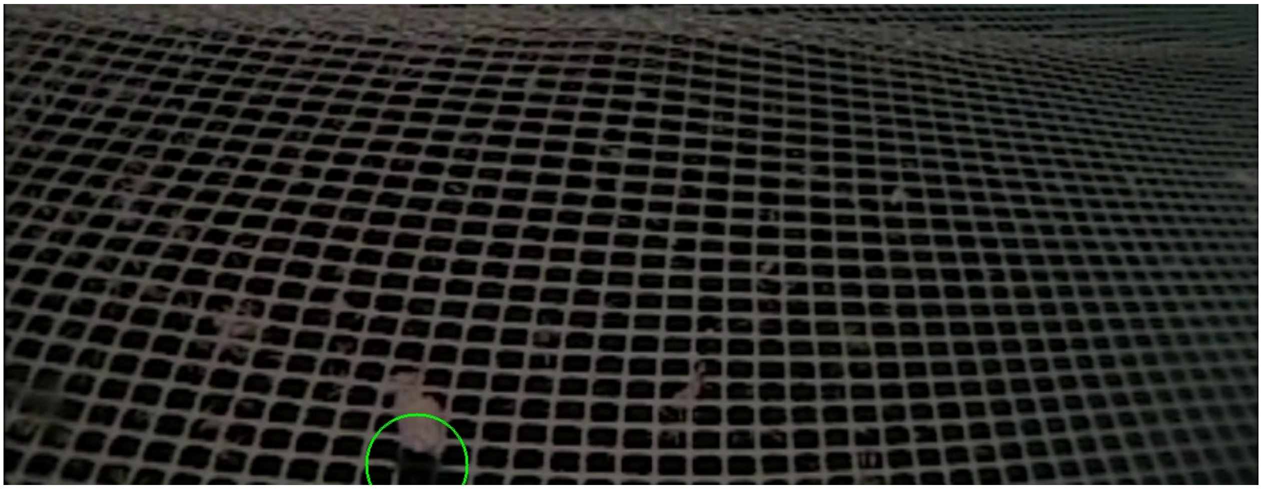
Frame Id	29378
Time	00:16:53
Percentage Analzed	40
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



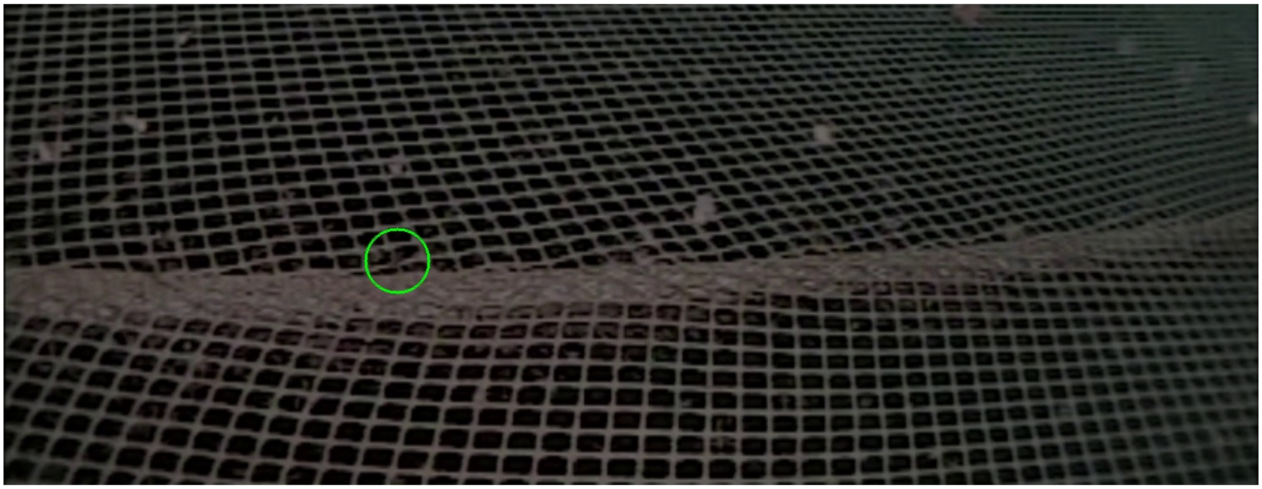
Frame Id	30456
Time	00:17:30
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



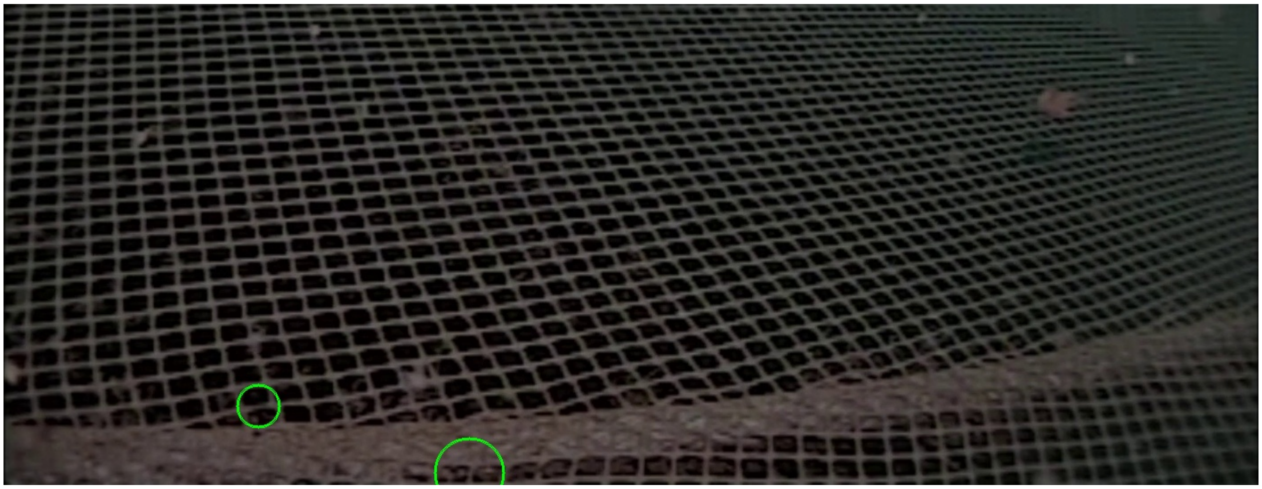
Frame Id	30470
Time	00:17:30
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



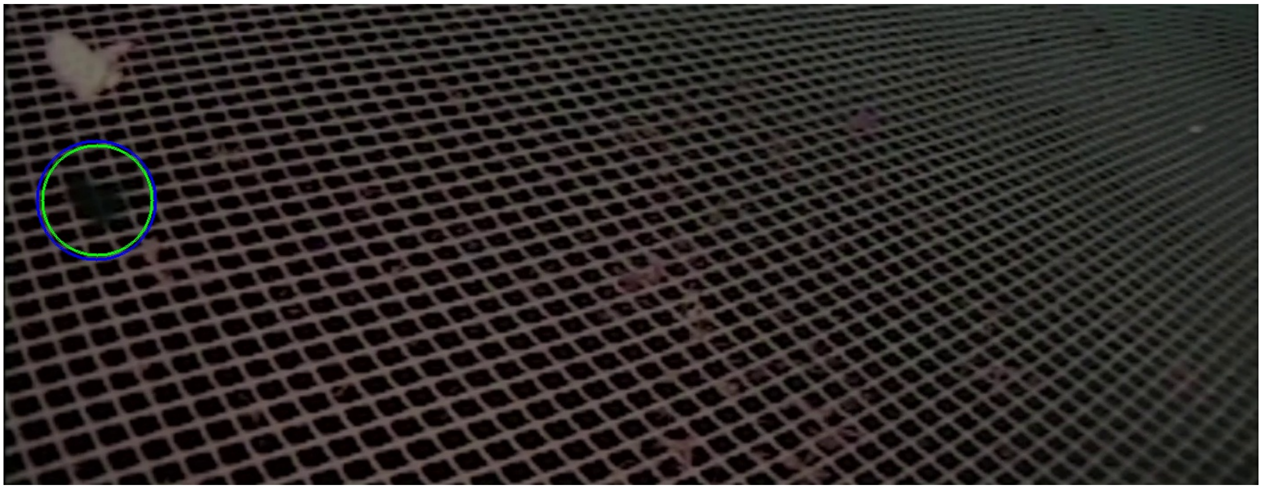
Frame Id	30498
Time	00:17:31
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



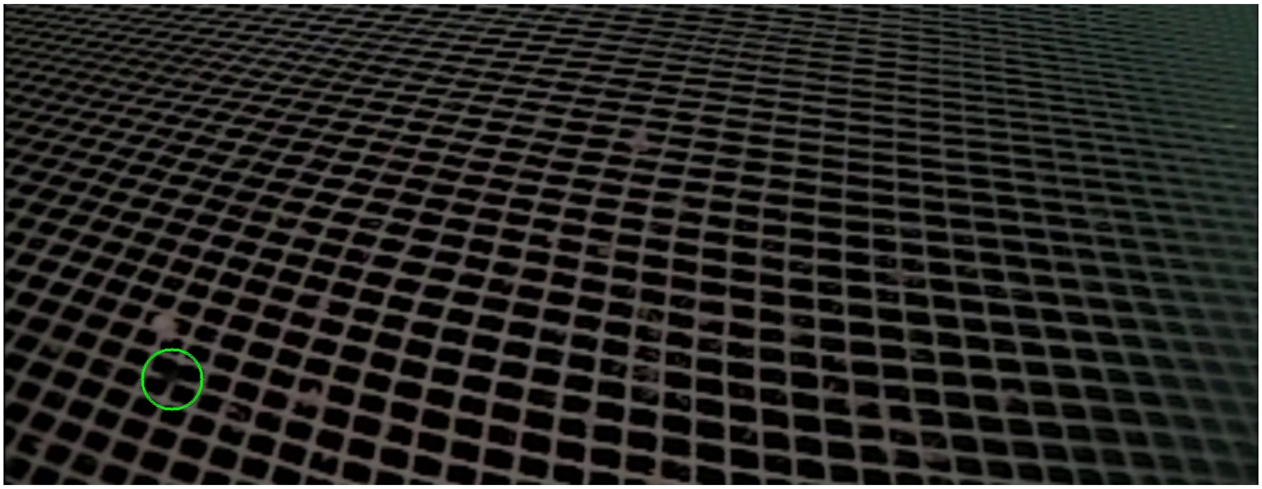
Frame Id	30512
Time	00:17:32
Percentage Analzed	40
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



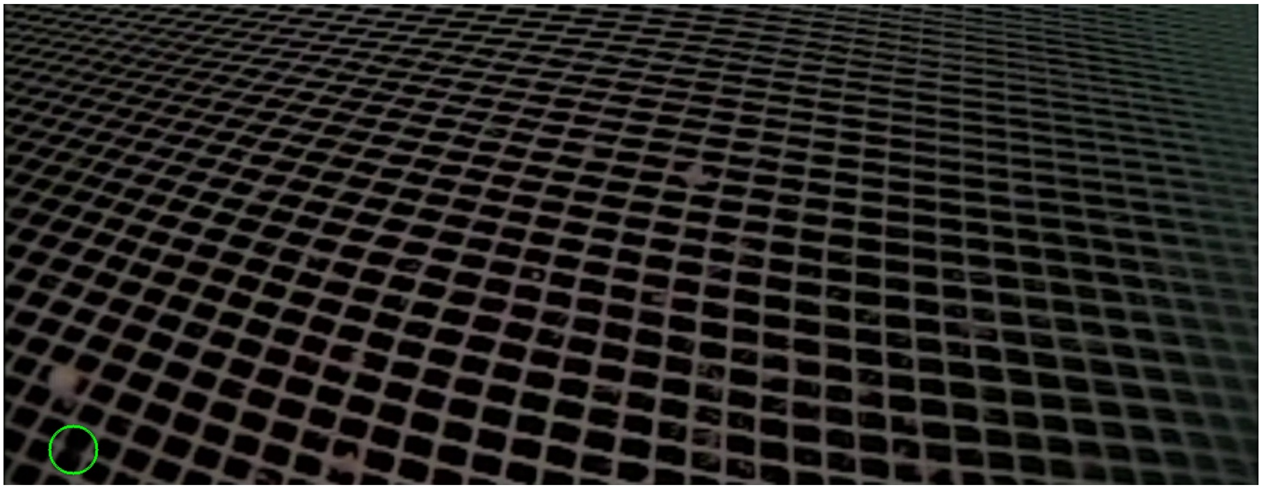
Frame Id	30736
Time	00:17:39
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	1
Knot Analysis (Hits)	0

Back to top



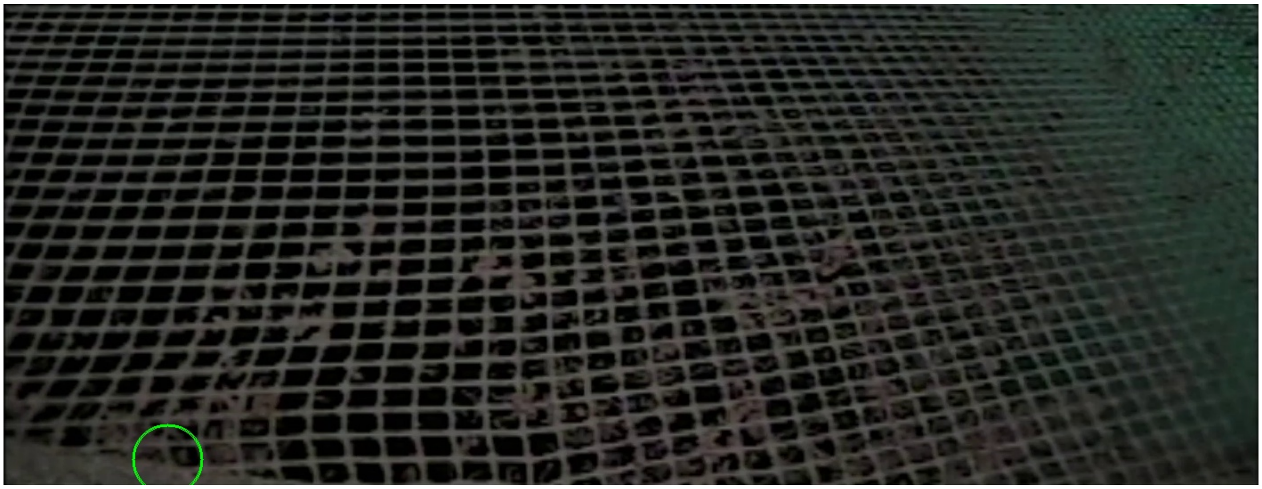
Frame Id	30876
Time	00:17:44
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



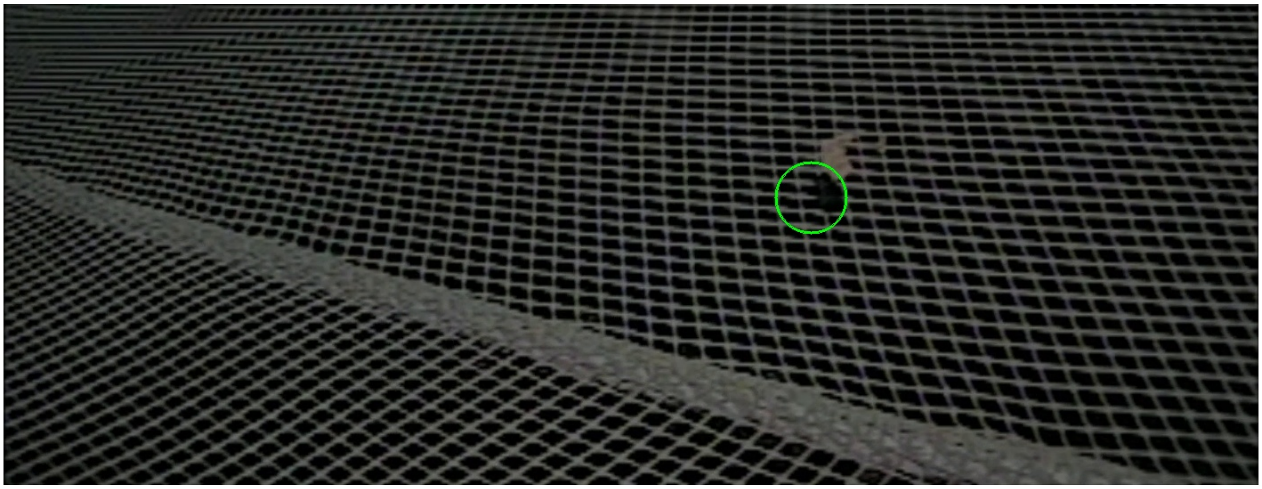
Frame Id	30890
Time	00:17:45
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



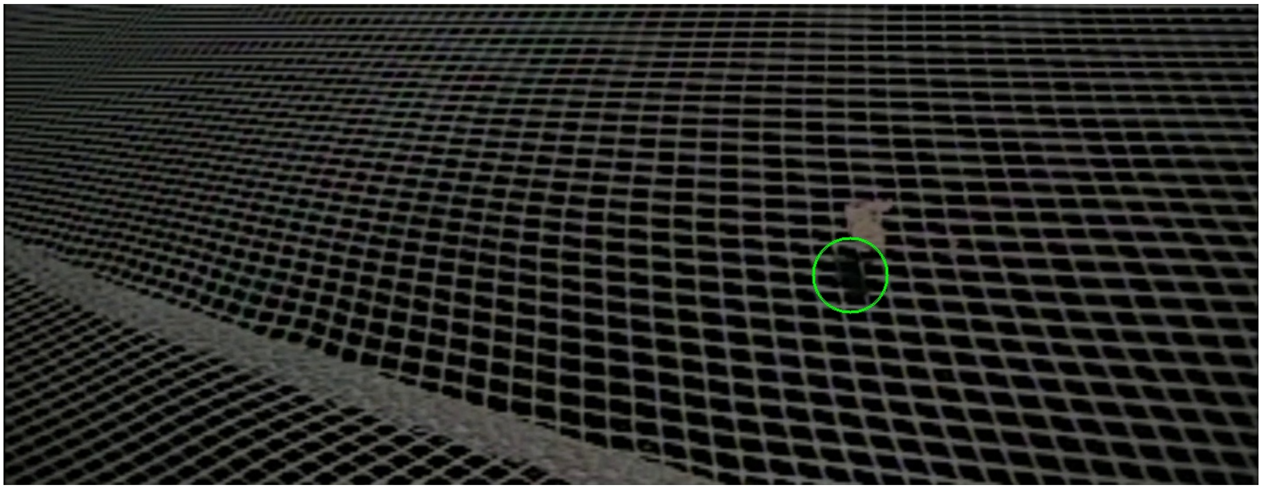
Frame Id	31114
Time	00:17:52
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



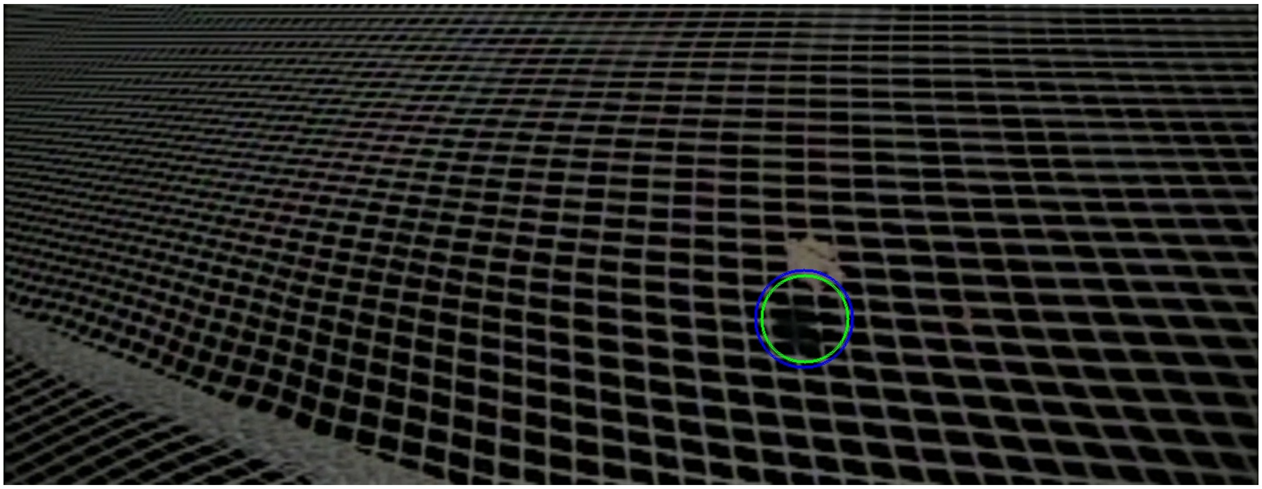
Frame Id	31534
Time	00:18:07
Percentage Analzed	40
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



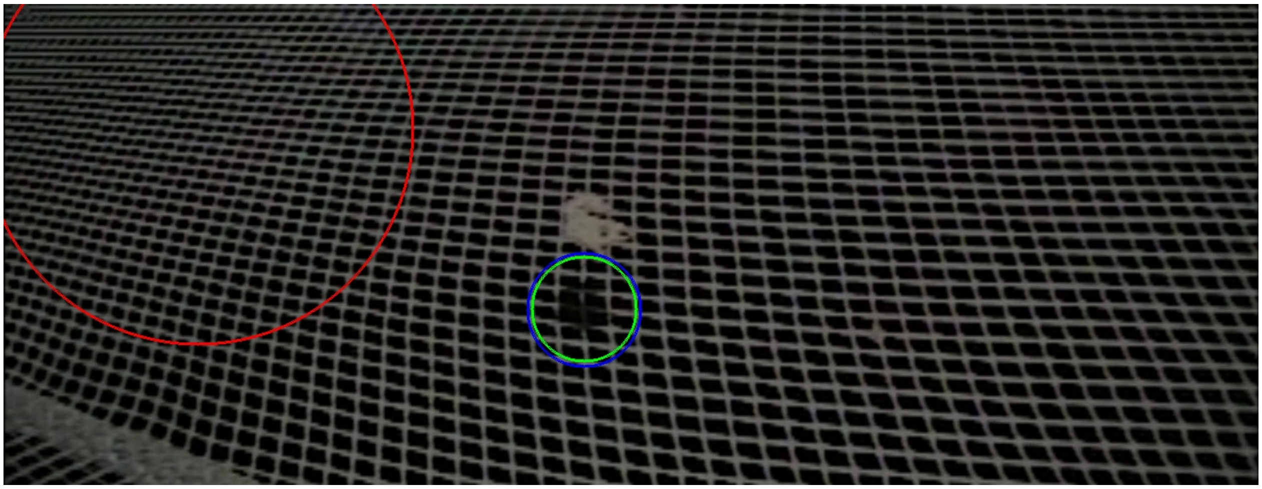
Frame Id	31548
Time	00:18:07
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



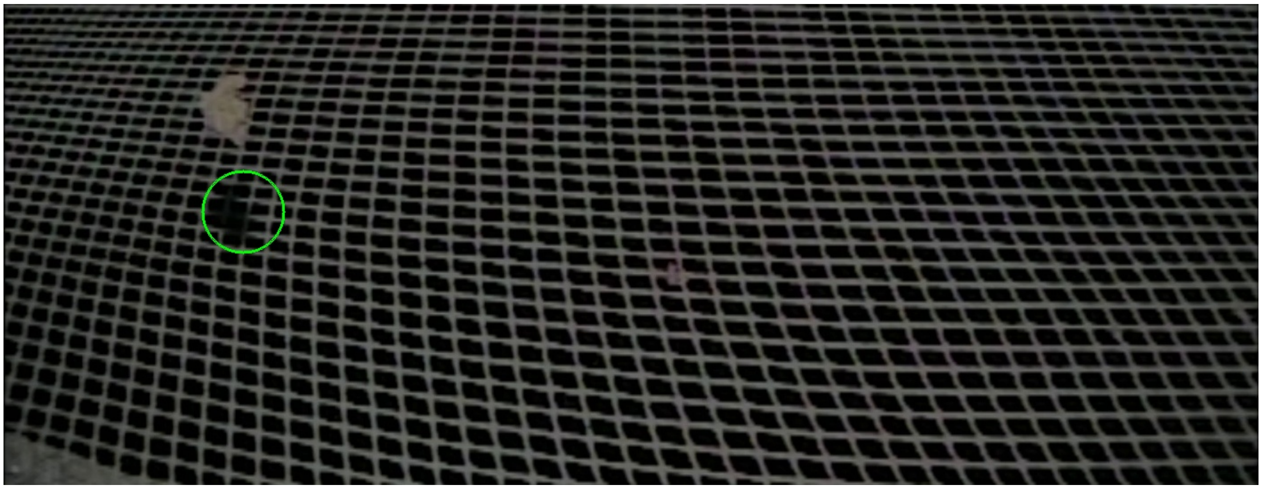
Frame Id	31562
Time	00:18:08
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	1
Knot Analysis (Hits)	0

Back to top



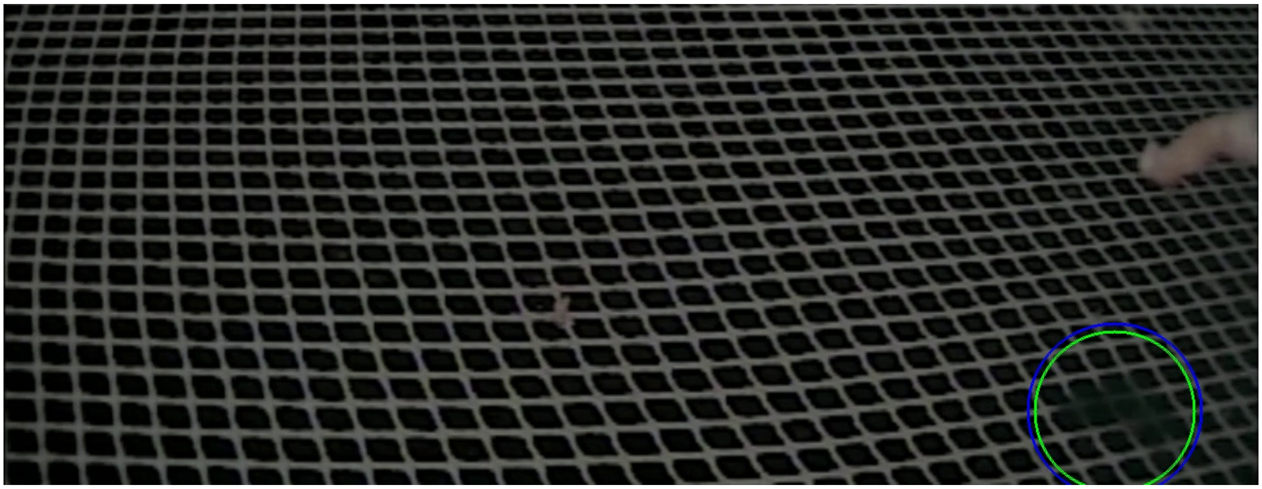
Frame Id	31576
Time	00:18:08
Percentage Analzed	80
Object Analysis (Hits)	1
Mask Analysis (Hits)	1
Knot Analysis (Hits)	1

Back to top



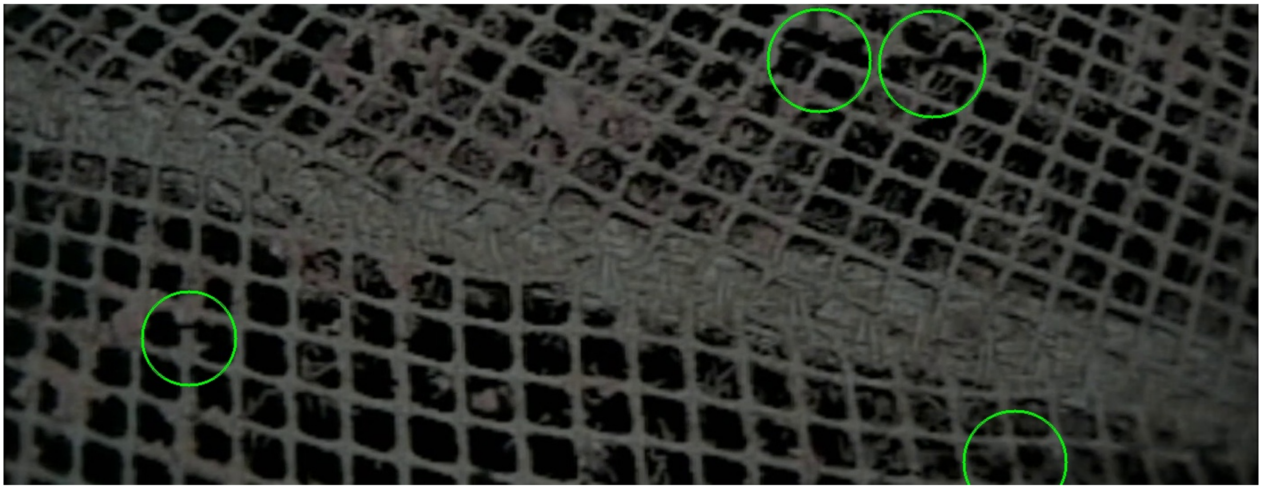
Frame Id	31590
Time	00:18:09
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



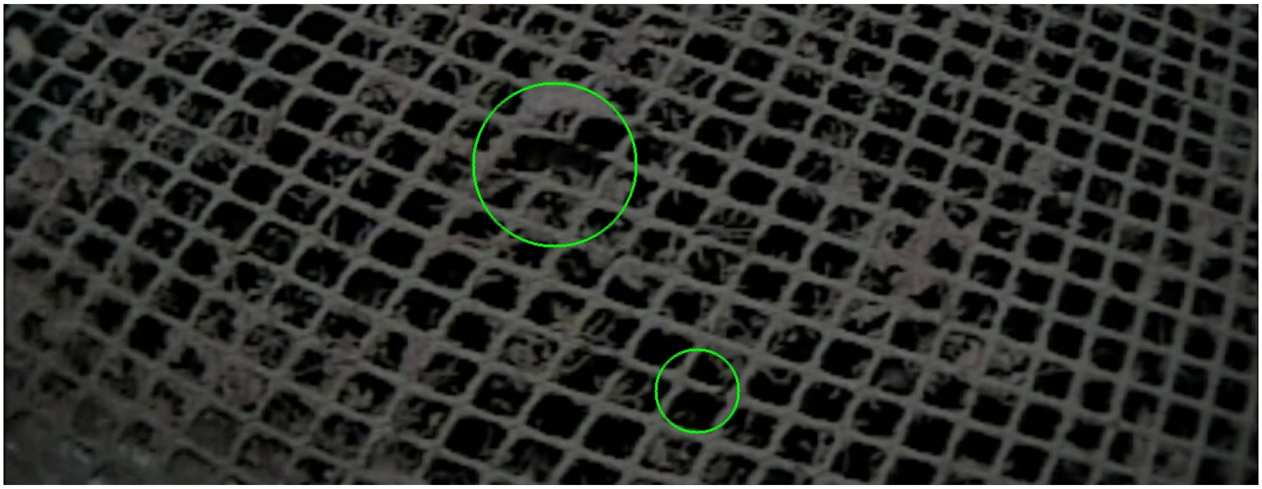
Frame Id	31674
Time	00:18:12
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	1
Knot Analysis (Hits)	0

Back to top



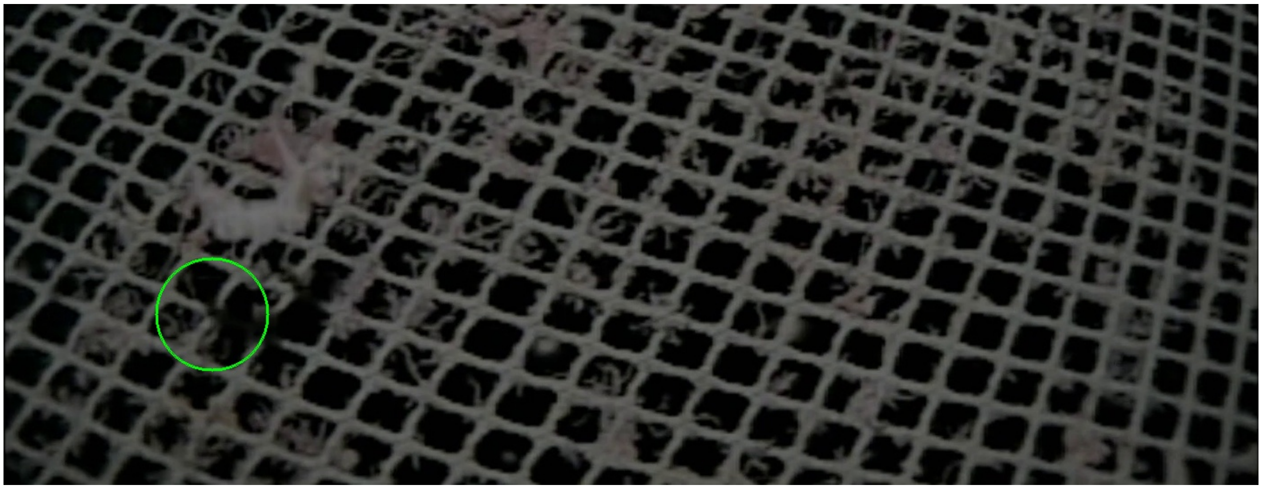
Frame Id	33438
Time	00:19:13
Percentage Analzed	80
Object Analysis (Hits)	4
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



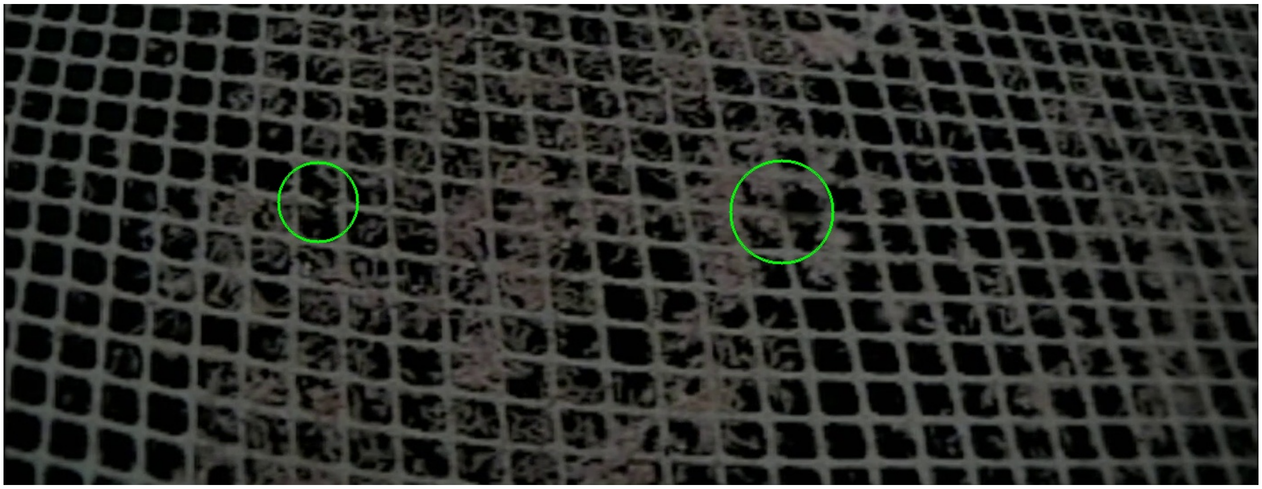
Frame Id	33452
Time	00:19:13
Percentage Analzed	60
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



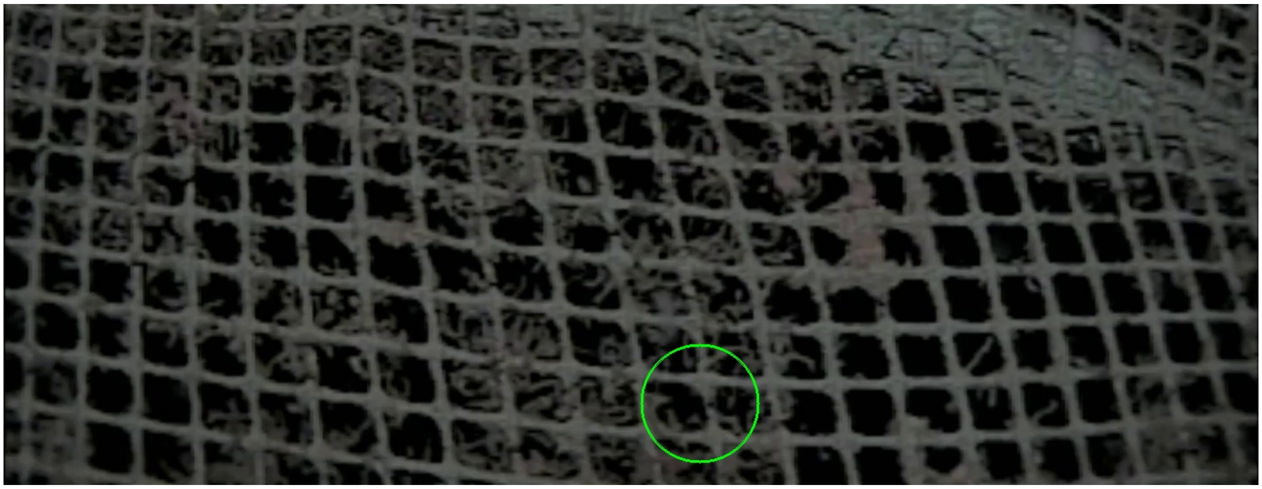
Frame Id	33466
Time	00:19:14
Percentage Analzed	40
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



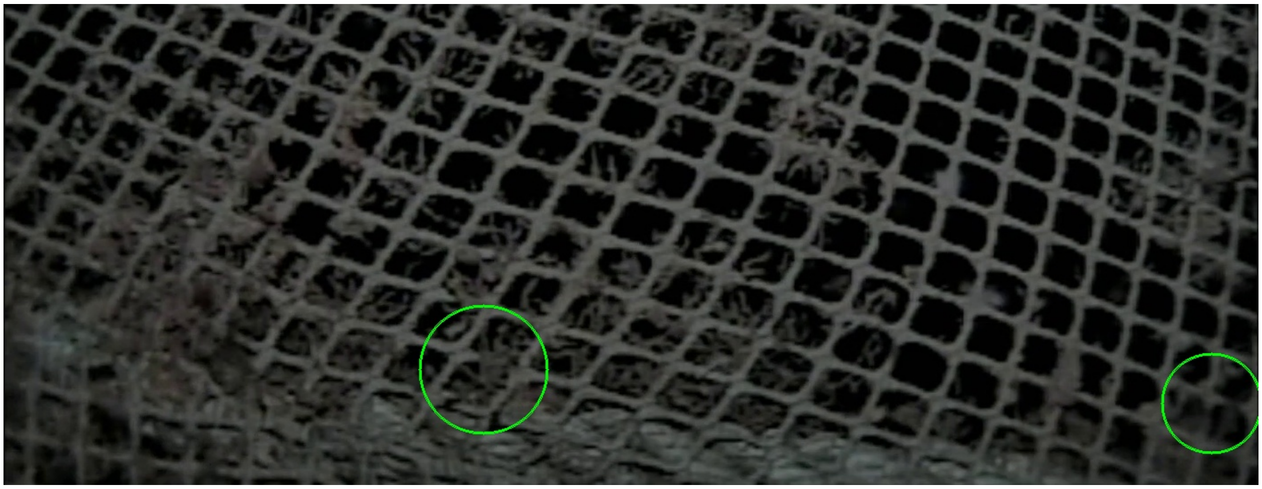
Frame Id	35468
Time	00:20:23
Percentage Analzed	100
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



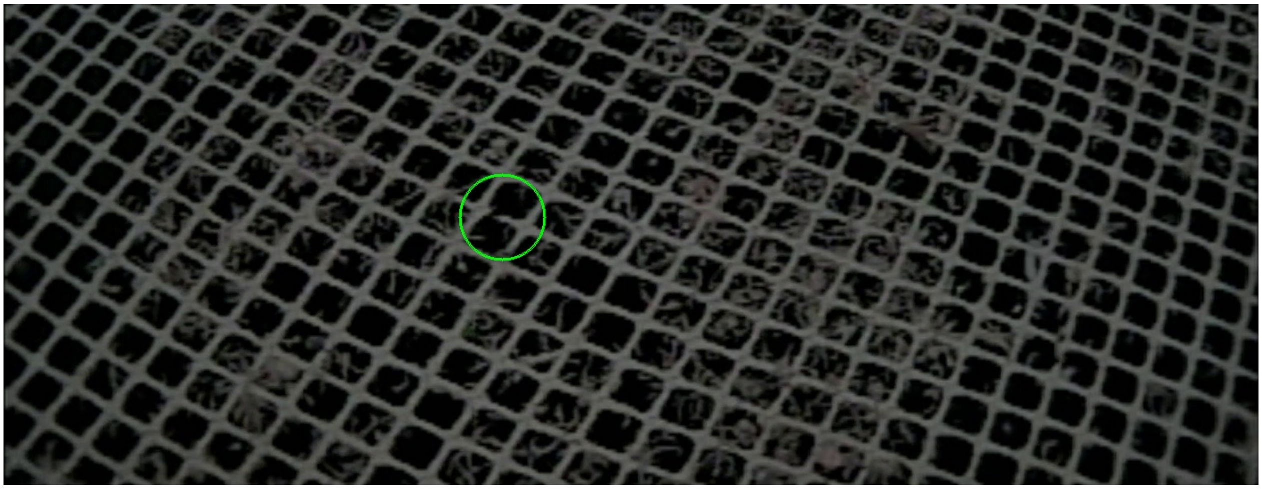
Frame Id	35524
Time	00:20:24
Percentage Analzed	60
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



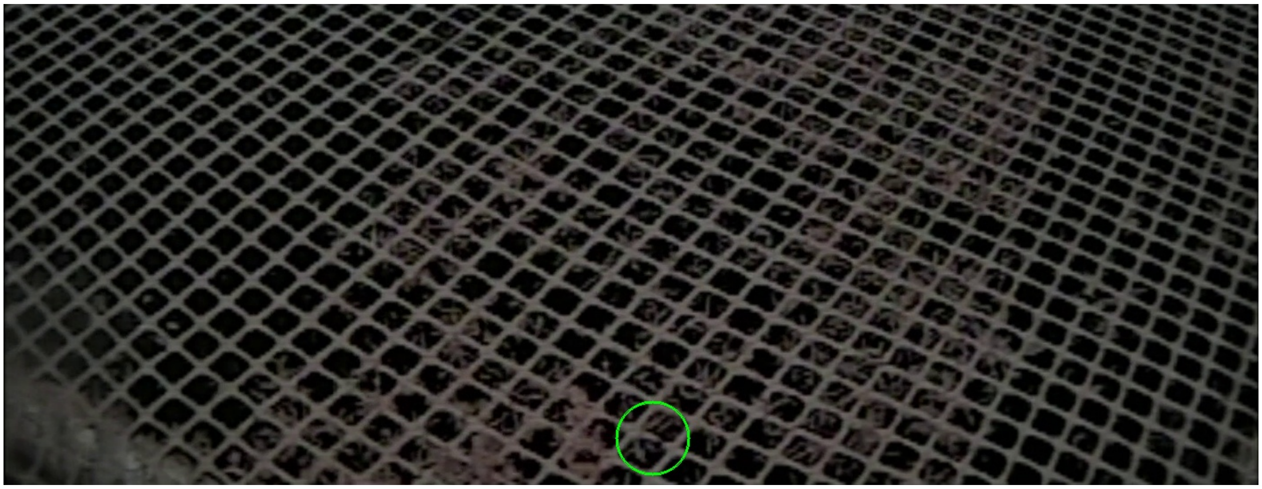
Frame Id	35538
Time	00:20:25
Percentage Analzed	80
Object Analysis (Hits)	2
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



Frame Id	35566
Time	00:20:26
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

Back to top



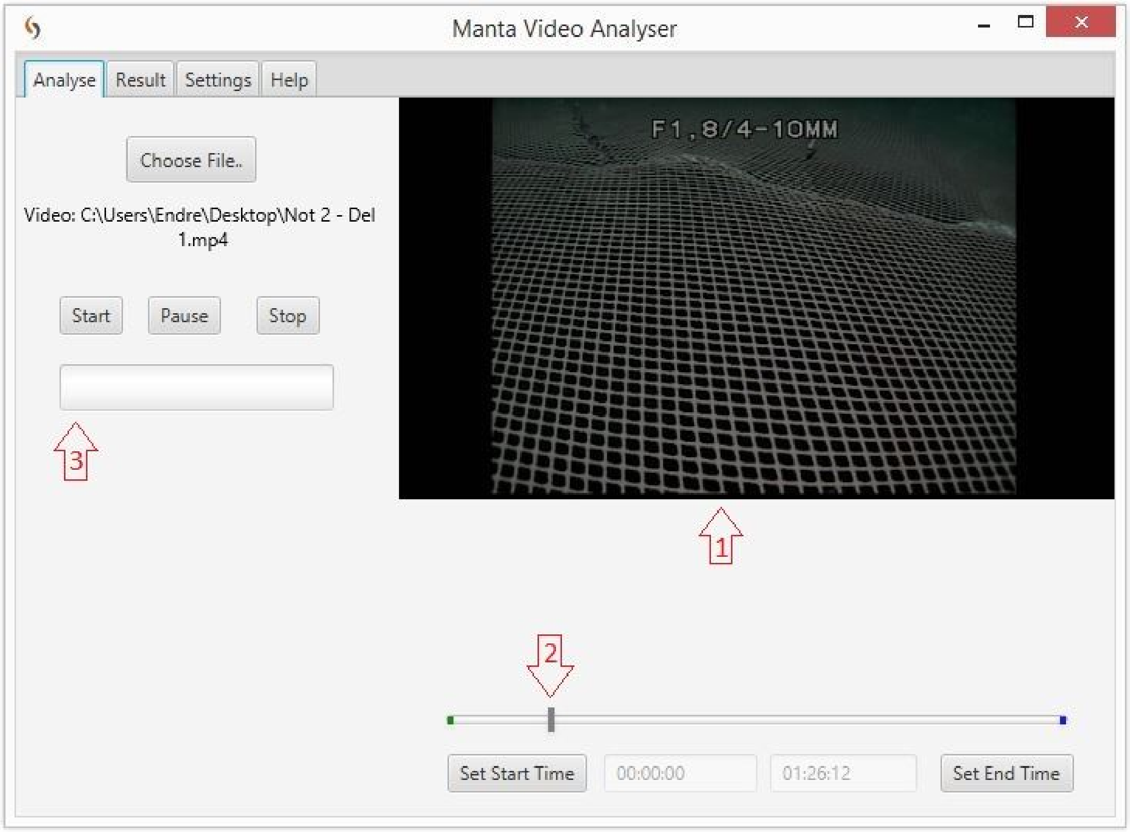
Frame Id	35664
Time	00:20:29
Percentage Analzed	100
Object Analysis (Hits)	1
Mask Analysis (Hits)	0
Knot Analysis (Hits)	0

User guide - Manta Video Analyser

The program consists of three different tabs

- Analyse tab – To analyse new videos
- Result tab – To manage results of analyzes
- Settings tab – To adjust the analysing parameters

The Analyse tab:



Video: C:\Users\Endre\Desktop\Not 2 - Del 1.mp4

00:00:00

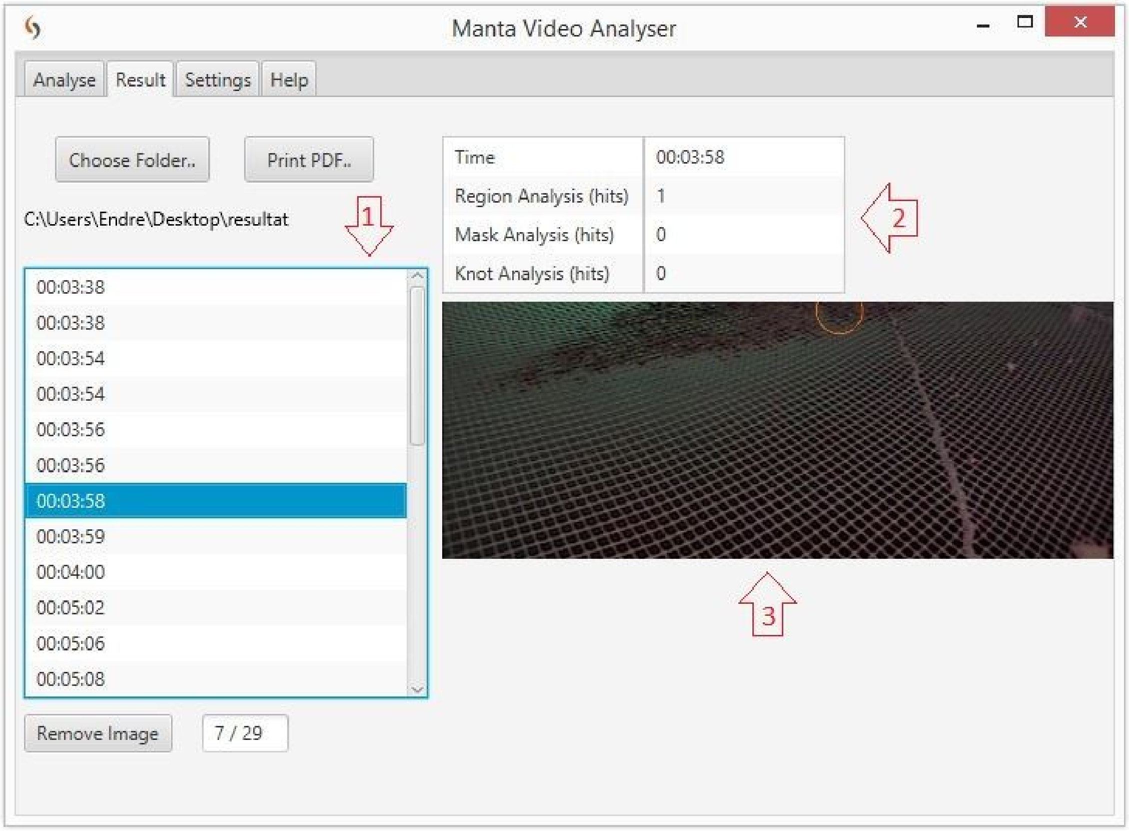
01:26:12

The analysing tab is is the tab that is used to start and control an ongoing analysis. Before a video is chosen, the “Choose file” button is the only functionality that is available in the analysis tab. By clicking the “Choose file” button, the user gets the opportunity to choose a video-file to upload. As soon as a video-file is successfully uploaded, all the other functionality on the page is activated.

Before the user starts an analysis, there is an opportunity to adjust which part of the video to be analysed. As default the whole video is chosen, but this can easily be adjusted by using the time-slider (pointer nr.2) and the set-time buttons. As the time-slider is moved, the preview window (pointer nr.1) will update the preview with pictures from the corresponding time in the video. This makes it easier for the user to find a special time in the video. When the user has moved the time slider to the wanted start-time, he can set the start-time by clicking the “Set Start Time” button. The start-time indicator will then be updated with the new start-time. The end time is set in the same way, by moving the time-slider to wanted end time, and using the “Set End Time” button to update end-time indicator.

The “Start” button is used to start an analysis. When the “Start” button is pressed, a popup-window lets the user choose a destination folder as file location for the results. The results should be stored in an empty folder (for example a new folder in a folder containing previous results), so the user is informed if previous results are detected in the destination folder. The user can then choose to overwrite previous results, or choose a new destination folder. As long as the program is analysing, the progress is displayed in the progress-bar (pointer nr.3), and the preview window shows samples from the video being analyzed. The program also displays time used and estimated remaining time for the analysis to complete. This information shows up text-based below the progressbar when the analysis is started. The user also has the opportunity to pause or stop the analysis by using the “Pause” and “Stop” button. When the analysis is finished, a popup-window asks the user if he wants to upload the results directly to the results tab. If this is declined, the user has to choose and upload the result file to the result-tab manually. Previous results can also be selected manually.

The Result tab:



C:\Users\Endre\Desktop\resultat

Time	00:03:58
Region Analysis (hits)	1
Mask Analysis (hits)	0
Knot Analysis (hits)	0

00:03:38
00:03:38
00:03:54
00:03:54
00:03:56
00:03:56
00:03:58
00:03:59
00:04:00
00:05:02
00:05:06
00:05:08

7 / 29

The result tab offers the user of the program a chance to study the results of analyses. Before a result-folder is chosen, the “Choose folder” button is the only functionality that is available in the result tab (unless the user has loaded results directly after performing an analysis). By clicking the

“Choose folder” button, the user gets the opportunity to choose a result folder to upload. As soon as a folder is successfully chosen, all the other functionality on the page is activated.

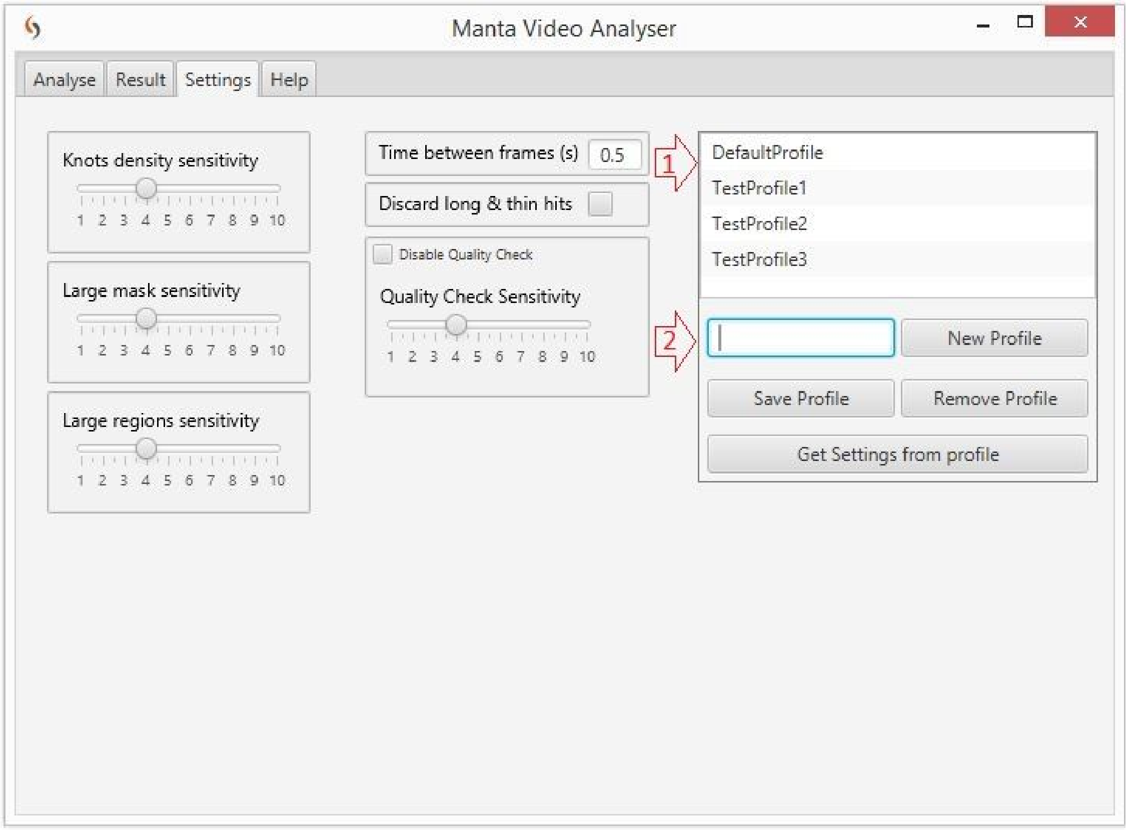
The frame-list (pointer nr.1) is filled as soon as the result folder is chosen and opened. In this list, the user can see all the frames from the current analysis that contains any hit. By clicking through the frame-list, the user can manually inspect each frame by watching the preview window (pointer nr.3) where the frame is presented. Each hit in the frame is marked by a circle which is placed exactly on the position where the hit is done. The circles uses color-code to tell which of the three analysis has done the hit. The information-list (pointer nr.2) shows the actual frames time in the video, and how many hits each of the analyses has detected in the frame.

The color code for the circles are:

- Green - Large objects analysis
- Blue - Large mask analysis
- Red - Knots detection analysis

Eventual incorrect hits can be removed by using the “Remove image” button. This button removes the hit, so that the frame is not included when the PDF file is created. The “Print PDF” button can be used to create a PDF document that contain all the information about the analysis. When the “Print PDF” button is pressed, a popup-window gives the user opportunity to add a comment that will be added in the PDF file. The user also has to type in a name and a location for the document to be saved in the popup-window.

The Settings tab:



Manta Video Analyser

Knots density sensitivity: 4

Large mask sensitivity: 4

Large regions sensitivity: 4

Time between frames (s) 0.5

Discard long & thin hits □

□ Disable Quality Check

Quality Check Sensitivity: 4

DefaultProfile
TestProfile1
TestProfile2
TestProfile3

In the setting tab, it is possible to fine-tune the analysis. Each of the three analysis functions in the program has a sensitivity setting that can be individually adjusted. The quality check has its own sensitivity setting. It is also possible to activate/ deactivate the entire quality check. It is possible to activate / deactivate the function that discards long and thin hits (because such hits are usually false hits). In addition, it is possible to regulate how long period there should be between each of the frames analysed. (The Manta does not move too quickly when washing, so it is not necessary to analyse every frame in the video to cover the entire net)

In order to easily change these settings, the group has created functionality to create and store entire profiles with settings. In an easy way, it is possible to create and manage settings profiles with a desired name. When the program is adjusted with desired settings values, it is possible to save these settings to the desired profile. The next time the program is started to do an analysis, one can easily retrieve the desired settings from the saved profiles. The different settings-profiles is shown in the profile list (pointer nr.1)

To create a new settings profile, the user has to write wanted profile name in the textfield (pointer nr.2), and press the "New Profile" button. Then the new profile will appear in the settings-profile list. If the user is trying to add a new profile with the same name as an existing profile, the new profile will replace the old one. The user can click on a profile in the profile list to make the profile active. The "Get Settings from profile" button updates the current settings according to the settings saved in the active profile. The "Remove Profile" button removes the active profile and all its settings completely. By clicking the "Save Profile" button, the active settings-profile is updated according to the current settings.

The advantage of making such a solution with profiles is that it is very quick and easy to change settings. The group has experienced during the testing that it is a great advantage to be able to adapt the settings to different videos. For example, a video from a cage with a lot of grouse can require a more sensitive quality check, and less sensitive analysis methods than a clean one. Light conditions and camera focus are also parameters that will affect which settings are most ideal. By using settings profiles, one can easily switch from a set of settings that suit one type of videos, to another profile that fits a different type of videos. This makes the program very user-friendly.

Knot Density Analysis

The knot density analysis algorithm uses a skeleton input image to find branch points (knots), and then uses the distance transform to find the density of knots. Any areas with too low density of knots is labeled a hit.

First, the branch points are found and printed on an empty image of the same size as the skeleton input image. The analysis uses a 5x5 neighborhood method to find the branch points. When the branch points are found, the analysis uses the distance transform on the branch point image. Then the distance transform image is thresholded to find the brightest areas (the areas with the lowest density of knots) How big an area has to be, to be considered as a hit, can be adjusted by changing the Knot Density Analysis sensitivity between 1-10 in the settings tab. When the sensitivity value is increased, potential hits has to be bigger to be considered a hit.

The Knot detection analyse is quite good on both good and less clear images. The disadvantage of this analysis is that it tends to overlook smaller holes in nets that have a lot of fouling. The analysis can sometimes detect knots in the fouling, this can contribute to failure.This happens especially if there is a lot of fouling in the area around small holes. Another source of error for this analysis is narrow holes. An example of this could be if masks are cut without the density of knots being seriously affected.

Large Mask Analysis

The large mask analysis algorithm attempts to find hits by looking at the average mask area of all the net masks, and then checking if any of the masks are too large to be a regular mask. The algorithm takes a skeleton image as input, and finds all the contours of masks in the input image. A contour-list is made, and all the contours that is found, is added to this list.

Further, the average mask area is found from the contour-list, by adding the area of all contours and dividing by the total number of contours. To determine if any of the masks is a hit, each mask is then checked again. This is done by iterating through the list, with each area being compared to the average area. How much bigger an area has to be than the average area to get considered as a hit, can be adjusted by changing the Large Mask Analysis sensitivity variable between 1-10 in the settings tab. The sensitivity value corresponds to how many times bigger than the average mask area an area has to be, to be considered as a hit. Example: if the sensitivity value is 4, then an object has to have 4 times bigger area than the average object area in the frame to be considered as a hit.

The large mask analysis is a very accurate analysis of frames with the net clearly visible. This analysis is able to detect holes all the way down to a single line being broken. However, due to the fact that it is based on the skeleton image, it depends on an accurate skeleton representation of the frame. If the lines in the net are vague or too close to each-other, or not visible or there at all, it is possible to get a false skeleton representation of that part. In some cases, this leads to false hits being created by the skeletonization algorithm. Because of this, the large mask analysis does not work well on parts of a frame where the net is not clearly distinguishable from the background. When the skeleton image is good, the large mask analysis has a very good accuracy.

Large Regions Analysis

The large regions analysis algorithm attempts to find hits by looking at the average maskarea of all the net masks, and then checking if any of the masks are too large to be a regular mask. The algorithm takes a black-white image as input, and finds all the contours of masks in the input image. A contour-list is made, and all the contours that is found, is added to this list.

Further, the average mask area is found from the contour-list, by adding the area of all contours and dividing by the total number of contours. To determine if any of the masks is a hit, each mask is then checked again. This is done by iterating through the list, with each area being compared to the average area. How much bigger an area has to be than the average area to get considered as a hit, can be adjusted by changing the Large Regions Analysis sensitivity value between 1-10 in the settings tab. The sensitivity value corresponds to how many times bigger than the average mask area an area has to be, to be considered as a hit. Example: if the sensitivity value is 4, then an object has to have 4 times bigger area than the average object area in the frame to be considered as a hit.

Large object analysis works better than the large mask analysis on frames where the net is not always clearly distinguishable from the background. This analysis detects quite well, without finding many false hits even though the image is unclear. But large object analysis is not as sensitive or accurate as large mask analysis on high quality images. Sometimes the Manta turns, so that the camera focuses far ahead. When the camera looks far ahead, it often sees areas with shadows because the light on the manta does not reach that long. When this happens, Large object analysis tends to detect the shadow as a hole. This is a weakness with this analysis.

FORPROSJEKT - RAPPORT

FOR BACHELOROPPGAVE

NTNU
Kunnskap for en bedre verden

TITTEL: FORPROSJEKTRAPPORT: VIDEO ANALYSE MANTA NET CLEANER

KANDIDATNUMMER(E):

DATO:	EMNEKODE:	EMNE:	DOKUMENT TILGANG:
28.01.2019	**IE303612**	**Bacheloroppgave**	- Åpen

STUDIUM:	ANT SIDER/VEDLEGG:	BIBL. NR:
AUTOMASJON	13/0	- Ikke i bruk -

OPPDRAGSGIVER(E)/VEILEDER(E):

Stranda prolog AS (v. Kjetil Osland Brekken og Jacob Hoseth) / Arne Gerhard Styve,
Saleh Abdel-Afou Alaliyat

OPPGAVE/SAMMENDRAG:

Vi er to automasjonsstudenter som skal jobbe med videoanalyse for Manta notvasker som vår bacheloroppgave. Oppgavegiver Stranda Prolog ble etablert i 1946 som et motorverksted, men driver i dag med logistikkløsninger til oppdrettsindustrien. De har utviklet en Manta notvasker som er en fjernstyrt undervannsfarkost som brukes til å rense noten i merder. Notvaskeren er utstyrt med flere videokameraer.
Oppgaven går ut på å benytte slik video fra Manta notvaskeren til å detektere eventuelle hull/ skader som kan finnes i noten. Vi ønsker også å fokusere på å lage en brukervennlig applikasjon for å hente videofiler, gjøre enkle tilpassinger til analysen og for eksempel legge inn en kommentar til analyseresultatene i etterkant.

Denne oppgaven er en eksamensbesvarelse utført av student(er) ved NTNU i Ålesund.

INNHOLD

1 INNLEDNING

Vi er to automasjonsstudenter som skal jobbe med videoanalyse for Manta notvasker som vår bacheloroppgave. Vi valgte denne oppgaven fordi vi ser på videoanalyse som et spennende og utfordrende emne. Vi ser også at video og bildeanalyse er svært aktuelt og brer om seg innenfor automasjon i dag. Muligheten til å samarbeide med en bedrift med lang historie og som leverer kvalitetsprodukter innenfor oppdrettsindustrien var også en positiv faktor i valget av oppgaven. Vi syntes begge at det var ekstra motiverende å arbeide med en løsning til et eksisterende produkt.

Oppgavegiver Stranda Prolog ble etablert i 1946 som et motorverksted, men driver i dag med logistikkløsninger til oppdrettsindustrien. Bedriften har i dag 60 ansatte og er hovedsakelig basert i Kristiansund. De har fremdeles virksomhet i de opprinnelige lokalene på Langøyneset på Averøya. De har utviklet en notvasker for å rense not/ merder for oppdrettsnæringen ved bruk av høytrykksspyling. Dette apparatet kan kjøres under vann fra en ekstern operatørstasjon. Føreren manøvrerer Mantaen rundt nede i merdene ved hjelp av en videostrøm fra kamera som er montert på Mantaen. Oppgaven går ut på å benytte slik video fra Manta notvaskeren til å detektere eventuelle hull/ skader som kan finnes i noten. Vi ønsker også å fokusere på å lage en brukervennlig applikasjon for å hente videofiler, gjøre enkle tilpassinger til analysen og for eksempel legge inn en kommentar til analyseresultatene i etterkant.

I dag blir eventuelle hull/ skader i merdene bare oppdaget dersom føreren legger merke til det på videobildene mens han kjører Mantaen. Dersom vi lykkes med vår oppgave vil slike hull/ skader kunne oppdages automatisk ved å benytte vår applikasjon.

2 BEGREPER

Manta net cleaner/Manta notvasker – Apparat som er utviklet av Stranda prolog AS for å rense not/merder for oppdrettsnæringen ved bruk av høytrykksspyling. Denne blir styrt fra et moderfartøy som senker den ut i merder med en kran. Føreren opererer den fra et kontrollrom ombord I moderfartøyet.

OpenCV (Open source computer vision) - Et bibliotek med funksjoner til bildebehandling. Skrevet i C/C++, men kan brukes i bl.a. Python/Java.

Netbeans - IDE (integrated development enviroment) for utvikling av Java-program.

Scrum - Rammeverk for utvikling av programvare.

Confluence er en åpen og delt nettbasert arbeidsplattform som er bygget for samarbeid om prosjekt i grupper. Her kan man blant annet lage møtenotater og statusrapporter.

Jira er også en nettbasert tjeneste som skal være et hjelpemiddel i fleksibel prosjektledelse. Her kan man opprette sprinter og legge inn arbeidsoppgaver i prioritert rekkefølge.

3 PROSJEKTORGANISASJON

3.1 Prosjektgruppe

Studentnummer(e)
476136
997499

3.1.1 Oppgaver for prosjektgruppen - organisering

Siden vi er bare to personer i denne gruppen blir ikke gruppeorganiseringen så avansert. Vi setter opp en prosjektleder og en sekretær formelt sett, men vi ser for oss at vi kommer til å samarbeide tett og fordele arbeidsoppgavene mellom oss etter hvert. Endre S. Skeide er innsatt som prosjektleder, Vebjørn Bjørlo-Larsen er sekretær.

3.1.2 Oppgaver for prosjektleder

Ansvarsområde:

- Ha ansvar for at gruppen jobber godt sammen.
- Ha ansvar for at gruppen har framgang og ikke står fast.
- Ha ansvar for å avtale møter med veiledere og/ eller bedrift/oppdragsgiver.

Arbeidsoppgaver:

- Se til at tidsfrister blir holdt.
- Se til at gruppen samarbeider godt.
- Se til at oppgavene blir godt fordelt innad i gruppen.
- Motivere gruppen til å holde god arbeidsmoral.

3.1.3 Oppgaver for sekretær

Ansvarsområde:

- Ansvar for møtenotater.

Arbeidsoppgaver:

- Komme med gode innspill.
- Ha en positiv holdning.
- Skrive møtenotater.
- Bidra til godt samarbeid og god arbeidsmoral.

3.2 Styringsgruppe (veileder og kontaktperson oppdragsgiver)

Arne Styve og Saleh Abdel-Afou Alaliyat er våre veiledere i denne bacheloroppgaven. Arne har god kunnskap om programmering, Saleh har god kunnskap om bildebehandling.

Kjetil Osland Brekken og Jacob Hoseth er våre kontaktpersoner fra vår oppdragsgiver Stranda Prolog AS. Underveis i arbeidet med oppgaven er det Kjetil vi vil komme til å ha direkte kontakt med, han vil deretter involvere Jacob når det er behov for det.

4 AVTALER

4.1 Avtale med oppdragsgiver

Vi har avtalt med oppdragsgiver at vi skal fokusere på å detektere hull/ skader i merdene, i tillegg vi vil prøve å lage en brukervennlig applikasjon.

Oppdragsgiver vil prøve å skaffe godt test-materiale i form av videobilder som vi kan benytte oss av under utviklingen. Vi har avtalt å kommunisere fortrinnsvis med Kjetil, han vil selv involvere Jacob dersom det er nødvendig.

4.2 Arbeidssted og ressurser

Vi vil i hovedsak komme til å jobbe på NTNU, der har vi lokaler, strøm og internett. Vi ser ikke for oss at denne oppgaven vil kreve særlige ressurser utover det. Vi tar sikte på å benytte egne datamaskiner, vertfall i første omgang.

Våre veiledere Arne og Saleh er tilgjengelige ressurspersoner. Særlig Saleh som har god kunnskap om bildebehandling vil nok være en viktig kontaktperson for oss framover.

Vi er innforstått med at bilder og videomateriale vi mottar, samt sensitiv informasjon om Manta notvasker ikke skal publiseres offentlig uten at Stranda prolog AS har godkjent det.

Vi har avtalt med veiledere at det skal skrives et møtereferat og lages en framdriftsrapport etter hvert møte. Vi har planlagt å holde et statusmøte annenhver uke. Dette møtet kommer normalt sett til å bli avholdt på fredager. I forkant av hvert av disse møtene skal vi skrive en statusrapport, slik at veilederne kan gjøre seg kjent med status i forkant av møtene.

4.3 Gruppenormer – samarbeidsregler – holdninger

Vi ønsker å legge følgende normer til grunn for arbeidet i gruppen og samarbeidet i prosjektet:

- Vi har planlagt å jobbe fast med prosjektet alle arbeidsdager fra 08:15 til 16:00. Normalt sett vil arbeidssted være på NTNU i Ålesund.

De dagene/ukene vi har obligatorisk undervisning i Industri 4.0 vil komme til å avvike fra denne normen.

- Gruppen er enige om at vi normalt sett vil prøve å oppnå våre mål for hver sprint, og er innstilt på å jobbe ekstra når det trengs.

- Gruppen er også enige om å bruke verktøyet JIRA til framdriftsplaner. Etter hvert statusmøte opprettes en ny "sprint" på to uker. Her skal vi sette opp oppgavene for perioden og fordele dem på gruppen. I løpet av sprinten skal oppgavene plasseres i "to do", "in progress" eller "done" i henhold til oppgavestatus. Etter hver sprint skal vi gå over arbeidsmengden og eventuelt gjøre nødvendige justeringer før neste sprint.

- Vi vil også ta i bruk wiki-plattformen Confluence til å legge ut statusrapporter, møtereferat, sprintoppgaver og annen informasjon om prosjektet. Den siste dagen i hver sprint skal vi skrive en statusrapport i Confluence hvor vi går gjennom framdriften i perioden. Her skriver vi hva vi har klart og ikke har klart, men legger vekt på hvorfor/hvorfor ikke vi fikk til en oppgave.

Som utøvere av en profesjon er det svært viktig for oss at det vi dokumenterer er noe vi selv mener er rett og noe som vi selv kan stå inne for. Det er viktig for oss at vi dokumenterer alle kildene våre godt, og at vi skiller tydelig mellom arbeid vi selv har jobbet frem, og arbeid som andre har jobbet frem. Vi ønsker å uttrykke oss på en faglig korrekt måte for å kunne formidle vårt arbeid mest mulig tydelig, slik at vi unngår eventuelle misforståelser eller uklarheter. Vi ønsker å innrette oss etter norsk lov, og prøve å tenke igjennom om vårt arbeid kan oppleves støtende for andre. Vi vil prøve å tenke igjennom hva vårt arbeid vil kunne føre til, hvordan vårt arbeid vil kunne komme til å misbrukes og hvilke følger det i eventuelt vil kunne få.

5 PROSJEKTBESKRIVELSE

5.1 Problemstilling - målsetting - hensikt

Den grunnleggende problemstillingen er at Stranda Prolog AS Ønsker at deres Manta notvasker skal inneholde funksjonalitet som gjør det enklere å oppdage og rapportere skader/ hull i merdene.

Målsetningen med oppgaven er at gruppen skal utarbeide en løsning på denne problemstillingen ved å benytte video/ bildebehandling av videostrøm fra kamera som allerede finnes på Manta notvaskeren.

Vi har satt opp følgende resultatmål:

- Hovedmålet er å utvikle en applikasjon som er i stand til å detektere skader/ hull i merdene, og rapportere det på en brukervennlig måte. Vi ser for oss å lage en applikasjon der bruker skal kunne velge en video og gjøre enkle innstillinger før applikasjonen analyserer videoen. Når analysen er fullført skal bruker kunne kontrollere eventuelle funn manuelt og lagre en kommentar.

- Det er et delmål å greie å lage programvare som kan hente inn video og dele den opp i bilder som vi kan utføre bildebehandling på.
- Det er et delmål å utvikle en programvare som kan detektere avvik i form av hull/ skader i noten i merdene.
- Det er et delmål å utvikle programvare som er i stand til å skille hull fra andre avvik/ gjenstander i nettet.
- Det er et delmål å utvikle en applikasjon som er i stand til å rapportere slike hull/skader til bruker på en brukervennlig måte.

Verdimålet slik vi ser det, er at vi får lært og praktisert mest mulig programmering og bildebehandling. Det er også et verdimål å få trening i å utføre et slikt prosjekt i samarbeid med veiledere og bedrift.

5.2 Krav til løsning eller prosjektresultat – spesifikasjon

Overfor oppdragsgiver skal vi i tillegg til å levere den programvaren vi utvikler, også levere dokumentasjon i form av brukerveiledning og bachelorrapporten når prosjektet er fullført.

5.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)

Vi har valgt å benytte Scrum som vår utviklingsmetode. Scrum er et rammeverk som passer godt for prosjekt som går ut på å utvikle programvare. Denne metoden går ut på å jobbe i sprinter på ofte 1-4 uker.

Gruppen og Veiledere har i fellesskap kommet frem til å jobbe i sprinter på to uker. Dette passer godt fordi vi har statusmøter med veilederne annenhver fredag. Vi vil anse hver periode fra et statusmøte og frem til neste statusmøte som en sprint. For hver sprint setter vi opp prioriterte oppgaver som vi ønsker å jobbe med. Når vi er ferdig med en sprint skriver vi en statusrapport der vi får frem hvor langt vi kom med de planlagte oppgavene, og hvorfor vi eventuelt ikke kom i mål.

Vi har valgt denne utviklingsmetoden fordi vi mener den er svært hensiktsmessig for arbeid med utvikling og testing av programvare. Dette er en utviklingsmetode som er kjent og mye brukt blant folk som jobber med å utvikle programvare. Styrken til denne metoden er at man planlegger fremgangen i mindre biter om gangen. Dette gjør det mulig å sette seg realistiske mål for hver sprint og korrigere arbeidsmengde og fordeling fra sprint til sprint underveis. Når man jobber med å utvikle programvare går mye av tiden til testing og feilsøking. Dette gjør det svært krevende om ikke umulig å beregne tidsbruk, derfor er det heller ikke hensiktsmessig å prøve å planlegge detaljert flere måneder frem i tid.

5.4 Informasjonsinnsamling – utført og planlagt

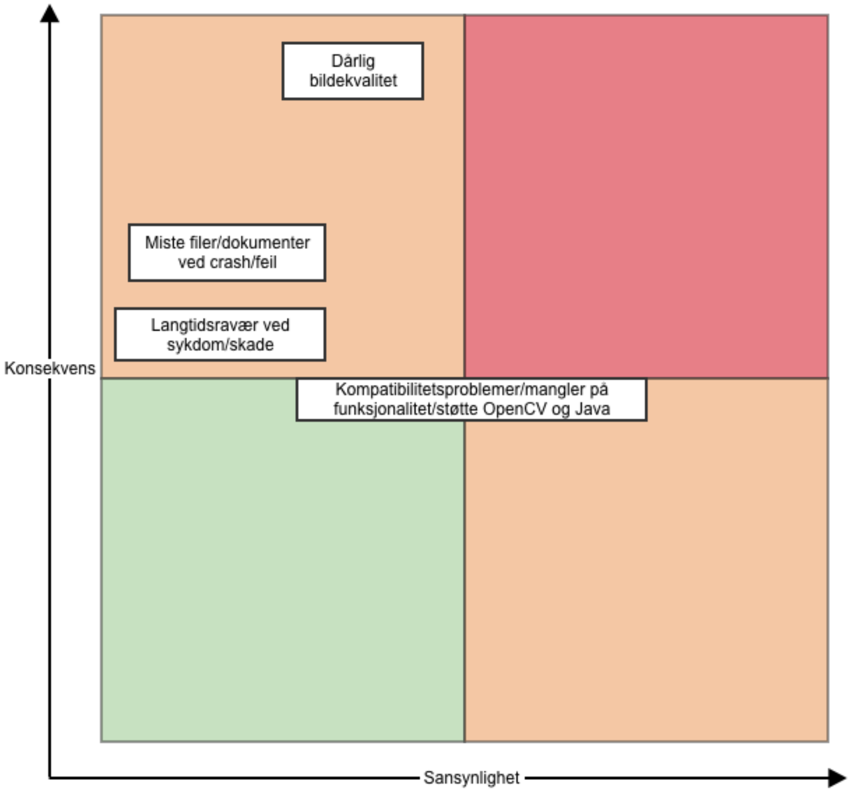
For å samle inn informasjon har vi hatt en god dialog med oppdragsgiver Stranda Prolog AS. Vi har allerede fått en del informasjon muntlig om Manta notvaskeren, hvordan denne fungerer og hva de ønsker vi skal fokusere på i vår oppgave.

Vi har også startet en prosess med å innhente videomateriale fra en Manta notvasker som vi kan bruke til testing av funksjoner for bildebehandling. Dette er en prosess som tar litt tid fordi filmen må produseres av en Manta notvasker som er i bruk og lagres direkte på en datamaskin uten å redusere kvaliteten på filmen. Dette vil bedriften gjøre ved første mulighet. Vi har også fått oppdragsgiver til å prøve å få tak i videobilder fra en not med skader/ hull for å ha relevant materiale til å teste programvare på.

Vi ser for oss å benytte web basert oppslagsverk for OpenCV og Java.

5.5 Vurdering – analyse av risiko

Diagrammet under viser vår risikoanalyse av prosjektet. Vi anser elementer mot høyre som mer sannsynlige, og elementer høyere oppe vil ha større konsekvenser for suksessen til prosjektet. I tillegg til å kartlegge risikoer har vi også vurdert å iverksette tiltak for å redusere sannsynlighet og planlegge hva vi kan gjøre for å minimere konsekvensen hvis problemet oppstår.


Dårlig bildekvalitet
Miste filer/dokumenter ved crash/feil
Langtidsravær ved sykdom/skade
Konsekvens
Kompatibilitetsproblemer/mangler på funksjonalitet/støtte OpenCV og Java
Sansynlighet

Dårlig bildekvalitet på videoene er den risikoen vi ser på som den største trusselen mot suksess. Er bildekvaliteten for dårlig vil det være svært vanskelig å utføre bildeanalysen, og dermed vil vi ikke kunne oppdage hull i merdene. Vi vurderer risikoen for dårlig bildekvalitet til å være noe sannsynlig. Forholdene under vann er utfordrende og kan variere mye fra video til video. Vi har liten mulighet til å redusere denne risikoen. Det er mulig å bytte ut kamera og få bedre belysning, men dette vil ta tid og er sannsynligvis utenfor vår tidsramme. Viser det seg at bildekvaliteten er for dårlig kan vi lage en teoretisk løsning, men det vil være vanskelig å implementere på Mantaen.

En annen stor risiko er å miste filer/dokumenter ved en krasj eller feil på pc/programvare. Vi vurderer ikke sannsynligheten for dette å være stor, men det kan ha store konsekvenser hvis vi er uforsiktige. Vi vil derfor ta regelmessige sikkerhetskopier av programmer og dokumenter, slik at skulle en feil oppstå er det ikke nødvendig å gjøre alt på nytt. Vi har også besluttet å benytte Git for å sikre oss mot tap av programmvare. Git har i tillegg en del fordeler som for eksempel versjonskontroll.

Et av risikoelementene vi har vurdert er langtidsfravær av en eller flere av gruppemedlemmene. Vi vurderer risikoen for langtidsfravær til å være lite sannsynlig. Konsekvensen dersom en eller flere av medlemmene på gruppen blir borte over lengre tid er derimot betydelige. Vi har ikke valgt å iverksette tiltak mot dette risikoelementet utover bevisstgjøring av samtlige gruppemedlem.

Vi har vurdert kompatibilitetsproblemer/ mangel på funksjonalitet/ støtte til å være en middels stor risiko, med middels stor konsekvens for oss. Det er vanskelig å vurdere risikoen for dette på forhånd, og lite vi kan gjøre for å sikre oss mot det etter at vi har bestemt oss for hvilken programvare vi ønsker å benytte. Men de programvarene vi har valgt er utbredte og anerkjente, så vi tror det vil gå bra.

Det kan være en miljøgevinst i å ferdigstille prosjektet fordi det vil bidra til å oppdage eventuelle skader i merder på et tidlig tidspunkt. Dette vil deretter bidra til at skadene kan utbedres raskt og dermed minske antall fisk som rømmer. Det er et potensielt miljøproblem at oppdrettsfisk rømmer.

Vi vurderer prosjektet som mulig å realisere i løpet av tilgjengelig tid med de begrensninger vi har satt. Men det er svært vanskelig å beregne hvor lang tid vi vil trenge. Det er mulig vi må avgrense prosjektet dersom deler av det, for eksempel bildebehandlingen viser seg å være mer tidkrevende enn vi ser for oss.

5.6 Hovedaktiviteter i videre arbeid

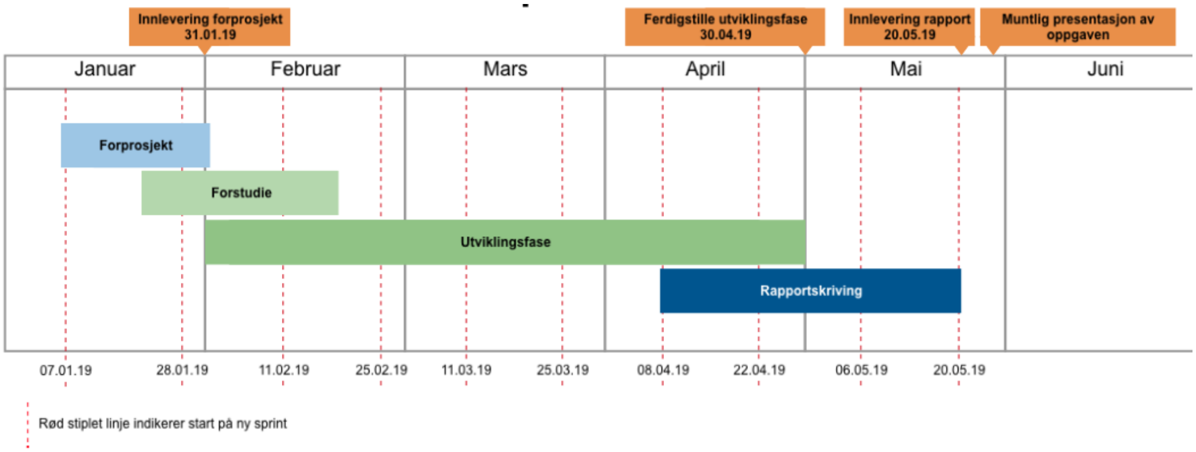
Beskrivelser av planlagte hovedaktiviteter og viktigste delaktiviteter for gjennomføring av prosjektet.

A1. Forprosjekt, Tidsramme 3 uker,
A2. Forstudie, Tidsramme 4 uker
A3. Utviklingsfase, Tidsramme 3 Måneder
A4. Rapportskriving, Tidsramme 6 uker

5.7 Framdriftsplan – styring av prosjektet

5.7.1 Hovedplan

-



Hovedplanen har oppmerket sprintdatoer i underkant og milepæler i overkant.

Forprosjektet går ut på å innhente informasjon om og sette oss inn i oppgaven.
Forstudiet går ut på å eksperimentere, teste ut funksjoner, sette seg inn i bildebehandlingen osv.
Utviklingsfasen går ut på å produsere selve programvaren.
Rapportskrivingen går ut på å skrive selve bachelorrapporten.

5.7.2 Styringshjelpemidler

Vi ønsker å bruke JIRA som et styringshjelpemiddel. Her kan vi legge inn hver sprint med prioriterte arbeidsoppgaver.

5.7.3 Utviklingshjelpemidler

Litteratur:

Scott E Umbaugh: Digital Image Processing and Analysis., CRC Press (2011)

Nettbaserte oppslagsverk som:

https://docs.oracle.com/javase/8/docs/api/
https://docs.opencv.org/4.0.1/javadoc/index.html

5.7.4 Intern kontroll – evaluering

Vi vil vurdere fremdrift for hver sprint ved å se hvor mange av sprint oppgavene vi greier å bli ferdige med for hver sprint. Ved å studere dette vil vi vurdere hvor god fremdriften har vert for hver sprint. Dette vil også bli tatt hensyn til i utviklingen av sprintoppgavene og med det beregnet fremdrift for neste sprint.
Vi vil gjennomføre et «Sprint retrospektiv møte» i slutten av hver sprint, i forkant av statusmøte med veileder. Her reflekterer vi over den siste sprinten i forhold til hva som gikk bra, og hva som kan forbedres i måten vi jobber på, tilgangen til ressurser osv.

Prosjektleder har ansvar for intern kontroll av fremdrift underveis i prosjektet. Dette blir utført ved å se på hvordan arbeidet med planlagte oppgaver for hver sprint ligger an.

Hver av de beregnede sprintoppgave er et delmål for en sprint. Når en sprintoppgave blir ferdigstilt endrer vi status på den i JIRA fra «in progress» til «done». Dette kjennetegner at et delmål er nådd.

5.8 Beslutninger – beslutningsprosess

Bestemmelse om avgrensing av oppgaven til å gjelde videostrøm fra ett kamera ble tatt i felleskap under første møte med både gruppemedlemmene, representantene for Stranda Prolog AS og våre veiledere til stede.

Gruppen planlegger å ta andre store beslutninger i fellesskap på statusmøtene med veileder og oppdragsgiver til stede.

6 DOKUMENTASJON

6.1 Rapporter og tekniske dokumenter

Det skal skrives en bachelorrapport mot slutten av oppgaven. Vi vil også lage en brukerveiledning til vårt program.

7 PLANLAGTE MØTER OG RAPPORTER

7.1 Møter

7.1.1 Møter med styringsgruppen

Gruppen har avtalt med veiledere at vi skal ha statusmøter annenhver fredag.
Vi har planlagt statusmøter på følgende datoer:

- 08.02.2019
- 22.02.2019
- 08.03.2019
- 22.03.2019
- 05.04.2019
- 19.04.2019
- 03.05.2019
- 16.05.2019

7.1.2 Prosjektmøter

- Vi har ikke planlagt egne prosjektmøter innad i gruppen, gruppen jobber tett hele uken.

7.2 Periodiske rapporter

7.2.1 Framdriftsrapporter (inkl. milepæl)

Det skal skrives statusrapport (i samband med Sprint Retrosective møtet) innen klokken 16:00 torsdag annenhver uke i forkant av statusmøte på Fredag. Denne rapporten skal skrives i Confluence.

Det skal skrives møtereferat annenhver fredag i etterkant av statusmøtene. Dette møtereferatet skal også skrives i Confluence.

Det skal skrives fremdriftsplan for neste sprint hver fredag i etterkant av statusmøtene. Denne lager vi som en sprint i Jira

8 PLANLAGT AVVIKSBEHANDLING

Dersom prosjektet ikke går som planlagt skal man kontakte veileder og deretter vurdere å endre fremgangsmåte.

9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

Vi ser ikke et behov for spesielt utstyr utover det som tidligere er nevnt i rapporten.

2019-01-11 Oppstartsmøte forprosjekt

Dato

11.jan.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat
- Kjetil Osland Brekken

Mål

- Oppstartsmøte for å komme i gang med forprosjekt for bacheloroppgave om Manta notvasker.

Møtereferat

Møtereferat første møte

Dato: 11.01.2019

Tid: 10:00-11:00

Til stede på møtet: Arne Gerhard Styve, Saleh Abdel-Afou Alaliyat, Endre Slettemark Skeide, Vebjørn Bjørlo-Larsen, Kjetil Osland Brekken, Jacob Hoseth.

Jacob Hoseth startet med å fortelle en del om Manta net cleaneren og hvordan han hadde tenkt seg oppgaven:

Denne blir nyttet fra et moderfartøy som senker den ut i mærer med en kran. Manta`en blir forsynt med kraft og komunikasjon igjennom en umbilical. Føreren av Mantaen opererer den fra et kontrollrom ombord I moderfartøyet. Mantaen har flere små propeller til manuvrering, og flere disker med høtrykks dyser som roterer under for å vaske nettingen I mærene. 8 videokamera er montert på mantaen og gir fører god oversikt. Mantaen blir senket til bunnen av mæren og føreren kjører den oppover i et c mønster mens den vasker. Jacob Hoseth mente vi burde begrense oppgaven til å jobbe med videostrømmen fra et bestemt kamera som står i forkant midt på Mantaen. Dette kameraet ser fremover, og har et bra bilde av nettet. Jakob mente at aktuelle vinklinger på vår oppgave kunne være deteksjon av hull I noten, eller produksjon av dokumentasjon/ rapport av vaskeresultat. Han mente at deteksjon av hull I noten ville være av størst verdi for Stranda prolog AS. Vi mener også at det vil være den mest interessante vinklingen og ønsker derfor å ha hovedfokus på dette i vår bacheloroppgave. Jacob mente de fleste mærene er hvite med firkantet maskemønster. En utfordring kan være at det ofte finnes en del distraksjonsmomenter I noten. For eksempel langsgående utstyr som støtter opp noten.

Kjetil vil være vår kontakt underveis I arbeidet med oppgaven, og han ville starte å samle sammen bilde og videomateriale som han kunne sende til oss.

Hovedoppgaven de neste to ukene vil vere å komme i gang med å lage en forprosjektrapport. Forprosjektrapporten vil kort fortalt beskrive oppgaven, og hvordan vi planlegger å gå frem for å løse den. For å gjøre dette må vi sette oss godt inn i, og få klarlagt flest mulige uklare forhold ved oppgaven.

Etter att Kjetil og Jacob hadde forlatt møtet gav Arne og Saleh oss en del praktisk info.

Vi avtalte med Arne å ha et møte torsdag 17.01.2019 08:15 – 09:00 med Kort inføring i JIRA & Confluence for Bachelor Manta net- cleaner prosjektet

Skriv inn oppgaven din her. Bruk «@» for å tildele til en bruker og «//» for å velge en tidsfrist.

Handlingselementer

2019-01-16 Møte med StrandaProlog AS v/ Kjetil Osland Brekken

Dato

16.jan.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Kjetil Osland Brekken

Mål

- Målet med møtet var å få en enda bedre forståelse for Manta Notvaskeren, hvordan denne fungere og hvordan den brukes. Få flere detaljer om oppgaven.

Etter Oppstartsmøtet 11.01.2016 har all informasjonen vi fikk fått tid til å synke litt inn, vi har sett over malen til forprosjekt, og startet å tenke igjennom hvordan vi vil vinkle oppgaven. I den forbindelse har det dukket opp en del spørsmål. Vi tok derfor initiativet til et møte med Kjetil for å få stilt en del spørsmål, og forsikret oss om at vi har forstått oppgaven korrekt.

Vi fortalte at vi har bestemt oss for at vi vil å fokusere på deteksjon av hull og skader i noten i vår oppgave. Vi ønsker også å generere en rapport over testresultatene. Kjetil mente dette hørtest ut som en god plan.

Vi ble enige om at vi skal prøve å få presentert resultatene i en GUI.

Vi snakket litt om bildebehandling i sanntid (altså direkte fra videostream) kontra bildebehandling i ettertid der vi benytter lagrede videofiler. Kjetil mente begge deler var aktuelt for Stranda Prolog, men at han anbefalte oss å fokusere på bruk av videofiler. Bildebehandlingen vil vere den samme uansett, og det vil være enkelt å tilpasse programmet til å kunne fungere i sanntid vet et senere tidspunkt, dersom det blir behov for det.

Vi spurte litt om det var speielle programmeringsspråk de benyttet. Kjetil mente vi burde bruke java siden det er det vi har mest erfaring med fra før av.

Vi snakket litt om programmvare og maskinvare de bruker i dag. Kjetil mente at mye av den programmvaren de benyttet i dag kom til å bli oppdatert i nær fremtid, så det trenger vi ikke ta hensyn til. Maskinene de bruker er kraftige.

Vi fikk også enda litt mer forklaring på hvordan Manta notvaskeren fungerer og hvordan den blir brukt.

*Skriv inn oppgaven din her. Bruk «@» for å tildele til en bruker og «//» for å velge en tidsfrist.*Handlingselementer

2019-01-25 Statusmøte

Dato

25.jan.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- saleh alaliyat
- Arne Styve

Mål

- Formålet med møtet er å avklare status og fremgang for inneværende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
15 min	Spørsmål om innhold i forprosjektrapporten	Endre og Vebjørn	• Vi bør fremstille en overfladisk fremdriftsplan selv om vi benytter Scrum, Denne trenger ikke inneholde mange elementer men den skal dekke hele semesteret frem til bacheloroppgaven er ferdig. Eksempel på innhold er forprosjektrapport, utvikling og rapportskriving. I tillegg merker vi av tidspunkter for planlagte sprinter. • Vi fikk forslag til hvordan vi kan utforme en risikoanalyse, og tips om å fokusere på risikoelementer som kan hindre oss i å lykkes med vår oppgave. Altså ikke risikoen for at vår oppgave/ vårt arbeid skader andre/ miljøet osv. • Vi fikk tips om å lese oss opp på Scrum som metode og legge til info om den i forprosjektrapporten. • Vi avtalte at Arne og Saleh skal få hvert sitt eksemplar av forprosjektrapporten når den er ferdig. • Vi tar sikte på å ferdigstille forprosjektrapporten senest mandag 28.01.2019
10 min	Spørsmål om avgrensing/ utviding av oppgaven dersom vi får god tid.	Endre og Vebjørn	• Vi ble enige om at vi i tillegg til deteksjon av skader/ hull i merdene vil fokusere på å lage en brukervennlig applikasjon for å analysere videofiler om tiden strekker til. Her tenker vi at bruker skal kunne velge videofil for analyse, enkelt gjøre instillinger/ tilpassinger til analysen, og deretter for eksempel kunne skrive en kommentar på resultat.

Handlingselementer

☑ Oppdater forprosjektrapporten i henhold til det som kom frem på møtet.

☑ Oppdater prosjektforsiden i Confluence med fremdriftsplan og litt mer info, et eventuelt nytt gruppemedlem skal kunne sette seg inn i prosjektet.

2019-02-08 Statusmøte

Dato

08.feb.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møtet er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10 min	Diskuterte litt hva vi har jobbet med de siste ukene	Endre og Vebjørn	• Arne opprettet et rom i Confluence der vi kan legge inn Forprosjektrapporten.
10 min	Spørsmål om å skrive informasjon til fil	Endre og Vebjørn	• Vi ba om synspunkt på hvordan vi bør lagre til minne så vi slipper å legge inn verdier hver gang man kjører programmet på nytt. • Arne kom med tips om å lagre til en xml-fil og benytte Java DOM parser. • Dette syntes vi hørtes greit ut og vil se videre på dette.
5 min	Bruk av Jira og Confluence	Arne	• Arne sa litt om hva han syntes om vår Confluence dashboard. Vi fikk tilbakemelding om at vi kunne legge til vår kontaktperson hos StrandaProlog på denne siden. • Vi fikk også tilbakemelding om at vi kan prøve å lage enda mer konkrete oppgaver for hver sprint, slik att det er lett å vite når vi er ferdige med hver oppgave.
5 min	Spørsmål om bildebehandling til Saleh	Endre og Vebjørn	• Vi har ikke kommet i mål med å finne metoder for å ta utsnitt av bilde med varierende bakgrunn/lysforhold (på deler av videomaterialet er deler av bakgrunnen lysere enn noten, mens resten av bakgrunnen er mørkere enn noten.) Vi ønsket å høre om Saleh kansje hadde noen tips eller ideer til hvordan vi kan komme videre i dette arbeidet. • Vi ble med Saleh til hans kontor og fikk pratet litt om dette. • Saleh mente vi var på rett vei, men at vi bare må fortsette å utforske og teste ulike metoder, vi fikk også tips til metoder vi kunne sjekke ut, som for eksempel Region growing.

Handlingselementer

□

2019-02-11 Møte med StrandaProlog AS v/ Kjetil Osland Brekken

Dato

11.feb.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Kjetil Osland Brekken

Mål

- Vi har på forhånd sendt Kjetil et eksemplar av vår Forprosjektrapport, og ønsker å ta et møte for å få tilbekemeldinger på Forprosjektrapporten, samt underrette Kjetil om hva vi har arbeidet med så langt.

Kjetil syntest forprosjektrapporten vår så fin ut, han mente rapporten så ut til å dekke det meste godt.

Vi fortalte om det vi har gjort til nå:

- laget kode for å hente opp video og plukke ut bilder for behandling
- har laget kode for å ta utsnitt av bilder
- laget rammeverk for å sette sammen og teste flere funksjoner
- testet mange ulike bildebehandlings funksjoner for å finne ut hva som fungerer best
- På bilder som er gode, greier vi å detektere hull/objekt i bildet.

Har jobbet med to ulike metoder for å detektere hull:

- detektere knuter i nettet, og deretter se på tettheten av knuter, ikke i mål med denne enda
- ser på hver maske i nettet som et objekt, og deretter vurdere størrelsen på hvert objekt, dette er også en metode vi jobber med.

En av problemstillingene vi har møtt er at en del bilder har ulik bakgrunn, deler av bildet har lysere bakgrunn enn nettet, mens andre deler har mørkere bakgrunn enn nettet, her er det vanskelig å skille ut hva som er nett og ikke.

Kjetil var godt fornøyd med innsatsen og mente vi var kommet godt i gang.

Handlingselementer

- □

2019-02-22 Statusmøte

Dato

22.feb.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møtet er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10 min	Fremdrift i inneverede sprint		• Vi snakket om hvordan vi syntes sprinten hadde gått, og hvordan vi selv syntes det gikk. Vi synest det går bra, vi jobber godt sammen, og føler vi har fremdrift hver dag.
10 min	Ta ut brukbare områder fra dårlige bilder	Vebjørn og Endre	• Diskuterte ulike løsninger for å skille ut de bildene/ deler av bilder som ikke er gode nok til å detektere hull/ skader på. Vi jobber med en løsning der vi tenker å dele opp bildene som ikke er gode nok i flere utsnitt, og deretter sjekker hvert utsnitt for seg. Vi fikk tilbakemelding på at dette er en aktuell måte å gjøre det på, men at det vil være enda bedre om vi finner en måte som tar ut eventuelle brukbare områder i bildet uavhengig av firkantet utsnitt. Eventuelt å bare detektere områder som er for dårlige, merke disse på en slik måte at de ikke blir prosessert senere i programmet. Dette er noe av det vi vil vi jobbe videre med i løpet av neste sprint.

Handlingselementer

□

2019-03-11 Statusmøte

Dato

11.mar.2019

Deltakere

- Vebørn Bjørlo-Larsen
- Endre Slettemark Skeide
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møtet er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Time	Item	Who	Notes
10 min	Snakket om hva vi har jobbet med de siste ukene og hva vi skal fokusere på de neste ukene	Endre og Vebjørn	• Sette opp en ny modul i tilbakeblikk for å skrive hva vi skal gjøre i neste sprint og beholde modulen der vi går gjennom hva vi kunne ha gjort bedre.
15 min	Gikk gjennom en del om GUI og hvordan den kan settes opp.	Endre og Vebjørn	• Arne snakket om likhetene og forskjellene mellom javafx og swing, kom fram til at vi bør gjøre enkle tester med begge for å finne ut hva som passet best for oss. • Arne viste hvordan vi kan bruke verktøy i NetBeans for å sette opp en javafx GUI med drag and drop. • Hovedprogrammet bør være et javafx prosjekt som vi legger inn våre andre klasser i. • Snakket om å ha dialog med Kjetil fra Stranda Prolog om utviklingen av GUIen, sende dem forslag og høre va de ønsker å ha i GUIen.
10 min	Diskuterte oppsett av hovedfunksjon	Endre og Vebjørn	• Vi viste et enkelt diagram av strukturen til hovedprogrammet, Arne mente det var et fornuftig utgangspunkt • Arne foreslo at vi setter opp et sekvensdiagram for å få oversikt over klassene og flyten mellom dem, evt. gjøre endringer basert på det. • Snakket om å bruke Giphy til å lage diagram og GUI eksempler.
10 min	Spørsmål om hvordan vi bør lagre data	Endre og Vebjørn	• Spurte om vi bør bruke en form for register eller skrive resultater til en fil (xml). • Arne mente det å skrive til xml-fil var en god løsning. • Kom fram til at det er fornuftig å lagre bilder med bilde ID i en mappe og en xml fil der resultatene blir lagret under denne bilde IDen.

Action items

- □

2019-03-15 Møte med StrandaProlog AS v/ Kjetil Osland Brekken

Dato

15.mar.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Kjetil Osland Brekken

Mål

- Formålet med dette møtet var å presentere status for Kjetil, og få tilbakemelding på arbeidet vårt så langt. Vi ønsket også å presentere skisser med forslag til GUI, for å få tilbakemeldinger, og eventuelle innspil på desse.

Vi startet med å presentere dagens status:

Vi har nå tre forskjellige funksjoner som detekterer uregelmessigheter i nettet:

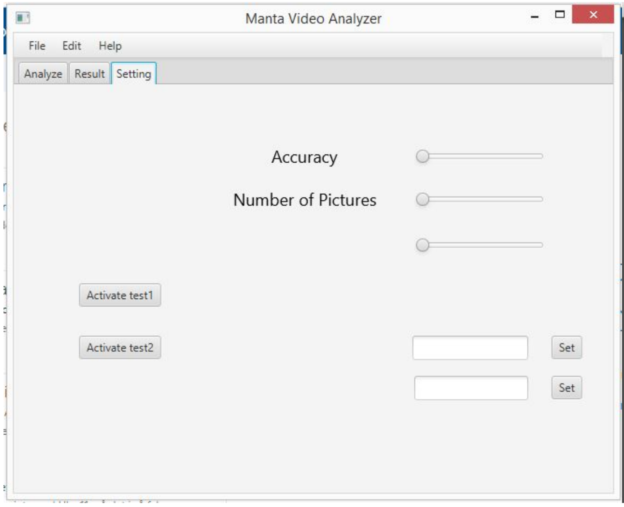
- detektere knuter i nettet, og deretter se på tettheten av knuter.
- ser på hver maske i nettet som et objekt, og deretter vurdere størrelsen på hvert objekt.
- en som også ser på masker i nettet som objekt men som ikke gjør om til skeleton bilde.

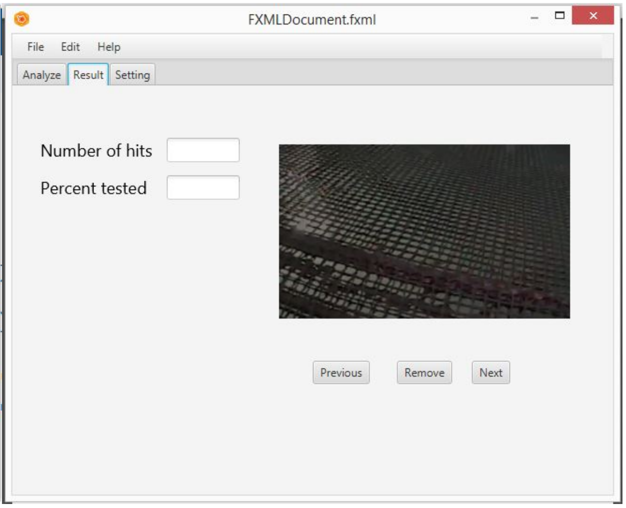
Vi har laget en kvalitetstest som benytter histogrammet til hvert bilde og ser etter bimodal fordeling.

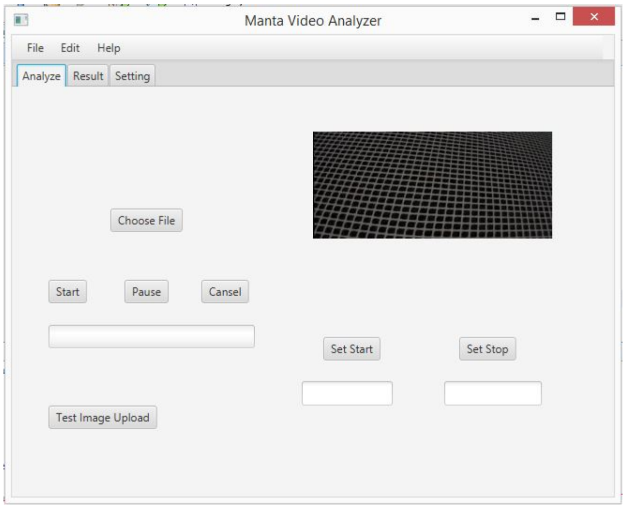
Vi tenker å lagre resultater i en xml fil og bilder med treff i en mappe underveis, og skrive en pdf fil til slutt.

Fremmover vil vi nå fokusere på å sette sammen alt vi har laget så langt til et fungerende program, slik at vi får testet og finjustert funksjonene bedre.

Vi presenterte og forklarte desse skissene vi har laget, de viser hvordan vi ser for oss at GUI`en kan se ut:







Kjetil syntest dette så og hørtes flott ut, han mente det var akkurat noe slikt han hadde sett for seg. Vi ba om innspill til GUI`en og vi fikk noen forslag:

- Kjetil ønsket at starter med å lage enkle (mer automatiske) justeringer på settings siden i GUI`en. Desse skal vere enkle å benytte.
- Om vi får tid kan vi utvide med mer manuelle og avanserte innstillingar.
- Farge konturer på treff i bildene som vi benytter i result siden på guien, slik at det blir enkelt for bruker av applikasjonen å se hvor i bildet skadene er detektert.
- Undersøke hvor vidt det er mulig å kjøre programmet på grafikkort kontra cpu.

Vi vil se på mulighetene for å inplementere dette i prosjektet etterhvert.

Handlingselementer

□

2019-03-25 Statusmøte

Dato

25.mar.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møte er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10.min	Presenterte arbeidet vi har gjort siden forrige møte.	Vebjørn og Endre	Vi gikk igjennom statusrapporten sammen, og viste Arne og Saleh hvor langt vi har kommet på GUI`en, og hvilken funksjonalitet vi har i dag.
5.min	Vi ba om tilbakemelding på programmet slik det er i dag	Vebjørn og Endre	• Arne og Saleh var generelt godt fornøyde med GUI`en, de syntest den så bra ut. • Arne kom med forslag om å lage til listevisning på resultatbildene. • Prøve å unngå at knappene på analyse-fanen flytter seg over bildeviseren når vi soomer.
5.min	Vi ønsket råd om hva vi bør fokusere på fremmover/ hva vi bør legge vekt på.	Vebjørn og Endre	• Arne mente vi burde passe på å ikke bruke for mye tid på å lage fancy GUI, men prioritere at GUI`en blir enkel og brukervenlig, samt at programmet blir stabilt. • Han foreslo også at vi starter å tenke på rapporten, i hvert fall opprette den, sette oss inn i malen, og tenke igjennom hvilken informasjon vi vil sette inn under de ulike delene.
10.min	Arne presenterte ulike hjelpemiddel for å gjøre vårt java program om til en kjørbar fil som kan installeres og kjøres for alle windows maskiner.	Arne	• Zotero er et hjelpemiddel vi kan sjekke ut for å hondtere kilder. • Install4j er et hjelpemiddel for å lage installeringsfiler. • Launch4j er et hjelpemiddel for å lage kjørbar fil.

Handlingselementer

☑ Arne Styve Skal finne tidligere bacheloroppgaver som er gode.

2019-04-10 Møte med StrandaProlog AS v/ Kjetil Osland Brekken

Dato

10.apr.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Kjetil Osland Brekken

Mål

- I forkant av dette møtet sendte vi en video til Stranda Prolog, som viser programmet med GUI i bruk, mens vi kommenterer og forklarer funksjonaliteten. Formålet med dette møtet var å få tilbakemeldinger på denne videoen. Vi ønsket å få eventuelle innspill til forbedringer på GUI og funksjonalitet før vi går inn i den siste fasen med ferdigstilling av programmet.

Kjetil syntes videoen så bra ut, han mente vi hadde fått til mye.

I videoen nevner vi at vi har planer om å forbedre en del småting, blant annet PDF dokumentet som viser treff, mulighet for å velge plassering på resultatfilene og å lage en hjelpeside med informasjon om hvordan programmet brukes. Kjetil kunne ikke se noen åpenbare ting som burde forbedres utover de detaljene vi selv nevnte i videoen.

Kjetil lurte på om vi selv så noen klare forbedringspotensial. Her nevnte vi at vi mener kvaliteten på videoene som blir analysert er der det største forbedringspotensialet ligger. Ved å endre på kameravinkel og bedre lysforhold vil man kunne forbedre treffsikkerheten i vårt program betraktelig. Når det kommer til forbedringer i vårt program har nok algoritmene som skiller ut hvilke områder som er for dårlige til å detektere en del forbedringspotensial. Det er nok også litt å hente på enda bedre kalibrering av analysene, men dette krever mye tid og mye videomateriale. GPU aksellerasjon vil kunne forbedre hastigheten på programmet, slik at det tar kortere tid å analysere en video

Kjetil vil prøve å få tak i enda mer film materiale som vi kan bruke til testing og kalibrering.

Handlingselementer

- □

2019-04-10 Statusmøte

Dato

10.apr.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møte er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10.min	Presenterte arbeidet vi har gjort siden forrige møte.	Vebjørn og Endre	• Vi viste Arne og Saleh settingssiden vi har laget i guien, og demonstrerte den nye funksjonaliteten.
10.min	Minte Arne på at han skulle sende relevante gode tidligere bacheloroppgaver, som vi kan bruke for å få et inntrykk av hvordan oppgaven bør settes opp.	Vebjørn og Endre	• Fikk et eksempel på tidligere bacheloroppgave.

Handlingselementer

- ☑ Arne Styve skal sende ny Jira link til Kjetil.
- ☑ Arne Styve skal sende bacheloroppgave om fiskefilet.

2019-04-26 Statusmøte

Dato

26.apr.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- Arne Styve
- saleh alaliyat

Mål

- Formålet med møtet er å avklare status og fremgang for inneverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10. min	Gikk igjennom fremdriften siden forrige statusmøte, og planer for neste sprint.	Arne	• Vi har ikke kommet i mål med å lage installasjonsfil for programmet, dette anbefalte Arne å ta igjenn når vi ser at vi nærmer oss slutten på rapporten.
10. min	Hadde en del spørsmål om oppsettet på rapporten	Vebjørn og Endre	• Vi fikk oppklart litt som gikk på kopiering, sitering og oppsett.

Handlingselementer

- □

2019-05-09 Statusmøte

Dato

09.mai.2019

Deltakere

- Endre Slettemark Skeide
- Vebørn Bjørlo-Larsen
- saleh alaliyat
- Arne Styve

Mål

- Formålet med møtet er å avklare status og fremgang for innleverende sprint, samt planlegge og diskutere eventuelle uklarheter rundt neste sprint. Møtet er også en fin mulighet til å stille veiledere eventuelle spørsmål vi har.

Debattpunkter

Tidspunkt	Element	Hvem	Notater
10. min	Gikk igjennom fremdriften siden forrige statusmøte, og planer for neste sprint.	Alle	• Vi er godt i gang med rapporten, og vil fortsette å ferdigstille denne i neste sprint.
20.min	Gikk igjennom det vi har skrevet i rapporten så langt, hadde en del spørsmål om oppsett/ innhold.	Alle	• Vi skal lage en svært detaljert og god javadoc, i klassene, da trenger vi ikke å skrive detaljert om hver klasse i rapporten, bare henvise til javadoc. • UML standard på klassediagram. • Flytdiagram. • Vi skal gjøre en analyse og presentere resultatene fra den i resultatdelen av rapporten. Vi skal også diskutere desse resultatene og vurdere hvorfor resultatene blir slik de blir i diskusjonsdelen på rapporten. • Det viktigste skal stå først i rapporten. Bidlebehandlingen er det viktigste.

Handlingselementer

□

2019-01-24 Tilbakeblikk

Dato	24.jan.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Samarbeid og oppmøte
- Har installert OpenCV
- Har mottatt en testvideo, har fått til å åpne den i Netbeans/ OpenCV og dele opp i bilder med ønsket tidsintervall for videre prosessering.
- Har fått satt oss litt inn i bildebehandlingsfunksjoner og har testet ut noen slike i Netbeans/OpenCV

Hva burde vi ha gjort bedre?

- Har jobbet videre med forprosjektrapporten, den har blitt ganske bra, men vi har en del spørsmål, særlig om avgrensing/ utviding av oppgave
- Venter fortsatt på høy-oppløst videomateriale fra frontkamera, Stranda Prolog prøver å få gjort slike opptak i løpet av de nærmeste dagene.

Handlinger

- ☐

2019-02-07 Tilbakeblikk

Dato	06.feb.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Samarbeid og oppmøte
- Har fått testet en del funksjoner, noen av funksjonene ser vi at vi kan få nytte av, noen er ikke nyttige for oss.
- Har mottatt bra bildemateriale fra Stranda prolog AS, som er bra å bruke til testing.
- Har startet å bygge et rammeverk for testing av funksjoner.

Hva burde vi ha gjort bedre?

- Har ikke kommet i mål med å finne metoder for å ta utsnitt av bilde, med varierende bakgrunn/lysforhold (på deler av videomaterialet er deler av bakgrunnen lysere enn noten, mens resten av bakgrunnen er mørkere enn noten.)

Handlinger

- □

2019-02-21 Tilbakeblikk

Dato	21.feb.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Samarbeid og oppmøte
- Har testet ut bruk av Java DOM parser, dette passer bra til vårt bruk.
- Har utviklet to forskjellige metoder for å detektere hull/ fisk. Begge tar utgangspunkt i skeleton bilde. Den ene benytter maskene, og lager et objekt av hver maske, deretter ser den etter objektstørrelser som utmerker seg. Den andre detekterer maskeknuter, og ser etter områder uten knuter.
- Vi har testet ut en måte å dele opp et bilde i flere mindre bilder for å kunne plukke ut brukbare deler av dårlige bilder.
- Begynt å se på metoder for skille ut gode/dårlige bilder. Vil fortsette å arbeide med det.
- Vi har hatt et møte med Stranda Prolog, for å orientere om hvordan vi ligger an så langt.

Hva vi tenker å arbeide med neste sprint?

- Vi vil i første omgang prøve å få gjort ferdig den metoden vi allerede jobber med, som går på å sjekke om bilder har god nok kvalitet, og deretter dele bildet i for eksempel 4 deler og plukke ut de delene som er gode nok.
- Vi vil deretter prøve å lage en metode som plukker ut bare de gode områdene i bilder og som lager en region of interest basert på dette. (uten å dele opp bildet i firkanter)
- Starte å planlegge GUI, sette opp en plan for hvordan den skal se ut, hvilke funksjoner den skal ha osv.
- Sjekke om det kan vere et alternativ å dektektere hull/ skader i nettet bare ved å lete etter store svarte områder.
- Se mer på avansert segmentering.

Handlinger

- □

2019-03-10 Tilbakeblikk

Dato	10.mar.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Sammarbeid og oppmøte
- Vi har satt oss inn i, og testet ut PDFBox, dette hjelpemiddelet vil vi benytte til å produsere en PDF for hver video vi tester. I denne PDF`en tenker vi å legge bilder som vi detekterer feil på, samt enkel informasjon om hvor mye feil, hvor i videoen osv.
- Vi har laget en ny måte å detektere hull på som benytter inverterte sort hvitt bilder og leter etter store konturer. Denne fungerer bra på noen av bildene.
- Vi har laget en metode som sjekker bildekvalitet basert på histogram, (se etter bimodal fordeling) denne vil vi bruke til å plukke ut dårlige bilder, eller deler av dårlige bilder.
- Vi har også laget en funksjon som vi kan bruke til å korrigere de delene av bilder som ikke er gode nok til å analyseres med våre metoder, slik att våre funksjoner ikke vil detektere feil her bare fordi bildene er dårlige.
- Vi har laget en foreløpig plan for hvordan vår hovedapplikasjon kan struktureres.
- Vi har ferdigstilt de funksjonene fra våre testfiler som vi ønsker å bruke videre slik at de er klare til brukt i den endelige applikasjonen.

Hva tenker vi å jobbe med i neste sprint?

- Så langt har vi utarbeidet mange funksjoner i testdokument som vi har testet hver for seg. I løpet av neste sprint ønsker vi å prøve å få satt desse funksjonene sammen til et fungerende komplett program.
- Vi ønsker også å klargjøre dette programmet for GUI, og forhåpentlig vis få startet å programmere en GUI.

Hva fikk vi ikke til?

- Vi har lett etter muligheter for å lage en region of interest som ikke deler bildene opp i firkanter, men uten å lykkes med dette.
- Dette er noe av det vi tenker vi kan bruke mer tid til dersom det blir tid til overs etter at vi har satt sammen en fungerende applikasjon med GUI.

Handlinger

- □

2019-03-25 Tilbakeblikk

Dato	24.mar.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Sammarbeid og oppmøte.
- Vi klargjorde hver av funksjonene vi har laget tidligere, slik at de passer inn i de klassene vi vil bruke i vårt program.
- Vi opprettet klasser, og satte de sammen til et fungerende program.
- Vi undersøkte og gjorde oss kjent med javafx og swing for å vurdere hvilke av desse vi skulle bruke. Begge er gode alternativ, vi valgte å benytte javafx i vårt program, i hovedsak fordi vi har inntrykk av at de fleste mener javafx er i ferd med å ta over for swing.
- Vi Installerte og satte oss inn i SceneBuilder, denne vil vil vi bruke som et hjelpemiddel når vi skal bygge vår GUI.
- Vi hadde et møte med Kjetil fra Stranda Prolog. Vi gikk igjennom status så langt, hva vi har gjort og hva vi vil fokusere på fremover. Vi viste frem skisser med forslag til hvordan vi ser for oss at vi vil lage guien. Kjetil var godt fornøyd med våre skisser og mente det var akkurat noe sånt han hadde sett for seg. Vi fikk også stilt noen spørsmål, og fikk gode svar og innspil til funksjonalitet o.l.
- Har også gjordt en del forbedringer på den testen som detekterer knuter i nettet, og er godt i gang med å kalibrere inn/ finhjustere denne.
- GUI
  - Har satt opp hovedvindu med 3 tabs, analyse, resultat og instillinger. Mulighet til å zoome fungerer for det meste.
  - Analyse-tab: lagt inn funksjonalitet for å velge videofil, gå gjennom video i GUIen, sette start og stop-punkt for analysen og starte selve analysen. Har også lagt til progress bar og tid brukt/estimert gjenstående tid.
  - resultat-tab: lagt til funksjonalitet for å lese inn resultat xml-fil og bla gjennom hvert bilde med informasjon om hvert bilde (treff, etc..).
  - lagring av resultat og bilder er hardkodet, skal arbeide mer med det

Hva burde vi ha gjort bedre?

Hva ønsker vi å jobbe med i neste sprint?

- Fortsette å jobbe med guien.
- Fortsette å jobbe mot ferdigstilling av programmet, ting som står igjen her er for eksempel en del som går på skriving og lesing av XML filer og skriving av PDF.
- Fortsette å kalibrere / justere inn bildebehandlingsfunksjonene for å få disse til å fungere mest mulig optimalt.

Handlinger

- □

2019-04-09 Tilbakeblikk

Dato	09.apr.2019
Deltakere	Vebørn Bjørlo-Larsen Endre Slettemark Skeide

Tilbakeblikk

Hva fikk vi bra til?

- Sammarbeid og oppmøte.
- Vi har kommet et godt stykke lenger med guien, for eksempel har vi nå bygget opp settingssiden, med enkle justeringer for å tilpasse programmet. Vi har også laget til mulighet for å opprette og lagre flere settings profiler som kan tilpasses for eksempel ulike videotyper, kameravinkler, lysforhold osv.
- Vi har kommet et godt stykke lenger med programmet, vi har nå implementert skriving og lesing av XML filer og skriving av PDF.
- Vi har også kommet en del lenger i arbeidet med å kalibrere / justere inn bildebehandlingsfunksjonene for å få disse til å fungere mest mulig optimalt.
- Vi har fått til å merke funn på bildene med ringer som indikerer hvor i bildet funn er lokalisert, og hvilken av våre tester som har gjort funnet.
- Vi har opprettet overleafprosjekt for selve bachelorrapporten, og er i gang med å sette oss inn i og planlegge oppsettet for rapporten.

Hva burde vi ha gjort bedre?

Hva ønsker vi å jobbe med i neste sprint?

- Starte å skrive Bachelorrapporten
- Sammenstille hele programmet, er en del småtteri her og der som må ferdigstilles
- Rydde koden, og legge inn kommentarer
- Lage installeringsfil, slik at programmet kan lastes ned og kjøres på ny maskin på en enkel måte.
- Sikre programmet, exception handling, trådsikkerhet, avslutte programmet på en sikker måte etc.

Handlinger

- □

2019-04-25 Tilbakeblikk

Dato	25.apr.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Sammarbeid og oppmøte.
- Vi har ordnet opp i hardkodet filplasseringer.
- Vi har forbedret PDF dokumentet med blant annet forside med analysestatistikk
- Vi er godt i gang med siste finpuss på koden,kommentarer osv.
- Vi har gjort om analysedelen fra singlethreaded til multithreaded
- Vi har prøvd å lage installeringsfil for programmet, men ikke kommet i mål med dette
- Vi har studert bachelor mal og andre bachelor oppgaver.

Vi har kommet i gang med bachelorrapporten:

- førte kapittel, Introduksjon, her er første utkast ferdig
- kapittel tre, material og metoder, her er vi godt i gang.

Hva burde vi ha gjort bedre?

Hva ønsker vi å jobbe med i neste sprint?

- Hovedfokus i neste sprint blir å skrive bachelorrapporten.
- Muligens prøve litt mer på å få til å lage installeringsfil.
- Gjøre siste detaljer/finpuss i koden

Handlinger

□

2019-05-08 Tilbakeblikk

Dato	08.mai.2019
Deltakere	Endre Slettemark Skeide Vebørn Bjørlo-Larsen

Tilbakeblikk

Hva fikk vi bra til?

- Sammarbeid og oppmøte.
- Vi har stort sett jobbet med rapporten, her har mye kommet på plass, men det står igjen litt, særlig under diskusjon, og sammendraget er ikke startet på.
- Programmet er ferdigstilt og bra kommentert.
- Vi har kommet i mål med å lage installeringsfil for programmet.
- Fant en feil i Knute analysen, som vi har fått rettet opp i.

Hva burde vi ha gjort bedre?

Hva ønsker vi å jobbe med i neste sprint?

- I løpet av neste sprint vil vi finpusse og ferdigstille rapporten.
- Dersom det blir tid, ønsker vi å teste ut noen potensielle forbedringer for kvalitetskontrollen.
- Dersom det blir tid ønsker vi å prøve å få lagt til mulighet for å justere størrelsen på videoutsnittet som blir analysert.
- Dersom det blir tid ønsker vi å prøve å få lagt inn manuell justering av hvor mange deler kvalitetskontrollen skal kunne dele bildet opp i.

Handlinger

- □

NTNU

Kunnskap for en bedre verden